Physica A 486 (2017) 65–78

Contents lists available at ScienceDirect

Physica A

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

A density based link clustering algorithm for overlapping community detection in networks

Xu Zhou [a,e], Yanheng Liu [b,c,*], Jian Wang [f], Chun Li [d]

[a] Center for Computer Fundamental Education, Jilin University, Changchun, Jilin 130012, China
[b] College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China
[c] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China
[d] China of Limited Company of Jilin Province Power Communication Company, State Grid, Changchun, 130012, China
[e] College of Communication Engineering, Jilin University, Changchun 130012, China
[f] Department of Intelligent Vehicle, China Automotive Engineering Research Institute (CAERI), Chongqing, 404100, China

HIGHLIGHTS

- A density based link clustering algorithm is proposed to detect overlapping communities.
- An updating strategy for unclassified edges is designed to assign them to the closest cluster.
- Our algorithm performs better than other algorithms in the experiments.

ARTICLE INFO

Article history:
Received 5 January 2017
Received in revised form 25 May 2017
Available online 30 May 2017

Keywords:
Overlapping community detection
Density based link clustering
Edge similarity

ABSTRACT

Overlapping is an interesting and common characteristic of community structure in networks. Link clustering method for overlapping community detection has attracted a lot of attention in the area of social networks applications. However, it may make the clustering result with excessive overlap and cluster bridge edge and border edge mistakenly to adjacent communities. To solve this problem, a density based link clustering algorithm is proposed to improve the accuracy of detecting overlapping communities in networks in this study. It creates a number of clusters containing core edges only based on concept named as core density reachable during the expansion. Then an updating strategy for unclassified edges is designed to assign them to the closest cluster. In addition, a similarity measure for computing the similarity between two edges is presented. Experiments on synthetic networks and real networks have been conducted. The experimental results demonstrate that our method performs better than other algorithms on detecting community structure and overlapping nodes, it can get nearly 15% higher than the NMI value of other algorithms on some synthetic networks.

1. Introduction

Community detection technology is widely applied in many different fields such as physics, biology, computer science, epidemiology, economy and sociology [1–3]. The technique of detecting overlapping communities in networks is crucial to guide significance for research of network topology, and the detected overlapping nodes which belong to multiple

* Corresponding author.
E-mail address: yhliuj@163.com (Y. Liu).

http://dx.doi.org/10.1016/j.physa.2017.05.032
0378-4371/

communities may play an influential role for network analysis. For example, it can be used to detect criminal gangs, find potential customers, and personalized information recommendation service [4,5] and so on.

The study of overlapping community detection has attracted an increasing attention recently. Many different algorithms coming from different fields such as physics, statistics and data mining have been presented to identify communities in networks. Originally speaking, the density based clustering algorithms in the scope of data mining have been well studied and applied to solve community detection problem due to its simplicity and the ability of finding clusters of different sizes and shapes even in the presence of noise. However, the community detection methods that based on density clustering algorithm are only to detect non overlapping structures [6,7]. This is because they take node as research object and the result is absolutely non-overlapping. Then to solve overlapping problem, hierarchy algorithms of clustering links in networks are put forward, but one particular drawback of those algorithms is that they can get highly overlapping structure, which is not applicable in most cases. Here, it gives us a hint. If we take density based clustering algorithm to cluster links instead of nodes in networks, then it is a new way to find overlapping community structures.

In a word, We explore the density based clustering algorithm and propose a density based link clustering algorithm (DBLC) for overlapping community detection in networks in order to solve the problem of getting highly overlapping structures by link based hierarchy clustering algorithm. In this algorithm, we take link/edge (we use link and edge interchangeably) as research object. Firstly a similarity function to measure similarity between two links is presented. To consider the effect of border link to the final clustering results, we obtained clusters with core links based on core density reachable definition. Then, an updating strategy is designed to assign unclassified edge to the closest community to which the edge belongs. By utilizing the idea of density based clustering, our algorithm can obtain the overlapping community structure with high quality. Furthermore, the clustering result does not depend on the order of processed links. It has the ability of identifying noise edges to avoid excessive overlapping. Simulation results on synthetic networks and real networks confirm the effectiveness of the algorithm on detecting communities and overlapping nodes.

The rest of the paper is organized as follows. Existing link based techniques for overlapping community detection and some basic definitions of density based clustering algorithm are reviewed in Section 2. A density based link clustering algorithm for overlapping community detection is presented in Section 3. The experimental results on synthetic datasets and real world datasets with their corresponding analysis respectively are shown in Section 4. Finally, conclusion of the paper and future research directions are given in Section 5.

2. Related works and background concepts

This section summaries the existing link based clustering algorithms for overlapping community detection and gives basic concepts of density based algorithms.

2.1. Link based clustering

Community structure is a common characteristic in the network. Community detection is to identify groups of densely connected vertices, and it is a necessary analysis mean widely existing in networks. With the development of research, it is found that the nodes of network have multiple roles. That is to say, a node can be divided into different communities. The existing non-overlapping community detection methods cannot meet the requirements of finding overlapping structures. In this way, several methods for discovering overlapping communities are proposed accordingly. The main techniques to detect overlapping communities according to the research object can be roughly divided into two categories, node based and link based overlapping community detection algorithms [8].

In this paper, we mainly focus on the link partition techniques. In this line of research, link instead of node is as the basic unit in the algorithm. The link based algorithms cluster the edges of network, and map the final link communities to node communities by simply gathering nodes incident to all edges within each link communities. A node in the original graph is called overlapping if links connected to it are put in more than one cluster. The link partition approaches usually have more advantage in detecting overlapping communities. Therefore, many link community detection methods are proposed and they range from hierarchical clustering algorithms to statistical approaches using generative model and random search methods with evolutionary algorithms. The typical link clustering algorithm for overlapping community (Link) is proposed by Ahn [9]. The author employed Jaccard index as similarity measure and presented a single-linkage hierarchical clustering algorithm to build a link dendrogram and a new density objective function for cutting best level of dendrogram. The algorithm tends to find small communities and cannot provide the view of the global community structure of the network. This methods will lead to clustering results with highly overlapping structure. The authors described a method based on a probabilistic model of link communities to find overlapping communities and implement it by using a closed form expectation maximization algorithm to analyze network [10]. Research of [11] extended the Infomap algorithm to find link communities by using the minimum description length principle. An extended link clustering method (ELC) for overlapping community detection is presented in [12]. The method employs a new link similarity and extension of modularity (EQ) to find the best level for hierarchical clustering dendrogram division. It gets better results within small networks. Evans introduced the idea of partitioning the links to change the original network into the edge graph, in which community detection is adopted, and then the result of community detection is obtained [13,14]. These methods are memory inefficient, they cannot be applied to large networks. The combination of the line graph with evolutionary algorithms has been explored in literatures [15–17].

Table 1
Symbols summary.

Symbols	Description
$n(v)$	Neighbor set of v
$N_\varepsilon(e)$	ε neighbor set of edge e
$\|N_\varepsilon(e)\|$	Cardinality of $N_\varepsilon(e)$
$sim(e_1, e_2)$	Similarity between edge e_1 and e_2
$nei(e)$	The neighbor set of edge e
$core(e)$	e is a core edge
$noise(e)$	e is a noise edge

Parliamentary Optimization Algorithm (POA) has been firstly proposed as a novel overlapping community detection method in [15]. Both methods in [16,17] are a genetic algorithm developed for detecting overlapping community with link clustering. The algorithm first finds the link communities by optimizing objective function, and then map the link communities to node communities based on a novel genotype representation method. The difference between two algorithms is their objective function, partition density in [16] and community score in [17] respectively. Those methods using evolutionary algorithms can get overlapping structures, but the precision of the results is not satisfying.

Considering the approaches for community detection based on density clustering algorithm, only a few proposals can be found. There is still space to research. This paper presents an overlapping detection algorithm based on density clustering. It can identify the isolated edges which do not belong to any community and avoid get communities with excessive overlapping nodes. At the same time, an updating strategy is designed to take into account the processing of border edges. In addition, we present a new similarity computation formula for assigning similarity between links that share a common node. A simple parameter selection strategy is added into this algorithm. The experimental results show the proposed algorithm is effective.

2.2. Basic concepts

A number of definitions of density based algorithm are reviewed. Considering a network represented by an undirected graph $G = \langle V, E\rangle$ in which V is the set of vertex and E is the set of edges. The purpose of overlapping community detection in G is to determine a partition $P = \{C_1, C_2, \ldots, C_m\}$ of all the nodes of G where communities can be overlapping ($\exists i, j \in m, C_i \bigcap C_j \neq \emptyset, i \neq j$). Communities are group of nodes having dense intra connections and sparse inter connections. The main symbols in this paper are listed in Table 1.

Definition 1 (*Neighbor Set of* v)**.** The neighbor of vertice v is the set of vertex which are adjacent to vertice v. It is defined as follow.

$$n(v) = \{w \in V | (v, w) \in E\} \cup \{v\} \tag{1}$$

Definition 2 (*Neighbor Set of* e)**.** Assume $e_{v,v'} \in E$, then its neighbor link $nei(e)$ can be defined as the set of edges which are adjacent to vertex v or v', and $e_{v,v'}$ does not belong to this set, that is

$$nei(e_{v,v'}) = \left\{e_{v,i} \in E | i \in n(v) \text{ and } i \neq v'\right\} \cup \left\{e_{j,v'} \in E | j \in n(v') \text{ and } j \neq v\right\} \tag{2}$$

where $n(v)$ and $n(v')$ represent the neighbor set of vertice v and v'.

Definition 3 (ε *Neighbor of Edge* e)**.** The ε neighbor of edge e is defined as follows. It contains the edges that satisfy the similarity threshold.

$$N_\varepsilon(e) = \{x \in nei(e) | sim(x, e) \geq \varepsilon\} \tag{3}$$

where $sim(x, e)$ denotes the similarity between two edges x and e.

Definition 4 (*Core*(e))**.** Given a integer μ, if the cardinal number of the ε neighbor set of e is larger than a threshold μ ($|N_\varepsilon(e)| \geq \mu$), then e is a core $core(e)$. Here, if an edge is not a core and it is in the ε neighbor of a core edge, it is taken as a border edge.

Definition 5 (*Directly Density Reachable*)**.** An edge $e_1 \in E$ is directly density reachable from an edge e_2 if two points are satisfied. One is that $e_1 \in N_\varepsilon(e_2)$ and the other is $|N_\varepsilon(e_2)| \geq \mu$ (e_2 is a core), the direct density reachability exists when there is at least one core between the pairs of edges.

Definition 6 (*Density Reachable*)**.** For all the edges in E, an edge e_j is density reachable from an edge e_i if there is a chain of edges which meets the conditions that e_{i+1} is directly density reachable from e_i with $e_i, e_{i+1}, \ldots, e_j$ and $j > i$. The last edge in the chain can be not a core. The density reachability is symmetric only for two core edges.

Definition 7 (*Density Connected*)**.** Edge e_1 is density connected to edge e_2 if there is an edge $e_3 \in E$ and both the two edges are density reachable from edge e_3. The density connectivity is symmetric for any two edges.

Definition 8 (*Edge Cluster and Noise*(e))**.** A cluster that is composed with edges is called edge cluster. The edge does not belong to any cluster is taken as a noise edge.

3. The proposed method

3.1. Description of the DBLC

In order to solve the problem of finding unnecessary small clusters and producing too many overlapping nodes by link clustering algorithm, we propose a density based link clustering algorithm to detect overlapping community structures. In this algorithm, it starts from a core link and find a cluster with all the core links based on core density reachability. Then it starts an unclassified core link to form another cluster until all the links are scanned. In this way, a number of clusters are obtained which contain the core links only. Next we take an updating strategy to deal with the unclassified border links. It can overcome the issue of bridge edge and border edge mistakenly belonging to adjacent communities. Our method is not sensitive to the order where the links are processed. Moreover, we present a new similarity computation formula for assigning similarity between links that share a common node. The experimental results demonstrate that DBLC can detect overlapping community structures and overlapping nodes effectively. It has the ability of identifying noise edges to avoid excessive overlapping, and it performs well especially when the number of communities that each overlapping node belongs to is large. In this section a detailed description of the algorithm is given. The framework of DBLC is shown in Algorithm 1.

In our research, the framework of DBLC includes of six steps. Step 1 and step 2 is to do initialization of the input network. It is to assign an index to each edge and get incidence matrix based on adjacency matrix. The incidence matrix shows the relationship between node and edge which can be used for converting link to node. Similarity of all the pairs of edges are obtained in Step 3 and the specific similarity measure is shown in next subsection. Step 4 can be called expansion step. A number of clusters are found at the end of Step 4. And the method to obtain one cluster starting with a core edge is based on the algorithm 2. Step 5 is the updating strategy, which can be taken as a post processing too. It is to assign the unclassified edge to the cluster to which it belongs based on the similarity between it and the cluster. In the end, it is necessary to transfer the link cluster into cluster with nodes according to the incidence matrix.

Algorithm 1 Framework of DBLC

Input: a complex network $G = (V, E)$, ε, μ
Output: the set of overlapping communities $P = \{C_1, C_2, \ldots, C_m\}$
Step 1. assign an index to each edge
Step 2. get incidence matrix W based on adjacency matrix A of G
Step 3. generating a similarity matrix between a pair of edges according to incidence matrix W
Step 4. $LP \leftarrow \emptyset$
  For each edge e_i in E
    If e_i is not classified and *core*(e_i)
      expand a new cluster X according to Algorithm 2
      $LP \leftarrow LP \cup X$
    End If
  End For
Step 5. update the link partition LP with unclassified edge according to Algorithm 3 to get final link partition FLP
Step 6. convert link set in each partition FLP to the node set P

Suppose n edges $e_1, e_2, \ldots, e_n$ compose of a density reachable chain, then there are two conditions, one is that all the edges n are core links, the other one is that $n - 1$ edges are core links and the last one is a border one.

Expanding from a core edge iteratively to get a new cluster based on the density reachable concept will make the result of clustering depend on the order of the processed links. This is because that a border edge can be reached by different paths and in the neighborhood of different core objects. Therefore, many density reachable chains may share same border link. That is to say, it can belongs to more than one clusters. However, the problem is that the border link will be assigned to the cluster discovered first (the object is connected via the first density reachable chain of the cluster), other than the cluster which the link has most similarity to. The result may be different starting from different edge based on density reachable concept. For this reason, we adopt the concept of core density reachable proposed by Thanh [18], a chain of core density reachable contains the core object only $e_1, e_2, \ldots, e_n$, where e_i is a core edge for all $i \leq n$. In order to identify clusters correctly, we do not consider the contribution of border edge in the expansion mechanism. After the analysis, the edge expansion strategy for clustering based on core density reachable is shown in Algorithm 2. It creates a new cluster with a core edge, and iteratively collects core density reachable edge from the edge. The process terminates when no new edge can be added to any cluster. Then it randomly selects another unclassified core edge to expand, the clustering process continues

until all the edges are scanned. There will be some clusters only containing core edges after expansion procedure. The post processing for unclassified border links is shown in Algorithm 3. In Algorithm 3, it is necessary to identify the cluster to which the unclassified edge belongs. It is to update the clusters with the unclassified edges synchronously to get the final link partition.

Algorithm 2 Edge expansion

Input: edge e, two parameters ε, μ
Output: a temp link cluster X

```
Q ← Ø.initial an empty queue Q
insert e into Q
while Q is not empty do
  y ← Q.peek(). y is the first one in Q
  to identify all density reachable edges into the set R
  for each x ∈ R do
    if x is not classified then
      X ← X ∪ x
      insert x into Q
    end if
  end for
  remove y from Q
end while
```

Algorithm 3 Updating strategy

Input: rough partition $LP = \{X_1, \ldots, Xj\}$
a set of unclassified edges $ue = (e_1, \ldots, e_i, \ldots, e_n)$
Output: Final link clustering sets *FLP*

```
for each edge e_i is unclassified in ue do
  max = ε, count = 0
  for each X_j in LP do
    if e_i is connected with a core edge in X_j then
      m = j
      count = count + 1
      find the maximum similarity sim_{i,j} between e_i and the edge in X_j as the similarity between e_i and X_j
      if sim_j ≥ max then
        max = sim_j
        k = j
      end if
    end if
  end for
  if count == 1 then
    the edge e_i belongs to the m^th partition
  else
    the edge e_i belongs to the k^th partition
  end if
end for
update link partition LP with unclassified edge to get final partition result FLP
```

3.2. Edge similarity

Edge similarity is the basis for clustering. Jaccard method is employed in Link algorithm [9]. Similarity between two edge $e_{u,v}$ and $e_{u,w}$ with a common node is equal to similarity between two vertices (v and w). It can be computed as the number of common neighbors of two vertices (v and w) divided by the number of union of their neighbors. Here, a edge similarity with considering the edge connecting common neighbor vertices of v and w is shown. The similarity formula between edge $e_{v,v'}$ and edge $e_{w,w'}$ is shown from Eqs. (4)–(7). The new defined similarity computation composes of two items. A parameter γ controls the weight of two items sim_1 and sim_2. Edges sharing a node are expected to be more similar than disconnected edges. For a pair of edges without common node, the similarity is zero. The first item is the number of common neighbors of two vertices divided by the number of union of two vertices' neighbors. The second item is the number of links between common neighbors divided by the number of links that could possibly exist between them, and it quantifies how close the common neighbors are connected with each other. It is intuition that if two friends have many common friends and common

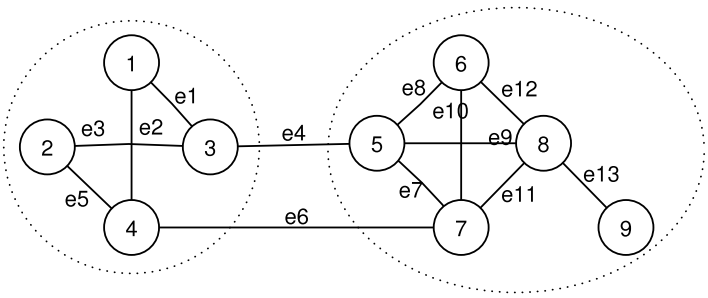

Fig. 1. A simple network with bridge edge.

friends know each other more (that is to say, there are more edges among common nodes), these two friends will have more chance to be together.

$$sim(e_{v,v'}, e_{w,w'}) = \gamma \times sim_1(e_{v,v'}, e_{w,w'}) + (1-\gamma) \times sim_2(e_{v,v'}, e_{w,w'}) \tag{4}$$

$$sim_1(e_{v,v'}, e_{w,w'}) = \begin{cases} 0 & \text{if } v \neq w \text{ and } v' \neq w' \\ \dfrac{comm(v,w)}{n(v) \cup n(w)} & \text{if } v' = w' \\ \dfrac{comm(v',w')}{n(v') \cup n(w')} & \text{if } v = w \end{cases} \tag{5}$$

$$comm(i,j) = n(i) \cap n(j) \tag{6}$$

$$sim_2(e_{v,v'}, e_{w,w'}) = \begin{cases} 0 & \text{if } v \neq w \text{ and } v' \neq w' \\ \dfrac{2 \times |E(comm(v,w))|}{(|comm(v,w)|)(|comm(v,w)-1|)} & \text{if } v' = w' \\ \dfrac{2 \times |E(comm(v',w'))|}{(|comm(v',w')|)(|comm(v',w')-1|)} & \text{if } v = w \end{cases} \tag{7}$$

where $|comm(i,j)|$ denotes the number of common neighbor of vertices i and j, $n(i)$ denotes neighbors of vertex i including itself, $E(comm(i,j))$ denotes the edges existing among common neighbors of vertices i and j.$|E(comm(i,j))|$ denotes the cardinal number of $E(comm(i,j))$. The parameter controls the weight of the two items, it is the Jaccard method when the parameter is set to 1, Here we take a toy example to show the difference for computing two edges by our method and Jaccard.

Taking toy network in Fig. 1 as example. The similarity between e_1 and e_2 computed by ours when $\gamma = 0.8$ is $0.8 * 0.33 = 0.264$, and the value computed by Jaccard method is 0.33. It seems ours is acceptable, because there is no link between node 1 and node 2 (common neighbors of node 3 and node 4). Node 3 cannot reach node 4 by two hops as 3-1-2-4 indirectly. However, if there is a link between node 1 and node 2. The similarity between e_1 and e_2 by Jaccard method is still 0.33. Ours will get 0.462. So it seems reasonable to consider the edge connecting common neighbors of two nodes.

3.3. Parameter analysis

The density based clustering is associated with two parameters. ε is the similarity threshold of a neighborhood for an edge and μ is a threshold for the number of an edge in this neighborhood. The selected parameters will affect the final result of clustering. Under a certain μ, some core edges may be considered as noise and a cluster may be divided into a number of clusters mistakenly when ε is too large. Some noise edge may be mistakenly considered as core edge and some disjointed communities may become one when ε is too little. Similarly, under a certain ε, the edge with more than μ neighbors turns into a core edge. If μ is too little, it will detect clusters with high density. Otherwise, If μ is too large, it will lead to produce a few core edges and the cluster with a few members will not be detected.

As analyzed above, we use a simple heuristic method taking EQ [19] as the objective function to determine two parameters. First of all, it is necessary to map similarity value based on Eq. (8) to narrow the searching domain for proper ε value. In this way, the interval of ε is restricted from [0, 1] to [0.5, 0.74]. Shrinking the range of ε will have a limited amount of choices.

$$sim'(e_{v,v',e_{w,w'}}) = \frac{exp(sim(e_{v,v'}, sim(e_{w,w'})))}{1 + exp(sim(e_{v,v'}, sim(e_{w,w'})))}. \tag{8}$$

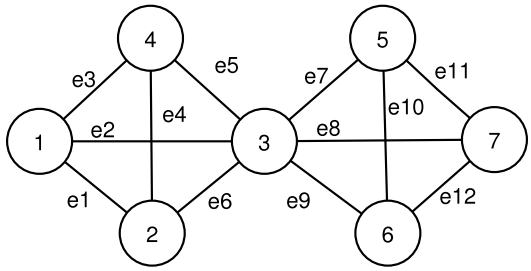

Fig. 2. A simple network with an overlapping node.

Our parameter selection method is simple and easy to implement. μ is less sensitive than ε as said in Ref. [7], and range of μ value is small. We adopt the strategy of maximizing the value of EQ to determine the optimal ε value when μ is given. Initially, the EQ is computed by setting an initial ε. Then keep on increasing the value of ε until community structure cannot improve the EQ value. The optimal ε value is the one corresponding to the greatest value of EQ.

EQ is a measure to evaluate the quality of overlapping community partition. It is defined in Eqs. (9) and (10)

$$EQ = \sum_{l=1}^{k} EQ_l \tag{9}$$

$$EQ_l = \frac{1}{|M|} \times \sum_{i \in H_l, j \in H_l} \frac{1}{O_i O_j} \left[A_{ij} - \frac{n_i n_j}{2|M|} \right] \tag{10}$$

EQ is equal to 0 when all nodes belong to the same community. In addition, a higher value of EQ indicates denser intracommunity connections of the community structure. Where M denotes the total number of edges in the network, k is the number of communities, and H_l is the node set of community. A_{ij} is adjacency matrix. $A_{ij} = 1$ only if two nodes are adjacent. $n(i)$ and $n(j)$ represents the degree of node i and node j. O_i represents the number of communities that node i belongs to.

Take an example here. For Link algorithm, it will assign the bridge edge to a community by hierarchy clustering algorithm in Fig. 1. But the bridge edges should not be divided into any community in fact. Our method successfully distinguishes the bridge edges e_4, e_6 and deals with them appropriately with parameter $\varepsilon = 0.54$ and $\mu = 2$, and the marginal edge e_{13} is divided to the cluster it connects to. The two detected clusters are {1,2,3,4} and {5,6,7,8,9}. The network in Fig. 2 contains two cliques with one overlapping node 3. DBLC can accurately reveal the overlapping communities when $\varepsilon = 0.64$ and $\mu = 4$. It evaluates the ability of DBLC to discover the overlapping node and bridge edges on the two toy networks.

3.4. Time complexity

We present an analysis of the time complexity of the algorithm DBLC. Given a graph with n nodes and m edges. The time needed in computing similarity between edges is $O(m^2)$ in step 3. It will check the neighbors of each edge to identify core edge in step 4, the complexity of which is $O(km)$. k is the average degree of the graph. The complexity of assigning unclassified edge to each link partition in step 5 is $O(lc)$, where c is number of clusters and l is the number of unclassified edge. The complexity of converting link set to node set in step 6 is $O(m)$. Because k is constant, and l is less than m. Therefore, the computational complexity of DBLC can be estimated to be $O(m^2)$.

4. Experimental results and analysis

Experiments on both synthetic and real networks are used to test and analyze the effectiveness of the DBLC algorithm comprehensively in this section. Here two classical overlapping community detection algorithms Link [9] and CPM [20] are for comparison with DBLC algorithm. Link is a link based overlapping community detection algorithm. CPM is an influential method based on the assumption that each community is a union of adjacent k-cliques. All the experiments are carried out on a PC with 2 GB RAM and 2.66 GHz Intel processor. The DBLC algorithm is implemented in C++. For Link and CPM algorithms, we used the software packages provided by the authors. They can be downloaded from http://barabasilab.neu.edu/projects/linkcommunities/ and http://www.cfinder.org/.

4.1. Evaluation criteria

Evaluating the quality of the detected overlapping community structures is nontrivial. The most widely used measures are the Normalized Mutual Information (NMI) and the modularity of overlap (Q_{ov}) [21]. NMI is used to measure the accuracy of

algorithm for networks in which the ground truth are known. Q_{ov} can be used to measure the clustering results for networks in which the ground truth are unknown. They both provide an overall measure of algorithmic accuracy at community level. If we want to evaluate an algorithm's ability to identify overlapping nodes, *Fscore* is another widely used measure criterion.

4.1.1. Normalized mutual information

NMI is a criterion commonly used to measure the difference between two overlapping communities [22]. Its value is between 0 and 1. If the value is higher, it indicates the detected community structure is more similar to the ground truth. For partition C and C', the main formulas of it are shown on Eqs. (11) and (12)

$$NMI = 1 - \frac{1}{2}[H(X|Y) + H(Y|X)] \tag{11}$$

$$H(X|Y) = \frac{1}{|C'|}\sum_{k}\frac{H(X_k|Y)}{H(X_k)} \tag{12}$$

where $X(Y)$ is the random variable associated to the partition C and C'. $H(X|Y)$ is the normalized conditional entropy of a cover X with respect to cover Y, $H(Y|X)$ is defined in the same way with $H(X|Y)$.

4.1.2. Fscore

It is used to measure accuracy of algorithms in detecting the overlapping nodes and it is the harmonic mean of precision and recall. Here the formula is shown as Eq. (13)

$$Fscore = \frac{2 * precision * recall}{precision + recall} \tag{13}$$

where *precision* is the number of correctly detected overlapping nodes divided by all the number of detected overlapping nodes, and *recall* is defined as the number of correctly detected overlapping nodes divided by the true number of overlapping nodes.

4.1.3. The modularity of overlap

It is an extension of the classical modularity for overlapping community detection defined by Nicosia. It takes into account the number of communities to which each vertex belongs and the degree of membership in each community. The higher the value is, the better the partition is. The overlap modularity can be formulated in Eq. (18).

$$F\left(\alpha_{i,c}, \alpha_{j,c}\right) = \frac{1}{\left(1 + e^{f\left(\alpha_{i,c}\right)}\right)\left(1 + e^{-f\left(\alpha_{j,c}\right)}\right)} \tag{14}$$

$$\beta_{l(i,j),c} = F(\alpha_{i.c}, \alpha_{j.c}) \tag{15}$$

$$\beta^{out}_{l(i,j),c} = \frac{\sum_{j\in V} F(\alpha_{i.c}, \alpha_{j.c})}{|V|} \tag{16}$$

$$\beta^{in}_{l(i,j),c} = \frac{\sum_{i\in V} F(\alpha_{i.c}, \alpha_{j.c})}{|V|} \tag{17}$$

$$Q_{ov} = \frac{1}{m}\sum_{c}\sum_{i,j\in V}\left[\beta_{l(i,j),c}A_{ij} - \beta^{out}_{l(i,j),c}\beta^{in}_{l(i,j),c}\frac{k^{out}_i k^{in}_j}{m}\right] \tag{18}$$

where $\alpha_{i,c}$ and $\alpha_{j,c}$ is belonging coefficient of node i and node j to community c separately. $\beta_{l(i,j),c}$ is the belonging coefficient of $l(i,j)$ for community c. $\beta^{out}_{l(i,j),c}$ is the expected belonging coefficient of any possible link $l(i,j)$ starting from a node into community c, and it is the average of all possible coefficients of belonging to c of l. $\beta^{in}_{l(i,j),c}$ is the expected belonging coefficient of any link $l(i,j)$ pointing to a node j in community c. $f\left(\alpha_{i,c}\right)$ is a simple linear scaling function: $f(x) = 60x - 30$. k^{out}_i is the number of outgoing links of node i. k^{in}_j is the number of incoming links of node j. m is the number of edges.

4.2. Experiments on synthetic networks

LFR benchmark network proposed by Lancichinetti and Fortunato is employed to generate artificial networks with well-defined communities [23]. LFR benchmark can simulate real networks well and generate scale-free network according to degree distribution and community size distribution. It allows communities to overlap. LFR provides many parameters to specify properties of generated networks, including the number of nodes N, exponent of power-law distribution of nodes degree τ_1, exponent of power-law distribution of community size τ_2, average degree $\bar{k}$, the maximum degree k_{max}, the

Table 2
LFR synthetic networks.

Network	N	μ	c_{min}	c_{max}	O_m	O_n
LFR1	1000	0.1	10	50	2,3,4,5,6	100
LFR2	1000	0.3	10	50	2,3,4,5,6	100
LFR3	1000	0.1	20	100	2,3,4,5,6	100
LFR4	1000	0.1	10	50	2,3,4,5,6	500
LFR5	1000	0.3	10	50	2,3,4,5,6	500
LFR6	1000	0.1	20	100	2,3,4,5,6	500

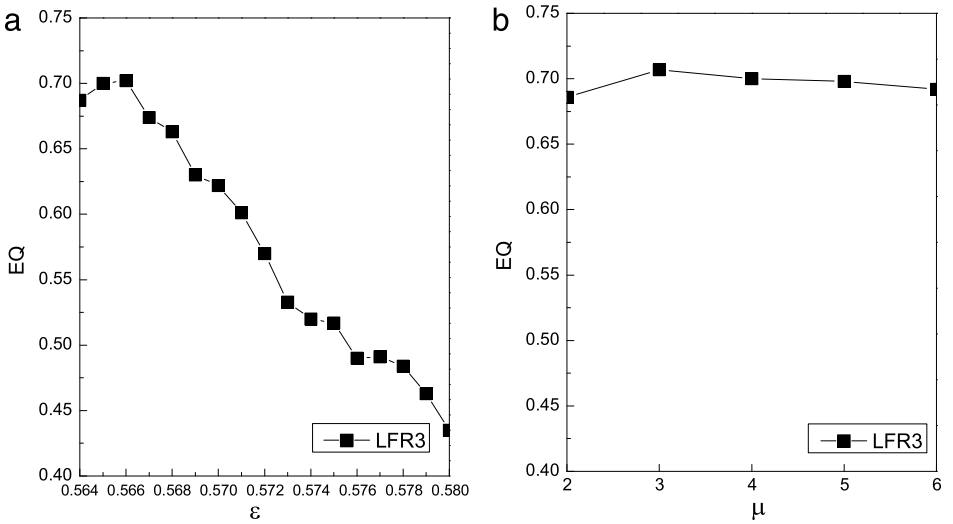

Fig. 3. The selection process of ε and μ in the operation (a) ε and the corresponding EQ when $\mu = 4$(b) μ and the corresponding EQ when $\varepsilon = 0.566$.

mixing parameter μ, the minimum community size c_{min}, the maximum community size c_{max}, the overlapping nodes in network O_n and the number of communities that each overlapping node belongs to O_m. c_{min} and c_{max} control the network topology.

We generate six synthetic networks with different parameters. $N = 1000$, $\tau_1 = 2$ and $\tau_2 = 1$, $k_{max} = 50$, and $\bar{k} = 10$. The mixing parameter is set to either 0.1 or 0.3. Community sizes vary in both small range (10, 50) and large range (20,100). O_n is set to 10% and 50% of the total number of nodes. O_m varies from 2 to 6 indicating the overlapping diversity of overlapping nodes. The specific parameters in the benchmark are given in Table 2.

Taking LFR3 as an example, the way of selecting parameter μ and ε is introduced as shown in Fig. 3. Given an initial $\mu = 4$, it is to adjust ε from 0.564 to 0.58 and find the corresponding EQ as shown in Fig. 3(a). The EQ is the maximum when $\varepsilon = 0.566$. With μ increasing from 2 to 6, the maximum value of EQ is about 0.71 when $\mu = 3$ as shown in Fig. 3(b). In summary, μ is set 3 and ε is set 0.566. The experimental results show that our method is valid to some extent. For CPM, parameter k varies from 3 to 5.

4.2.1. Identifying community structures

NMI is used to compare the accuracy of identifying overlapping communities between various algorithms. The comparative results are shown in Fig. 4. Each graph shows a comparative result of three algorithms on one network. Generally speaking, there is a common variation tendency for curve in each graph, and the curve is declining with the number of membership Om varying from 2 to 6. In other words, when the network is much fuzzier and there are more nodes belonging to multi communities, it is hard to get the true partition.

Fig. 4 clearly shows the very good performance of DBLC with respect to other methods. From Fig. 4(a)–(c), it exhibits that results of DBLC can nearly uncover 90%, 80%, 75% accurate communities respectively on LFR1, LFR2, LFR3 with $Om = 2$. It still can uncover 70% accurate communities on these three benchmarks with increasing Om. CPM can nearly uncover 70%, 60%, 40% accurate communities respectively on LFR1, LFR2 and LFR3 with $Om = 2$. Link can detect lower than 50% accurate communities on LFR1, LFR2 and LFR3. Seen in Fig. 4(d)–(f), when the network is become fuzzier and has more overlapping nodes, the ability of finding overlapping communities of all the algorithms on LFR5 and LFR6 becomes weak. The NMI value of DBLC obtained is about 0.4 on LFR5 with $Om = 2$. The NMI value decreases with increasing Om from 2 to 6. However, NMI value of Link and CPM is nearly 0.2, 0.3 respectively. They cannot detect the overlapping community structure

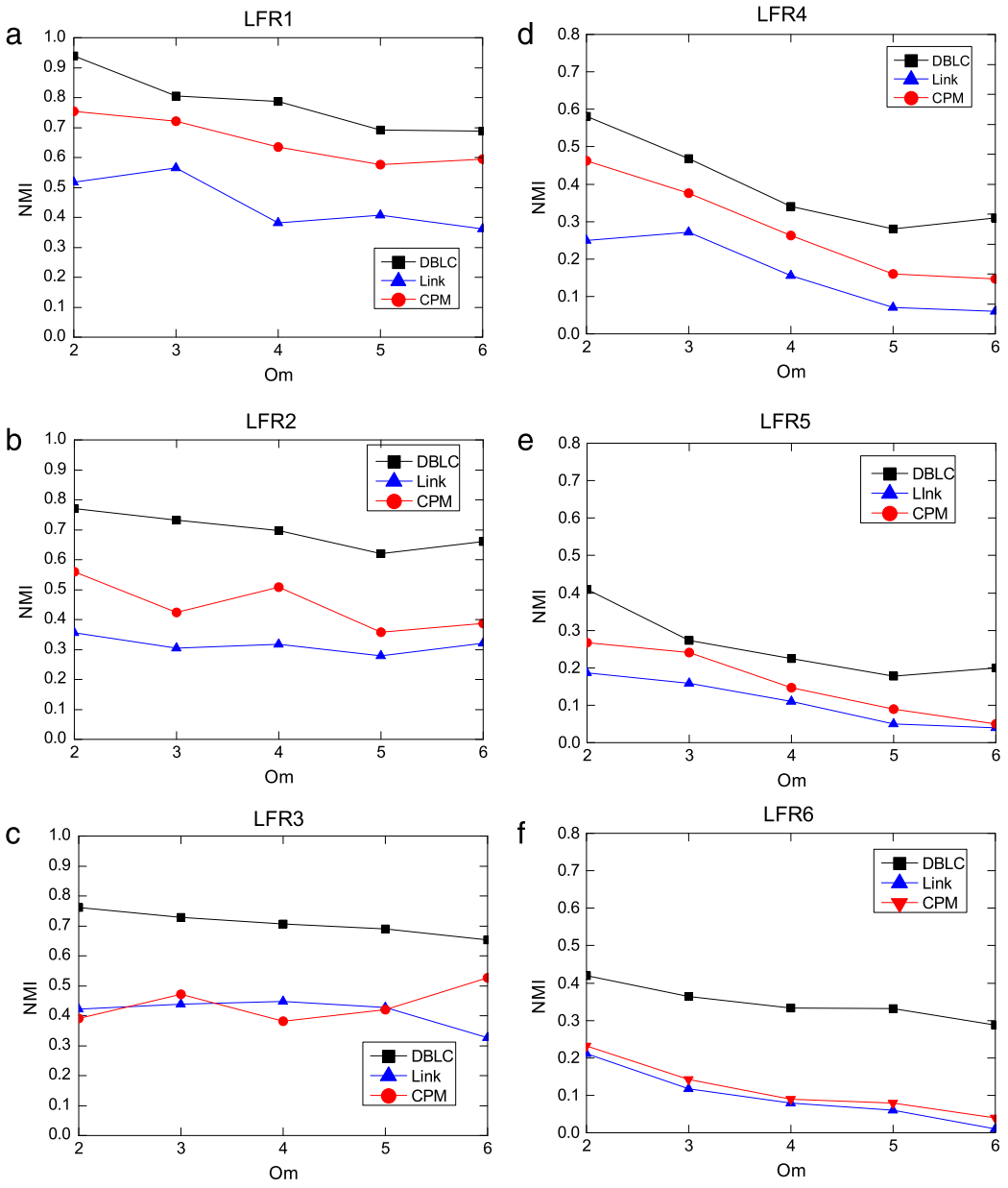

Fig. 4. The comparative NMI value obtained by three algorithms on six synthetic networks.

very well. In most cases, Link algorithm gets the lowest NMI value due to its detecting clusters with highly overlapping structure.

The NMI values of three algorithms on LFR1, LFR2, LFR3 and LFR4 are depicted in Fig. 5. In order to examine the effects of mixing parameter on different algorithms, we can only compare the performance of each algorithm on networks LFR1 and LFR2 in Fig. 5(a)–(c). The two networks only have different μ. It is obviously to see that the NMI value obtained by each algorithm on LFR1 is higher than that of each algorithm on LFR3. The community structure becomes fuzzy and the performance of all the algorithms deteriorates with the increasing value of μ.

To investigate effects of community size range on the algorithms, the performance of each algorithm on networks LFR1 and LFR3 in Fig. 5 is compared. The two networks have same size of 1000 nodes, but they have different community size range. Observing the tendency of diamond and uptriangle for these three algorithms, mostly the diamond is above the uptriangle, that is to say, NMI value for LFR1 with small community size range (10, 50) is typically higher than that for LFR3 with large community size range (20, 100). For Link algorithm, it always obtain many overlapping communities and it only detects the

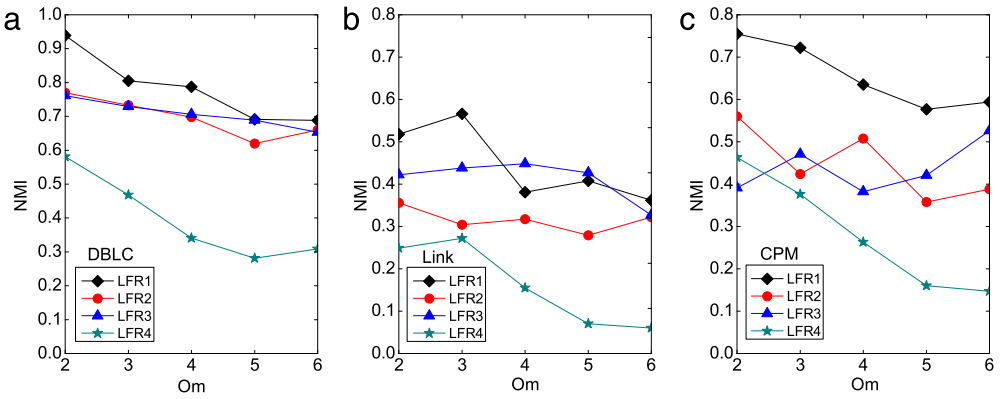

Fig. 5. The NMI value of each algorithm on four synthetic networks.

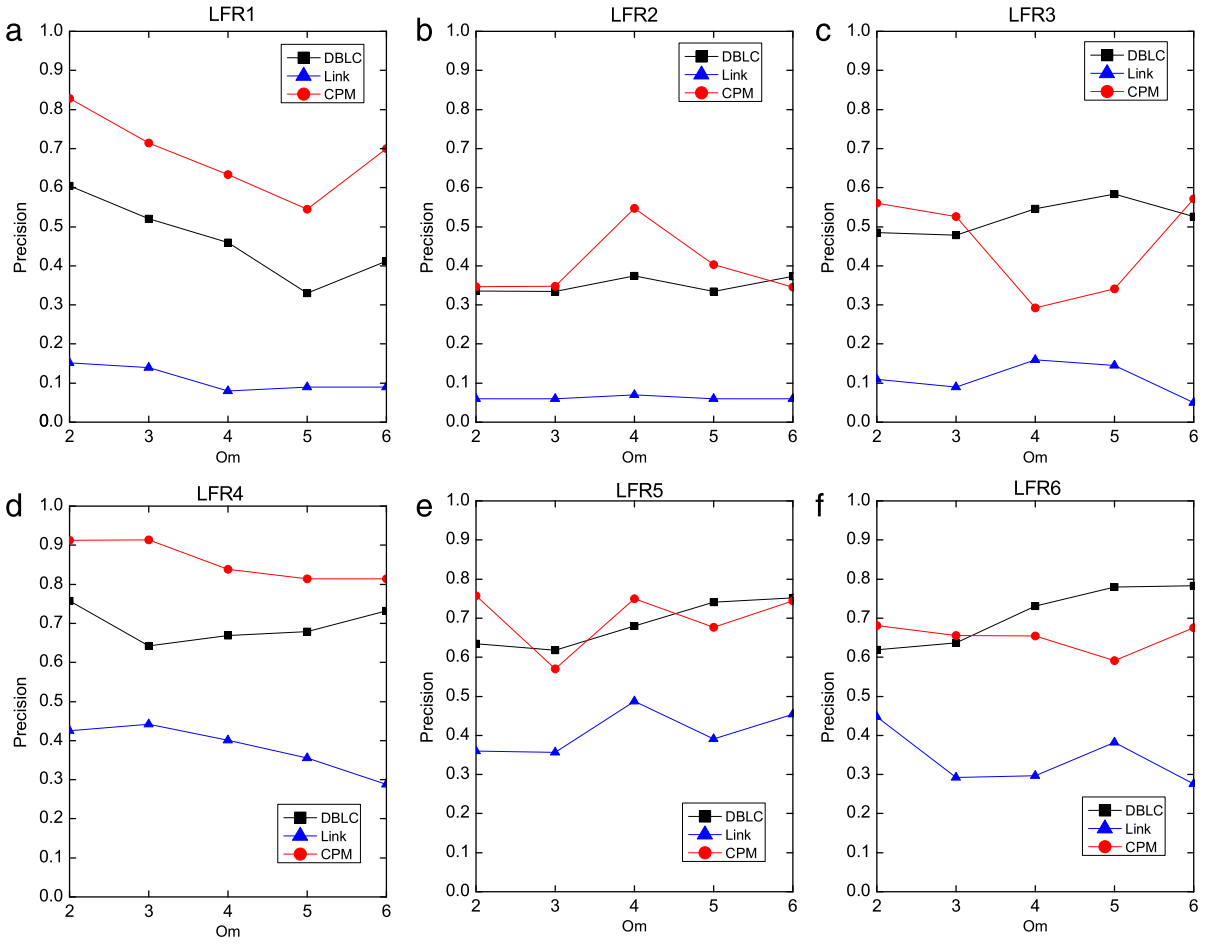

Fig. 6. The comparative precision value of three algorithms on six synthetic networks.

community with large size for the synthetic dataset LFR3 when Om = 4, then the NMI value for community size (20, 100) is larger than that for community size (10, 50), so we see the curve of diamond is not above uptriangle for Link algorithm when Om = 4 in Fig. 5(b).

To test the effects of overlapping nodes on different algorithms, we can compare the NMI value of each algorithm on networks LFR1 and LFR4. Observing the tendency of diamond and star for three algorithms, the diamond is above the star. It demonstrates that the algorithms can get better result when there are less overlapping nodes in the networks.

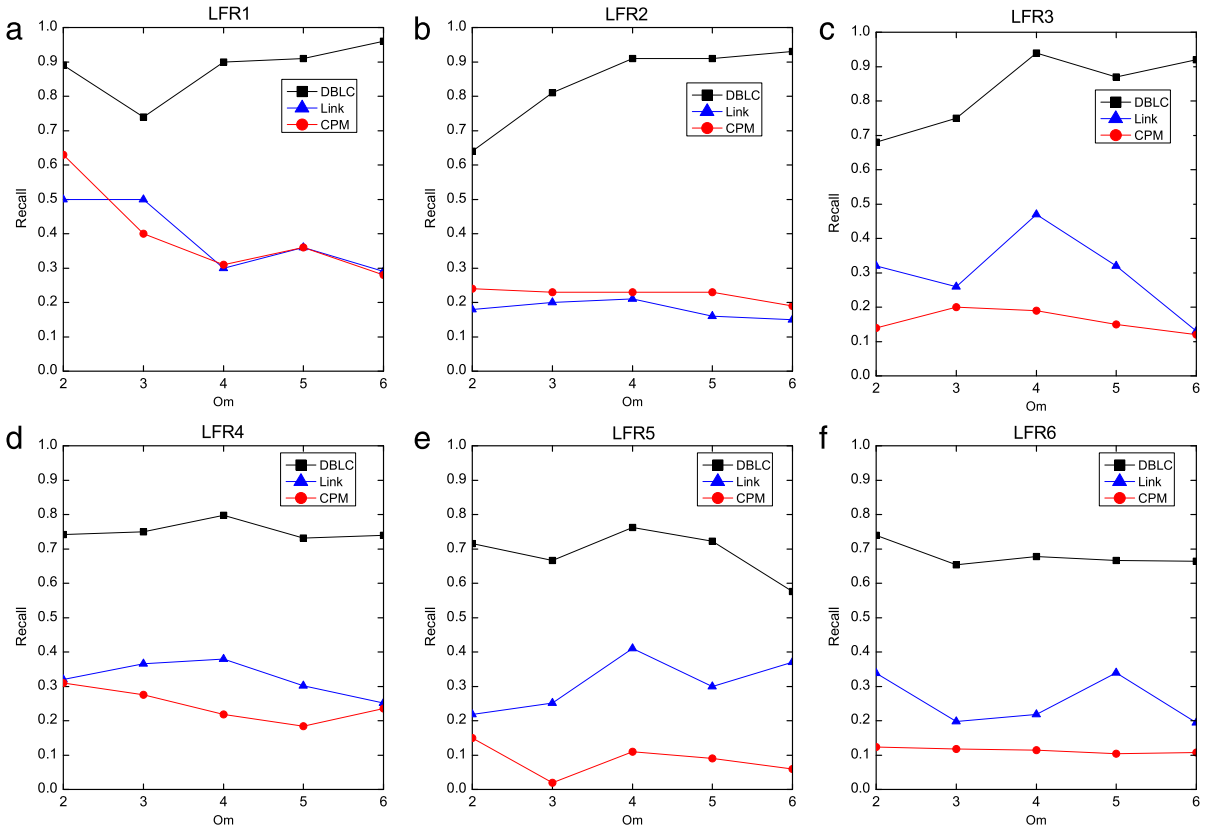

Fig. 7. The comparative recall value of three algorithms on six synthetic networks.

4.2.2. Identifying overlapping nodes

The precision results of three algorithms on LFR1, LFR2, LFR3, LFR4, LFR5 and LFR6 are shown in Fig. 6. It can be seen that CPM always achieves higher precision on some networks compared with other algorithms. This is because CPM always detects a few overlapping nodes which includes a few correctly detected overlapping nodes. For Link algorithm, it detects too many overlapping nodes and includes a few correctly detected overlapping nodes only, so the precision of Link is very low. However, the precision of our algorithm DBLC is in middle level.

The recall results of three algorithms on LFR1, LFR2, LFR3, LFR4, LFR5 and LFR6 are shown in Fig. 7. It is obviously to see that our method can get much higher recall result than other algorithms. It gets nearly 0.9 in LFR1. It means that there are many correctly detected overlapping node in the detected overlapping nodes. CPM does not perform well when the network has more overlapping nodes and large community range size. The performance of link is similar to that of CPM on LFR1 and LFR2.

Fscore results of three methods on different networks are illustrated from Fig. 8(a)–(f) respectively. Our algorithm can get higher Fscore than link and CPM on LFR1 to LFR6 though we get lower precision than CPM algorithm. This is because that Fscore is the harmonic mean of precision and recall, and DBLC gets much higher recall than other algorithms.

To be pointed that, the performance of algorithm in detecting communities is not always coincident with that in detecting overlapping nodes as Ref. [1] shows. For instance, CPM algorithm is superior to Link algorithm at detecting community structure on LFR5 as shown in Fig. 4(e), but Link algorithm performs better at detecting overlapping node aspect on LFR5 as shown in Fig. 8(e). Here, when *Om* is larger, our algorithm can perform better in identifying overlapping nodes.

4.3. Real networks

We analyze the performance of three algorithms on real networks in this section. The description of six real networks is shown in Table 3. The overlapping community structures of the real networks are unknown, so NMI cannot be taken as evaluation measure. Here, the validity of results of the algorithm is determined by the Q_{ov} value. The value of ε on each real network is associated with the similarity between links, and the optimal value of ε and μ is reported in Table 4.

The Q_{ov} value of our algorithm and other comparative algorithms is shown in Table 4. From Table 4, it is found that proposed DBLC algorithm outperforms Link and CPM considerably on all the six datasets. It is worth noticing that our

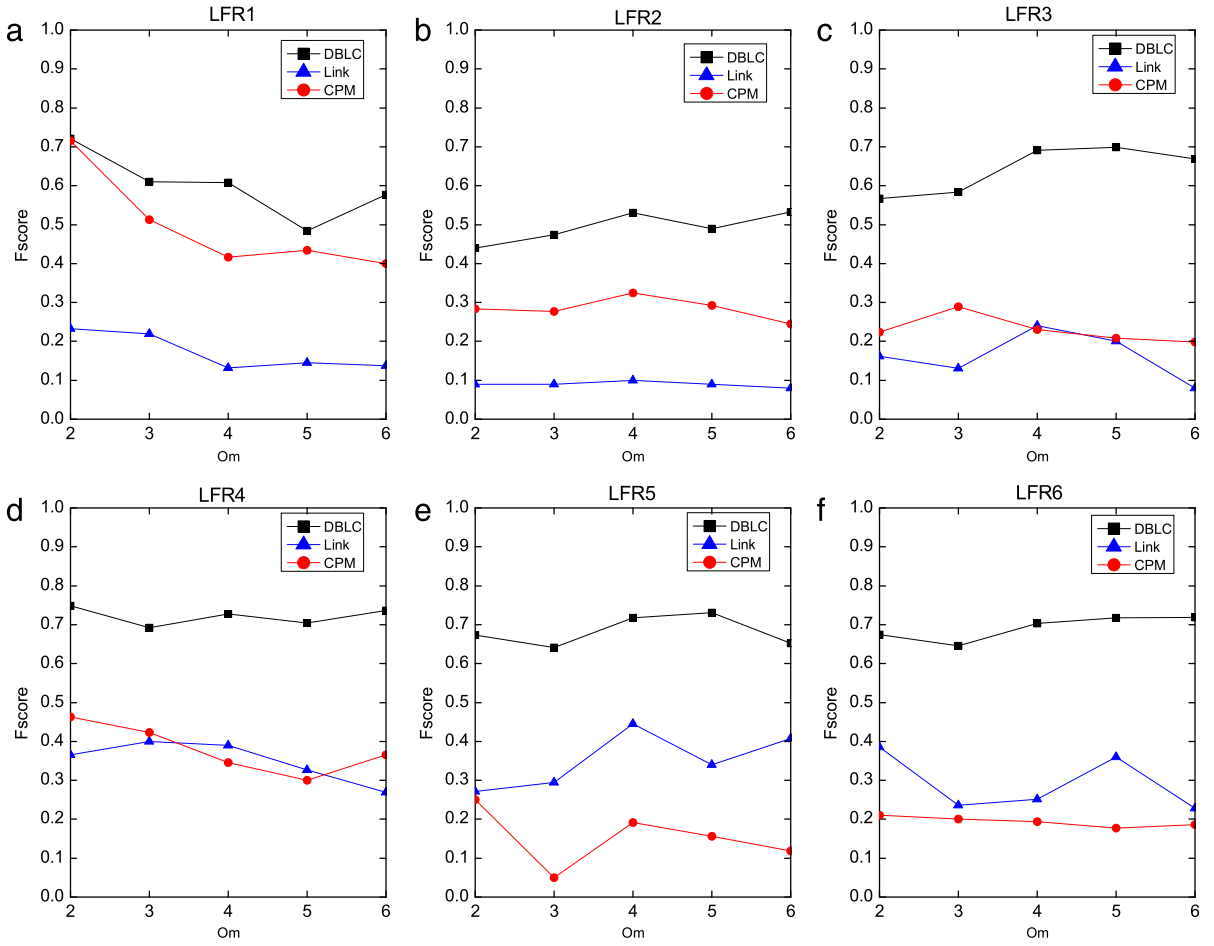

Fig. 8. The comparative Fscore value of three algorithms on six synthetic networks.

Table 3
Summary of real world network.

Network	Node	Edge	Community	Average degree
Karate [24]	34	78	2	4.5
Dolphins [25]	62	159	2	5.1
Books [26]	105	441	3	8.4
Football [27]	115	613	12	10.6
Netscience [28]	379	914	Unknown	4.8
Email [29]	1133	5451	Unknown	9.6

Table 4
Qov value of different algorithms on real world networks.

Datasets	DBLC	ε, μ	Link	CPM	k
Karate	0.686	0.59, 4	0.352	0.484	3
Dolphins	0.766	0.585, 4	0.275	0.593	3
Books	0.834	0.595, 3	0.118	0.758	3
Football	0.669	0.59, 4	0.361	0.589	4
Netscience	0.804	0.595, 4	0.443	0.594	3
Email	0.484	0.569, 6	0.07	0.462	3

algorithm gets Q_{ov} value of 0.766 on dolphins, which is much higher than that of Link and CPM with 0.275 and 0.593 respectively. In conclusion, the results obtained show the capability of density clustering method to effectively deal with community identification in networks.

5. Conclusion and future work

Extracting and understanding community structure in complex network is one of the most intensively investigated problems in recent years. To overcome the issue of bridge edge and border edge mistakenly belonging to adjacent communities, a density based link clustering algorithm for overlapping community detection is proposed. In this study, we propose a new similarity computation formula for assigning similarity between links, and we first achieve a number of rough link clusters based on core density reachable concept during the expansion step. Then a post processing strategy is designed to assign border link to the community it belongs to. It guarantees that the result of clustering is independent on the order where the links are processed during the expansion. Furthermore, a simple parameter selection strategy is introduced to get appropriate parameters. Experiments on synthetic and real life networks show the capability of method to correctly detect communities and overlapping nodes with comparable results with some classical approaches. We would like to try to modify the presented algorithm to handle weighted and directed networks in future.

Acknowledgments

We would like to thank the anonymous referees for their many valuable suggestions and comments. This work was supported by National Nature Science Foundation [61373123, 61572229, U1564211]; Jilin Provincial Science and Technology Development Foundation (20170204074GX); Jilin Provincial International Cooperation Foundation [20150414004GH].

References

[1] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: the state of the art and comparative study, ACM Comput. Surv. 45 (4) (2013) 1–37.
[2] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks, ACM Trans. Inf. Syst. Sec. 10 (4) (2008) 1–26.
[3] C. Liu, Z. Jiang, An improved multi-objective evolutionary algorithm for simultaneously detecting separated and overlapping communities, Nat. Comput. 15 (4) (2016) 635–651.
[4] K. Deng, J. Zhang, J. Yang, Mobile recommendation based on link community detection, Sci. World J. 259156 (2014) 1–13.
[5] S.R.F.Z.X.W. Zhiguo Qu, John Keeney, Multilevel pattern mining architecture for automatic network monitoring in heterogeneous wireless communication networks, China Commun. 13 (7) (2016) 108–116. http://dx.doi.org/10.1109/CC.2016.7559082.
[6] H. Jin, S. Wang, C. Li, Community detection in complex networks by density-based clustering, Physica A 392 (2013) 4606–4618.
[7] M. Gong, J. Liu, L. Ma, Q. Cai, L. Jiao, Novel heuristic density-based method for community detection in networks, Physica A 403 (2014) 71–84.
[8] X. Xu, N. Yuruk, Z. Feng, A.J. Thomas, Scan: A structural clustering algorithm for networks, in: Proceedings of International Conference on Knowledge Discovery and Data Mining, ACM, San Jose, California, USA, 2007, pp. 824–833.
[9] Y.Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multiscale complexity in networks, Nature 466 (7307) (2010) 761–764.
[10] B. Ball, B. Karrer, M.E.J. Newman, Efficient and principled method for detecting communities in networks, Phys. Rev. E 84 (3) (2011) 036103.
[11] Y. Kim, H. Jeong, Map equation for link communities, Phys. Rev. E 84 (2) (2011) 026110.
[12] L. Huang, G. Wang, Y. Wang, E. Blanzieri, Link clustering with extend link similarity and EQ evaluation division, Plos One 8 (6) (2013) e66005.
[13] T. Evans, R. Lambiotte, Line graphs link partitions and overlapping communities, Phys. Rev. E 80 (1) (2009) 016105.
[14] T. Evans, R. Lambiotte, Line graphs of weighted networks for overlapping communities, Eur. Phys. J. B 77 (2) (2010) 265–272.
[15] F. Altunbey, B. Alatas, Overlapping community detection in social networks using parliamentary optimization algorithm, Int. J. Comput. Netw. Appl. 2 (1) (2015) 12–19.
[16] C. Shi, Y. Cai, D. Fu, Y. Dong, B. Wu, A link clustering based overlapping community detection algorithm, Data Knowl. Eng. 87 (2013) 394–404.
[17] C. Pizzuti, Overlapping community detection in complex networks, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, 2009, pp. 859–866.
[18] T.N. Tran, K. Drab, M. Daszykowski, Revised DBSCAN algorithm to cluster data with dense adjacent clusters, Chemometr. Intell. Lab. Syst. 120 (2013) 92–96.
[19] H. Shen, X. Cheng, K. Cai, M. Hu, Detect overlapping and hierarchical community structure in networks, Physica A 388 (8) (2009) 1706–1712.
[20] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.
[21] V. Nicosia, G. mangioni, V. Carchiolo, M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities, J. Stat. Mech. Theory Exp. 29 (3) (2009) P03024.
[22] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 115 (3) (2009) 033015.
[23] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Phys. Rev. E 80 (2009) 016118.
[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: Biol. Sci. 271 (S2) (2003) S186–S188.
[26] V. Krebs, 2014, http://www.orgnet.com.
[27] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.
[28] M. E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.
[29] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E 68 (6) (2003) 065103.