IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 13, 2018, accepted November 20, 2018, date of publication December 3, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2884447

A Community Merger of Optimization Algorithm to Extract Overlapping Communities in Networks

QI LI[1], JIANG ZHONG[1], QING LI[1], CHEN WANG[1], AND ZEHONG CAO[2], (Member, IEEE)
[1]College of Computer Science, Chongqing University, Chongqing 400044, China
[2]Centre for Artificial Intelligence, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia

Corresponding author: Jiang Zhong (zhongjiang@cqu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1402400, in part by the Frontier and Application Foundation Research Program of CQ CSTC under Grant cstc2017jcyjAX0340, in part by the Key Industries Common Key Technologies Innovation Projects of CQ CSTC under Grant cstc2017zdcy-zdyxx0047, in part by the Chongqing Technological Innovation and Application Demonstration Project under Grant cstc2018jszx-cyzdX0086, in part by the Social Undertakings and Livelihood Security Science and Technology Innovation Fund of CQ CSTC under Grant cstc2017shmsA0641, in part by the Fundamental Research Funds for the Central Universities under Grant 2018CDYJSY0055, and in part by the Graduate Research and Innovation Foundation of Chongqing under Grant CYB18058.

ABSTRACT A community in networks is a subset of vertices primarily connecting internal components, yet less connecting to the external vertices. The existing algorithms aim to extract communities of the topological features in networks. However, the edges of practical complex networks involving a weight that represents the tightness degree of connection and robustness, which leads a significant influence on the accuracy of community detection. In our study, we propose an overlapping community detection method based on the seed expansion strategy applying to both the unweighted and the weighted networks, called OCSE. First, it redefines the edge weight and the vertex weight depending on the influence of the network topology and the original edge weight, and then selects the seed vertices and updates the edges weight. Comparisons between OCSE approach and existing community detection methods on synthetic and real-world networks, the results of the experiment show that our proposed approach has the significantly better performance in terms of the accuracy.

INDEX TERMS Overlapping community detection, complex network, weighted network, dense subgraph.

I. INTRODUCTION

Many complex systems in the real world, such as social network, scientific collaboration network and protein interaction network, can be abstracted into complex networks [1]. With the deepening of research on complex networks, it is found that the nature of community structure is often found in many complex networks, which is characterized by relatively close links within communities and relatively sparse links among different communities [2].

The problem of community detection has been addressed in many different fields such as biology, physics, and mathematics [3]. A large number of community detection algorithms have been developed in recent years. These methods can be generally divided into two categories. The first category is non-overlapping community detection method that partition complex network into several independent community structures, where vertices belong to only one of the communities, including clustering algorithm based on graph partitioning [4], [5], clustering algorithm based on spectral analysis [6], [7], clustering algorithm based on hierarchical [8] and density-based clustering algorithm [9], [10]. The second category is overlapping community detection method that allow one vertex to belong to multiple communities at the same time [11]–[13]. For example, humans belong to different social communities, depending on their hobbies, professions, families, etc. There have been many different methods to detect overlapping communities. Clique percolation by Palla *et al.* [14], neighborhood ratio matrix by Eustace *et al.* [15], label propagation by Raghavan *et al.* [16], core-vertex and intimate degree by Wang and Li [17], and subspace decomposition by Eustace *et al.* [12] are some of the popular approaches.

Of all the methods, *Local Expansion* and *Optimization* [18] method is well popularity by researchers. In this method,

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

a particular vertex is first selected called seed vertex as a community, which start to expand from the seed vertex. This expansion is achieved by adding neighborhood vertices to the community. At each step, vertices in the neighborhood of the current community are added according to the value of the scoring function. When the vertices in the neighborhood do not improve the value of scoring function, the expansion will be terminated. The performance of the algorithm is determined by the seed selection and expansion strategy [19]. Our proposed method is based on *local Expansion* and *Optimization*. The main reason is that this method is robust and scalable. The seed expansion is completed independently, multiple seeds can be expanded in parallel, thus greatly accelerating the speed of the algorithm. And *Local Expansion* method only needs local neighborhood information of vertices and does not need global information of network topology. Therefore, the method shows good performance when the entire network is too large to handle. Figure 1 shows Zachary's karate club network.

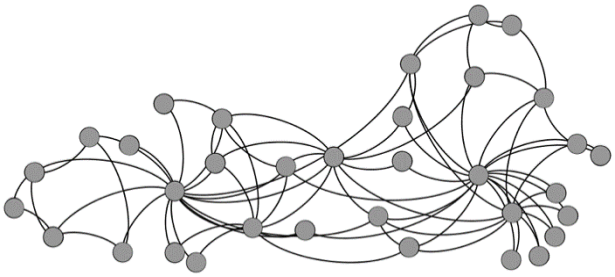


FIGURE 1. Zachary's karate club network.

A large number of meaningful community detection algorithms have been proposed in complex networks, but there still are some challenges that required to be solved. One of the challenges is determination of community boundaries. Some algorithms have their own limitation in terms of applicability and their performance depends on the selection of the seed vertex and expansion strategies. For instance, in some real-world networks where community boundaries are hard to identify, these algorithms fail to identify community boundaries [20]–[22]. Another challenge is: edges or vertices in the real complex networks often contain some vital prior information. For example, the protein interaction network is obtained by high-throughput experiments, and the edge weight often represents the experimental credibility. The edge weight in the cooperative network usually represents the closeness degrees of cooperation between cooperative objects. EM-BOAD [3] has better performance for community detection in weighted networks. Community detection using local communities [20] works well when the seed vertex is at the center of local community [21].

In order to solve the above-mentioned problems, in our study, we propose an overlapping community detection algorithm for complex networks based on the seed expansion strategy (OCSE) applying to both the unweighted and the weighted networks. The algorithm adopts edge-reweighting method to reset the network edges weight depending on the influence of the network topology as well as the original edge weight, and promotes the defuzzification of the network community structure. Then the algorithm adopts vertex-reweighting method to reset the network vertices weight depending on the new edges weight. The rule of seed vertex selection makes the seed vertex is more representative. Combined with the degree of membership, we give a reasonable expansion process. The proposed algorithm successfully detects communities on benchmark networks and real-world networks. The main contributions of our study are summarized as follows:

(1) We propose an edge-reweighting method. More specific, different from the existing approaches [23], [24], we comprehensively consider the network edge information and network topology information to recalculate edges weight, which is to enhance the weight of the community edges further.
(2) Different from the existing seed expansion methods [19], [25], we redefine the weight of vertices in the network according to the new edges weight, select a vertex with the highest weight as community seed vertex, and then iterate to obtain weighted dense subgraphs according to the seed expansion strategy and the weight updating method in linear time.
(3) We improve the content of the degree of membership to measure the link closeness between vertices and weighted core communities, and allocate unclustered vertices to the weighted core communities to obtain more accurate community detection performance.
(4) Our proposed OSCE approach applies to both unweighted and weighted networks. More specific, we report on the experimental outcomes based both real and artificial network datasets, and the results show clearly that our proposed approach has the higher accuracy compared to existing community detection algorithms.

The rest of this paper is organized as follows; Section II, briefly outlines a list of related works and the motive of our research. Detailed steps of the proposed method are presented in Section III. Section IV, presents the experiment results, and comparison with completing algorithms. Finally, this paper is concluded in Section V.

II. RELATIVE WORK

There have been many relevant studies on edge-reweighting method. It can be mainly summed up from two aspects: one is based on the attributes of edges, the edges weight are defined by the attributes of existing edges, such as the betweenness [26], [27]. Another is to redefine the edges weight based on the similarity between vertices, such as the common neighbor ratio and clustering coefficient [28], [29].

Edge-reweighting can be directly applied to the existing community detection algorithms to improve the accuracy of the community detection results [30], [31]. A prior study [32]

uses local information of the network and edge-reweighting method to reconstruct the weighted network. Community detection on this weighted network is more accurate than that on the original network. Another prior work [33] proposed an strategy to detect overlapping communities based on the vertex location. In this method, the community belonging of vertices is determined based on the position of vertices in the topological potential field. The random walk network preprocessing approach [31] was based on random walk. In proposed method, the author thinks that two random walks start from two different vertices in the network, if these two walking behaviors have a high degree of similarity, then the probability of the two vertices in the same community is larger, if these two walking behaviors are obviously different, the probability of the two vertices in the same community is lower. Indeed, Edge-reweighting method is simple and efficient, which can improve the accuracy of community detection results. The main reason is that edge-reweighting method is adopted to improve the weight of the inner edges of the community in the network and reduce the weight of the edges between the communities and make the communities more recognizable.

In real networks, the edge feature can be reflected by the edge weight. For instance, in the social network, the edge weight can be represented as the close relationship between individuals. The edge weight in scientific cooperation network represents the number of cooperation between scientists. In the protein interaction network, the edge weight indicates the credibility of the protein interaction through high-throughput experiments. Due to the importance of edge attributes [34], the edge weight with realistic significance should be considered to detect community. In terms of above reasons, we considered the attributes of the edges and the similarity between the vertices to reweight the edges for community detection. The detailed process is introduced in the following sections.

III. OVERVIEW OF THE PROPOSED METHOD

In this section, we describe the main steps of our approach, as well as illustrate the supports on how OCSE improve the process of finding communities.

A previous study [26] has illustrated that the vertex similarity measures can enhance community detection, but how to combine the vertex similarity with the actual edge weight is our first research point. Focusing on these issues, we design a method of recalculation of edges weight, which is taken into consideration the edges information and topological structure of the network. Then the vertices weight is recalculated according to the new edges weight information. The new vertices weight is more genuinely reflect the importance of vertices in the network. The maximum weight vertex is chosen as the seed vertex. The weighted dense subgraphs are obtained according to the seed expansion strategy and weight update strategy. However, there are number of overlapping vertices in some weighted dense subgraphs, these weighted dense subgraphs with higher overlap rate should be merged into a weighted core community. At the current stage, the dense subgraphs have been found out, but there are still some remaining sparse vertices in the network that are not divided. In order for all vertices to have community ownership, we put forward the concept of the degree of membership to measure the link closeness between vertices and weighted core communities, and then allocate the uncluttered vertices to the corresponding weighted core communities according to the degree of membership. The results of community detection are obtained. The flow-graph of the proposed algorithm is shown in Figure 2 based on the seed expansion strategy, called OSCE.

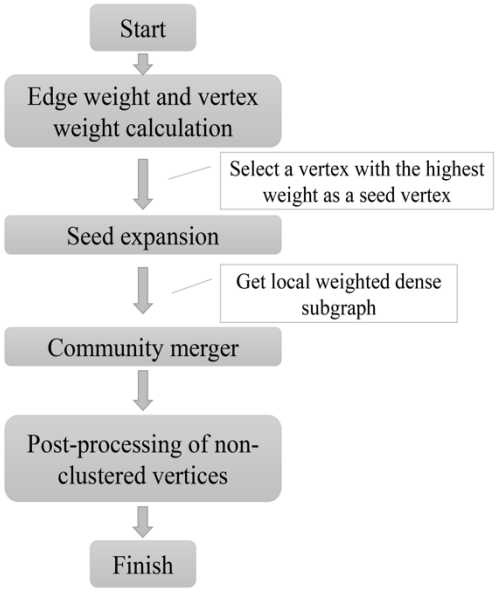


FIGURE 2. The flow-graph of the proposed OCSE algorithm.

A. EDGE-REWEIGHTING

Given a network $G = (V, E)$, the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, and $n = |V|$. Each edge (v_i, v_j) denotes a connection relationship between a pair of vertices v_i and v_j. $m = |E|$., the set of v neighbors ($N_G(v) = \{u|(u, v) \in E\}$) is abbreviated as $N(v)$, and the degree of v is denoted as D_v. In addition, $N_G(U) = \bigcup_{x \in U} N_G(x)$ denotes the subgraph U neighbors in G.

Hub-Promoted similarity measure (HP) [32] is defined as the number of common neighbors dividing by the minimum of the number of two vertices' neighbors, which is proposed for quantifying the topological overlap of pairs of substrates in metabolic networks [26]. The vertices with lower degree determine the similarity value. Furthermore, analogously to HP, Hub-Depressed similarity measure (HD) [35] considers the opposite effect on hubs. It is defined as the number of common neighbors dividing by the maximum of the number of two vertices' neighbors. We introduce these two preference relationships into the networks. In order to measure the connection strength between vertices more accurately,

we need to define the weight between vertices first, as shown in Equation (1).

$$w'(v_i, v_j) = \alpha \times \frac{|N(v_i) \cap N(v_i)|^2}{\min\{|N(v_i)|, |N(v_i)|\}^2} + \beta \times \frac{|N(v_i) \cap N(v_i)|^2}{\max\{|N(v_i)|, |N(v_i)|\}^2} + (1 - \alpha - \beta) \times u(v_i, v_j) \quad (1)$$

Where $N(v_i) \cap N(v_j)$ represents the common neighbors between v_i and v_j. If the edge (v_i, v_j) exists in the network, $u(v_i, v_j)$ is equal to the original edge weight. Conversely, if the edge (v_i, v_j) does not exist, $u(v_i, v_j)$ is equal to zero. The first two items of Equation (1) reflect the similarity degree of v_i and v_j in the network topology. The third item of Equation (1) reflects the original edge weight. Equation (1) thoroughly integrates the network topology and the real edge information. However, for the edge weight, only make sense if there is an edge between two vertices. Therefore, we redefined the edge weight according to Equation (1). In summation, the final edge weight $w(v_i, v_j)$ definition is illustrated in Equation (2). The constant $\varepsilon \in [0, 1]$ is used to distinguish whether there is an edge between vertices in the network.

$$w(v_i, v_j) = \begin{cases} \varepsilon + \dfrac{1 - \varepsilon}{w'_{avg}} \times w'(v_i, v_j), & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases}$$
$$w'_{avg} = \frac{\sum_{(v_i, v_j) \in E} w'(v_i, v_j)}{|E|} \quad (2)$$

B. SEED VERTICES SELECTION

The seed vertices are usually more important than other vertices in the network and lie in comparatively dense areas of the network, which should be lie in the topological centrality of the community. Therefore, the seed vertices should be far apart in the network structure. Based on these principles, the vertex weight is defined as shown in Equation (3).

$$wd(v_i) = \sum_{v_j \in N(v_i)} w(v_i, v_j) \times D_j \quad (3)$$

The vertex weight $wd(v_i)$ is positively related to the degree of v_i neighbors and the weight of adjacent edges of v_i, which reflects the importance of v_i in the network. A vertex with the highest weight is selected as the first seed vertex.

In order to make as many vertices as possible to have their own community in the process of seed expansion. When selecting the seed vertex of the next community, the selection probability of vertices that have been seed vertices should be reduced. Therefore, after finding a dense subgraph C_t, reducing the weight of the edges in C_t can avoid selecting the vertices in C_t as seed vertex with high probability. The new edges weight in C_t is $w(v_i, v_j)/\sqrt{|C_t|}$, and $(v_i, v_j) \in E(C_t)$. If a vertex weight changes too much, it should not be selected as the seed vertex of another dense subgraph. This ensures that the seed vertices are far apart in the network structure. Therefore, we need to define the changing rate of the vertex weight. The changing rate of the vertex weight is defined as shown in Equation (4).

$$rate(v) = 1 - \frac{wd'(v)}{wd(v)} \quad (4)$$

Where $wd(v)$ represents the weight of the vertex v before updating the edge weight. $wd'(v)$ represents the weight of the vertex v after updating the edge weight. If $rate(v) > \theta$, the vertex v should not be selected as the seed vertex again. Figure 3 shows that the seed vertices in the karate club network are obtained by OCSE (black vertices). It can be seen that the number of seed vertices are moderate and they have relatively good representation.

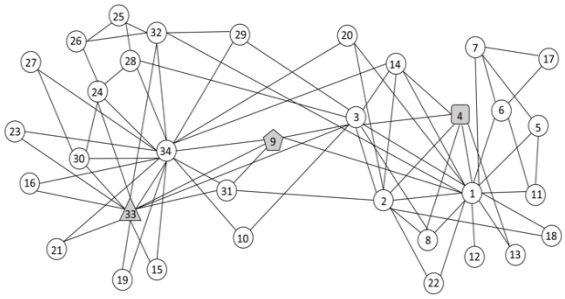


FIGURE 3. Seed vertices (33, 9, 4) in Zachary's karate club network obtained by OCSE.

C. SEED EXPANSION STRATEGY

In order to get a weighted dense subgraph for the current seed vertex. We design an evaluation function to evaluate the dense degree of the subgraph. Assuming S is a connected subgraph of G, V_S represents the vertex set of S, and E_S represents the edge set of S. That is $n_s = |V_S|$ and $m_s = |E_S|$. Add $\frac{n_s(n_s-1)}{2} - m_s$ edges to S to get a complete graph S'. The newly added edges weight is set as the average edge weight of G. The dense degree of S is evaluated by the difference between the weight of the existing edges in S and the newly added edges weight in S'. Equation (5) is the evaluation function for S. The larger the value of $f(s)$, the denser the subgraph S. According to the definition, if there is only one vertex in$S$, then $f(S) = 0$.

$$f(s) = \sum_{(v_i, v_j) \in E_s} w(v_i, v_j) - \frac{n_s(n_s - 1) - 2m_s}{2} \times \frac{\sum_{(v_i, v_j) \in E_s} w(v_i, v_j)}{m} \quad (5)$$

For a vertex $v(v \notin V_s)$, we define a fitness function $\delta_S(v) = f(S \cup \{v\}) - f(S)$. During the seed vertex expansion process, the vertex with the largest value and $\delta_S(v) > 0$ is allocated to the current subgraph S. OCSE executes the above procedure iteratively until no other vertices satisfy the condition and produces a current weighted dense subgraph. Figure 4 shows that OCSE produces weighted dense subgraphs according to the fitness function. Different shapes

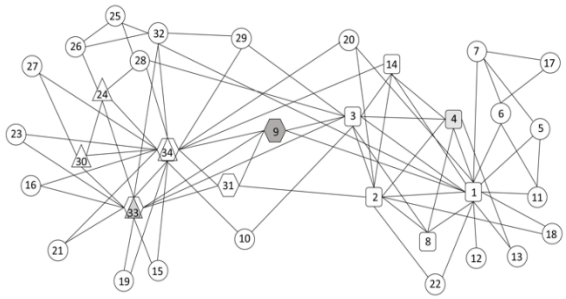


FIGURE 4. Weighted dense subgraphs in Zachary's karate club obtained by OCSE.

represent different dense subgraphs (triangle vertices, hexagonal vertices and quadrilateral vertices).

D. COMMUNITY MERGER

After completing the above steps, OCSE can find many dense subgraphs. However, there may be many overlap vertices between some dense subgraphs. These dense subgraphs should be merged. Figure 4 shows that there are half vertices overlapped between the dense subgraph consisting of vertices {9, 31, 33, 34} (hexagonal vertices) and the dense subgraph consisting of vertices {24, 30, 33, 34} (triangular vertices). These two subgraphs should be merged. In this paper, the condition for merging of two dense subgraphs is that the number of overlap vertices between two subgraphs is no less than half the number of a smaller dense subgraph. The communities after merging are called weighted core communities. Figure 5 shows that OCSE obtains the weighted core communities in Zachary's karate club.

E. POST-PROCESSING OF UNCLUSTERED VERTICES

The weighted core communities are obtained by OCSE, which includes some local dense subgraphs. However, there are still some unclustered vertices (shown as the round vertices in Figure 5). Therefore, we first define the degree of membership to measure the link closeness of unclustered vertices and the weighted core communities. Equation (6) shows the degree of membership of the vertex v to the weighted core community C. The value of $b(v, C)$ is positively related to the adjacent edges weight of v and the adjacent vertices weight of v in C. Select a weighted core community with the maximum value and allocate v to the weighted core community. The algorithm iteratively allocates the unclustered vertices to the existing weighted core communities according to Equation (6) until all the unclustered vertices are allocated. Figure 6 shows the final clustering result obtained by OCSE in Zachary's karate club.

$$b(v, C) = \sum_{v_x \in N(v) \cap V_C} w(v_x, v) + \sum_{v_x \in N(v) \cap V_C} wd(v_x) \quad (6)$$

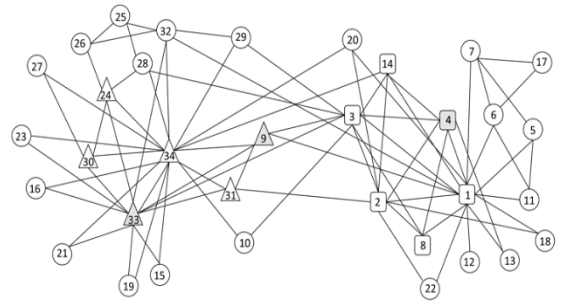


FIGURE 5. Weighted core communities in Zachary's karate club obtained by OCSE.

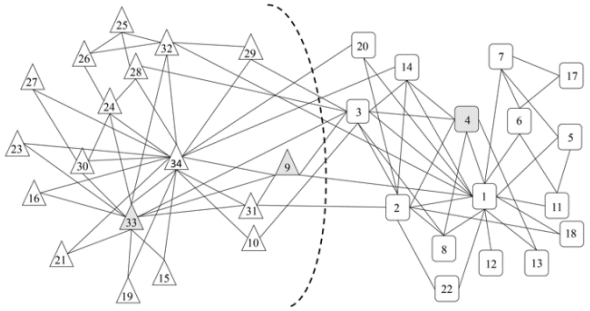


FIGURE 6. Final communities in Zachary's karate club obtained by OCSE.

F. ALGORITHM COMPLEXITY ANALYSIS

Algorithm 1-3 describe the procedure of OCSE in details. *Step 1-4* of Algorithm 1 initializes vertices and edges weight in the network. The average time complexity is proportional to the total number of edges. It takes at most $O(c \times |E|)$. *Step 5-23* of Algorithm 1 obtains the weighted dense communities, and all edges in the network are traversed in each iteration. The time complexity of this phase depends on the number of iterations. Assuming that the number of iterations is t, the total time is $O(t \times E)$. Algorithm 2 is the merging operation for the weighted dense communities. Assuming there are S ($S \ll |V|$) weighted core communities, And its time complexity is $O(S^2)$. Algorithm 3 is the process of allocating unclustered vertices and the maximum time complexity is $O(|V|)$. Summarizing the above analysis, the total average time complexity of OCSE is $O(c \times t \times |E| + |V|)$, c and t are constants.

IV. EXPERIMENTAL

We analyze and test the performance of the OSCE algorithm and competing algorithms on several real and synthetic networks. The first experiment is to test our algorithm on synthetic networks, and the second experiment is to test our algorithm on real-world networks. Furthermore, we also applied the same networks in the competing algorithms, including NLA [33], MCMOEA [36], COPRA [37], CPM [14], BASH [38], IEDC [39], ACC [40] and Li *et al.* [13]. These competing algorithms listed in TABLE I. Note that the time

Algorithm 1 Seed Expansion Process

Input: A network $G = (V, E, U)$. V and E represent the set of the vertices and edges respectively. U denotes the weight matrix of G

Output: Set of weighted dense subgraphs $\{C1, C2, \ldots, Ct\}$

Initialize: $n = |V|$, $m = |E|$, $t = 0$, $F = V$
if $\alpha \geqslant 0$ **then**
  $C = \emptyset$, $\beta = 0.7\text{-}\alpha$
  calculate the weight matrix of each edge $W_{n\times n}(G)$ and the vertices weight $wd(v)$ in F
  if $F \neq \emptyset$ **then**
    $t = t + 1$, $Ct = \emptyset$
    select a vertex v with the highest weight in F
    $F = F\text{-}\{v\}$, $Ct = Ct \cup \{v\}$
    if $\delta_{Ct}(x) = \max_{v\in N(Ct)} \delta_{Ct}(v)$ and $\delta_{Ct}(x) > 0$ **then**
      $Ct = Ct \cup \{x\}$, go to ***step 9***
    end if
    if $|Ct| \leq 3$ **then**
      $t = t - 1$, go to ***step 5***
    end if
    update the vertices weight in Ct according to formula (6)
    for each vertex $x \in Ct$ **do**
      if $\mathrm{cr}(x) \leq \theta$ **then**
        $F = F\text{-}\{x\}$
      end if
    end for
    go to ***step 5***
  end if
end if
return $\{C1, C2, \ldots, Ct\}$

Algorithm 2 Community Merger Process

Input: Set of weighted dense subgraphs $\{C1, C2, \ldots, Ct\}$

Output: Set of weighted core communities $\{C1', C2', \ldots, Cl'\}$

for each weighted dense subgraph $Ci \in \{C1, \ldots, Ct\}$ **do**
  if $|Ci \cap Cj|/\min\{|Ci|, |Cj|\} \geqslant 0.5$ and $i \neq j$ **then**
    $Ci = Ci \cup Cj$
  end if
end for
return $\{C1', C2', \ldots, Cl'\}$

Algorithm 3 Post-processing of Unclustered Vertices

Input: Set of weighted core communities $\{C1', C2', \ldots, Cl'\}$

Output: result of community detection

Initialize: $T = V - \bigcup_{i=1}^{l} C_i'$, $\xi(C) = \emptyset$
repeat
  for each vertex v in T **do**
    for each core community $\{C1', \ldots, Cl'\}$ **do**
      calculate $b(v, Cl')$
      $\xi(C) \leftarrow b(v, Cl')$
    end for
    Select the maximum value of $\xi(C)$ and its corresponding community set is C_j
    $C_j = C_j \cup \{v\}$
    $\xi(C) = \emptyset$
  end for
until $T \neq \emptyset$
return result of community detection

TABLE 1. Algorithms included in the experiments.

Algorithm	Complexity	Reference
NLA	$O(n^2)$	[33]
MCMOEA	$O(\max\{nD_{\max}^3, m^2 \times PS, m \times PS \times gen_{\max}\})$	[36]
COPRA	$O(vm\log(vm/n))$	[37]
CPM	$O(\alpha n^{\beta \ln(n)})$	[38]
BASH	$O(n^2)$	[39]
IEDC	$O(n^2)$	[40]
ACC	$O(n^2)$	[41]

TABLE 2. Settings of important parameters.

Parameters	Description	Values
ε	distinguish whether there is an edge between a pair of vertices	0.2
θ	changing rate of the vertex weight	0.3
α	balance coefficient	$\frac{2m}{n(n-1)}$
β	balance coefficient	$0.7 - \frac{2m}{n(n-1)}$

complexity given is for the worst case. α, β and v are constant. The maximum vertex degree in G is D_{max}, PS is the size of population and gen_{max} is the maximum number of generations. The hardware environment of the experiment is a PC with a stand-alone Intel Xeon processor with 2.67GHz and 32G memory. The software platform is python 2.7 in Windows. The parameters setting of OCSE applied in both experiments are given in TABLE II, and are determined by a preliminary experiment using a small number of graph instances [41].

A. EVALUATION METRICS

The detected communities is $C = \{C_1, \ldots, C_s\}$. The known communities is $O = \{O_1, \ldots, O_t\}$. Many methods have been put forward to measure the effect of community detection, but only a few measures are suitable for overlapping communities. In our study, we introduce three well-known evaluation criteria to measure the performance of the applied algorithms, Average F1 Score (F1-score) [42], Normalized Mutual Information (NMI) [43] and Overlapping Modularity (Q_{ov}) [44].

NMI is a useful information measure in information theory. It is introduced by Leon Danon et al. and used to measure the similarity between the detected communities and the known communities. The formula is as follows:

$$I(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{O_B}\log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_{i=1}^{C_A}Ni.\log(\frac{N_{i.}}{N})+\sum_{i=1}^{O_B}N_{.j}\log(\frac{N_j}{N})} \tag{7}$$

Where C_A is the number of known communities, O_B is the number of detected communities, $N_{i.}$ represents the sum of the elements of the ith row in matrix N, and $N_{.j}$ represents the sum of the elements of the jth column in matrix N. NMI is a good evaluation index for the network with known communities. The larger the value of the function, the more similar the detected communities is to the known communities, that is, the higher the quality community detection. When the result of community detection is consistent with the real community, the value of the function is 1.

F1-score measures the correctly classified members in each community based on the ground-truth information.

$$\frac{1}{2}\left(\frac{1}{|C|}\sum_{C_i\in C}F1(C_i,O_{g(i)})+\frac{1}{|O|}\sum_{O_i\in O}F1(C_{g'(i)},O_i)\right) \tag{8}$$

Where the best matching g and g' is defined as follows:

$$\begin{aligned} g(i)&=arg\max_j F1(C_i,O_j)\\ g'(i)&=arg\max_j F1(C_j,O_i) \end{aligned} \tag{9}$$

Where $F1(C_i, O_j)$ is the harmonic mean of Precision and Recall. It is defined as follows.

$$F1=\frac{2\times precision\times recall}{precision+recall} \tag{10}$$

The core idea of Q function [45] is that if there are more edges in a subgraph than in random network, some community structures are exist in the subgraph. In order to evaluate the quality of overlapping community detection. Nicosia *et al.* [44] proposed Q_{ov} function. It is defined as follows.

$$\begin{aligned} Q_{ov}&=\frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}[\beta_{l(i,j),c}A_{ij}-\frac{\beta^{out}_{l(i,j),c}k_i^{out}\beta^{in}_{l(i,j),c}k_j^{in}}{m}]\\ \beta^{out}_{l(i,j),c}&=\frac{\sum_{j\in V}F(\alpha_{i,c},\alpha_{j,c})}{|V|}\\ \beta^{in}_{l(i,j),c}&=\frac{\sum_{i\in V}F(\alpha_{i,c},\alpha_{j,c})}{|V|} \end{aligned} \tag{11}$$

The value of Q_{ov} function is closely related to $F(\alpha_{i,c}, \alpha_{j,c})$. In Nicosia' experiment, he adopted the following definition.

$$F(\alpha_{i,c},\alpha_{j,c})=\frac{1}{\left(1+e^{-f(\alpha_{i,c})}\right)\left(1+e^{-f(\alpha_{j,c})}\right)} \tag{12}$$

Where $f(\alpha_{i,c})$ is a simple linear function. $f(x)=2Px-P$, $P\in R$, according to the article [44], we set P equal to 30. If all vertices belong to a community, the value of Q_{ov} function is 0. Generally speaking, the higher the value of Q_{ov}, the higher the quality of overlapping community structure.

B. TEST ON SYNTHETIC NETWORKS

In this section, we use the LFR (Lancichinetti-Fortunato-Radicchi) benchmark [46] proposed by Lancichinetti and Fortunato to produce overlapping artificial networks. The LFR benchmark contains dozens of adjustable parameters. The detailed explanation of each parameter is shown in TABLE III.

TABLE 3. Description of LFR benchmark parameters.

Parameter	Description
N	Number of vertices
k	Average degree
$maxk$	Maximum degree
μ	Mixing parameter
$t1$	Power law distribution of vertex scale
$t2$	Power law distribution of community size
$minc$	Minimum number of vertices in the community
$maxc$	Number of vertices in the largest community
O_n	Number of overlapping vertices in the community
O_m	Number of overlapping vertices linking communities

The most significant impact on the LFR benchmark networks are the mixing parameters μ, the number of overlapping vertices in the community O_n, and the number of overlapping vertices linking communities O_m. We mainly study the performance of different algorithms under these three parameters, and set the average degree of the artificial networks to 10, according to the literature [47], this figure is close to the average degree of most real networks. The maximum degree (*mark*) is set to 50, and the community size is in the range of [20, 50]. Considering the stochastic factor, we run five times for each algorithm and take the average of these values as the results.

We first test the NMF and F1 value of different algorithms when O_m increases from 2 to 8. The results are shown in Figure 6. With the increase of O_m, the number of overlapping vertices linking communities is continuously increasing, which increases the difficulty of detecting overlapping communities. As seen in Figure 7, we can see that ACC, MCMOEA, Ref. [13] and OCSE show better performance, which have strong robustness. CPM, NLA, BASH and COPRA are more sensitive to parameter O_m, which lead to the results of volatility on the low side.

Figure 8 shows that the NMF and F1 value of different algorithms when μ increases from 0 to 0.45. μ is the mixing parameter which represents each vertex shares a fraction $1-\mu$ of its links with the other vertices of its community and a fraction μ with the other vertices of the network. We set the

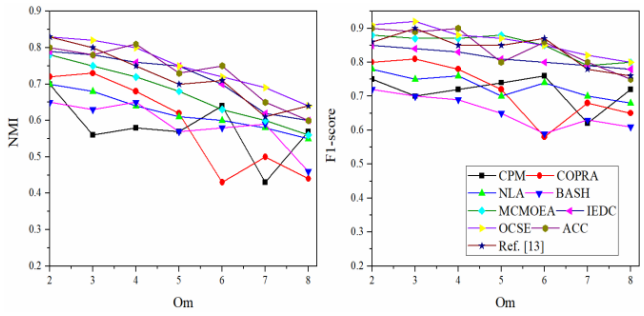


FIGURE 7. NMI (left) and F1-score (right) for networks with different O_m.

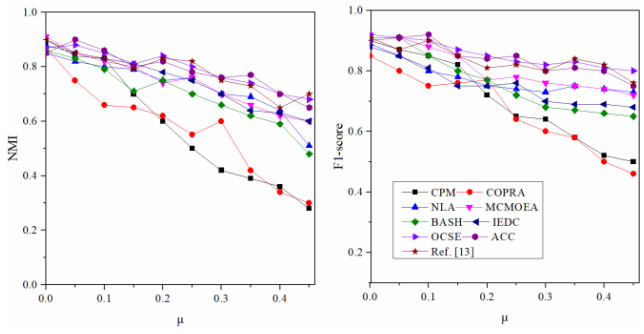


FIGURE 8. NMI (left) and F1-score (right) for networks with different μ.

range of μ to [0, 0.45]. When μ exceeds 0.5, the generated community structure in the network is difficult to reflect and it is more similar to the random network. When $\mu = 0$, there is no additional noise in the artificially synthesized network. It can be observed that all the algorithms perform well, and both the NMI and F1 value are close to one. With the increase of μ and the result in the performance degradation of all algorithms. But the performance of OSCE slows down, even when $\mu = 0.45$, it still maintains higher NMI value and F1 value. The NMI and F1 value of OSCE decrease slowly with the decline of community quality, which shows the effectiveness and robustness of OSCE. In addition, the ACC algorithm also shows better performance.

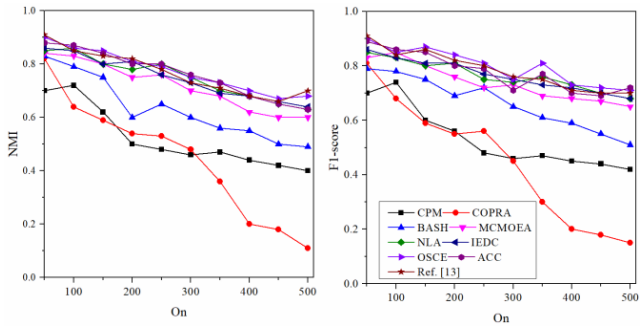


FIGURE 9. NMI (left) and F1-score (right) for networks with different O_n.

Figure 9 shows that the NMF and F1 value of different algorithms when O_n increases from 50 to 500. O_n represents the number of overlapping vertices in the community. All algorithms efficiency are decreased with the increase of the number of overlapping vertices. However, it can be seen that the performance of OSCE is superior to other algorithms, the worst is COPRA.

Based on the above results, it can be seen that although the proposed method does not perform best on different types of networks, overall, compared with the other eight algorithms, it shows good performance.

C. TEST ON REAL-WORLD NETWORKS

We use the standard networks [41], [48] of research on community detection to analyze the performance of OSCE. These datasets are described in detail in TABLE IV.

TABLE 4. Description of real-world networks.

Date sets	Vertex number	Edge number	Average degree
Karate	34	78	4.6
Dolphins	62	159	5.1
Football	115	613	10.7
Polbooks	105	441	8.4
blogs	1224	9250	15.1
Email	1133	5451	9.62
Lemis	34	78	4.59
jazz	3198	5484	3.43
Neural	297	2345	15.8
power	4941	6591	2.67

1) ZACHARY'S KARATE CLUB

The network is a classical dataset in the field of social network analysis. Sociologist Zachary spent two years observing the social relationships among 34 members of a karate club in an American university. Based on the association of these members in the club and outside, he construct a social network of members, which consists of 34 vertices. If there is an edge between two vertices, which means that the two members are friends who have frequent contacts.

2) COLLEGE FOOTBALL NETWORK

The network consists of 115 vertices with 12 communities. Each vertex represents a college team that participated in the 2000 U.S. football season, and the edge between the two vertices indicates that at least one match was played between the two teams.

3) DOLPHINS SOCIAL NETWORK

D. Lusseau and others studied social relationship between 62 bottlenose dolphins in Doubtful Sound Strait of New Zealand. Each vertex in the network represents a dolphin. Each edge represents the frequent contact between two dolphins.

4) BLOGS NETWORK

Adamic and Glance recorded hyperlink relationships of Weblog in 2005. We removed the isolated single vertices, which can reduce noise.

5) POLL BOOKS NETWORK

Amazon's book sales record is related to American politics. Each vertex in the network represents a book sold on Amazon, and the edge represents the two books have been bought at the same time by buyers.

6) EMAIL NETWORK

The mail exchange relationship between the university students. Each vertex in the network represents an email account, and each edge represents the relationship between the sending and receiving emails.

7) LEMIS NETWORK

The data set is the relationship between the characters that is summarized by Hugo's Les Miserables. Each vertex represents a character in the novel. If multiple roles appear in a scene at the same time, there will be an edge between them.

8) JAZZ NETWORK

A network of relationships between jazz musicians, the vertices in the network are musicians, and the edge represents the relationship of common playing.

9) NEURAL NETWORK

A directed, weighted network representing the neural network of nematode.

10) POWER GRID

An unweighted network representing the topology of the western states power grid of the United States.

Considering the stochastic factor, we run 5 times for each algorithm and recorded the mean values in our results. The results are shown in Figure 9 – Figure 11.

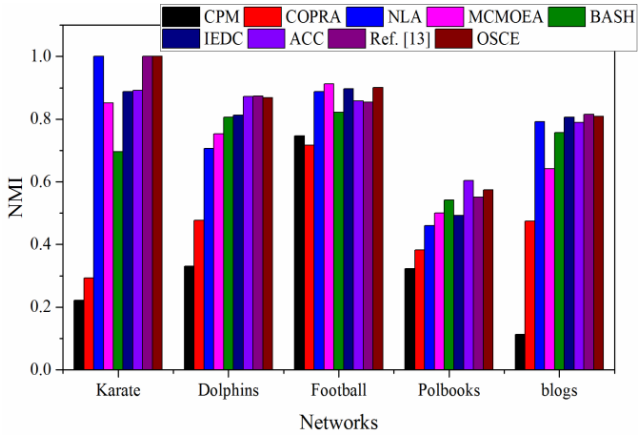


FIGURE 10. NMI on real-world networks with ground truth.

As seen in Figure 10 and Figure 11, The OSCE algorithm performs better than other algorithms on Karate and Football datasets, and the communities obtained are very close to the real situation. The NMI value on Karate is even stable at one. The performance on the blogs dataset is second, and is very close to the result of the best performance algorithm (Ref. [13]). For real-world networks without ground truth, we use overlapping modularity (Q_{ov}) to evaluate the results of community detection. As seen in Figure 12, the OSCE model on all five data sets shows the relatively good performance.

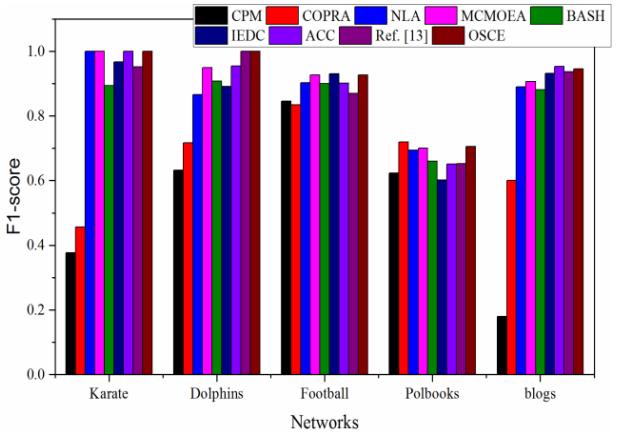


FIGURE 11. F1-score on real-world networks with ground truth.

Overall, OCSE considers the network topology and the original weight information in the community detection process. It can produce results that are close to the real network, and exhibit better performance.

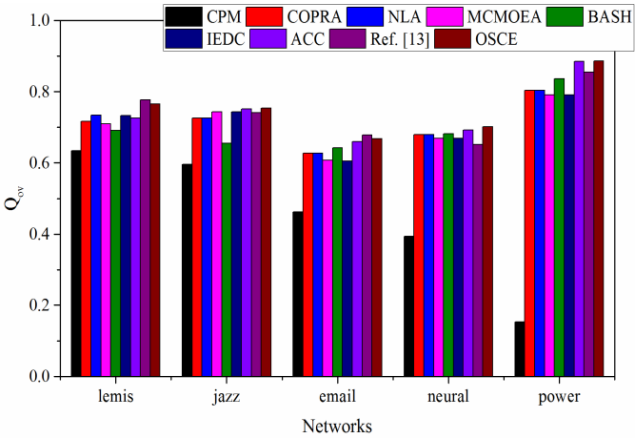


FIGURE 12. Q_{ov} on real-world networks without ground truth.

D. SIMILARITY MEASURES

This section mainly verifies that our measure method of HP combined with HD is superior to other similarity measures. For the method we proposed, the structural similarity is mainly reflected in the first two items of Equation (1). We replace the first two items of the Equation (1) into other similarity measures. The experiment is carried out on the real world networks. We introduce 7 typical similarity measures and list the definition of the weight between vertices based on these similarity measures separately.

1) Common Neighbor similarity measure (CN) [35]: It is the simplest and classical measure index of local information. It only considers the number of common neighbors between two vertices, and ignores the differences of common neighbors and the relationships between common neighbors. For Common Neighbor similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = |N(v_i) \cap N(v_j)| + u(v_i, v_j) \tag{13}$$

2) Salton similarity measure (Salton) [49]: The Salton algorithm adds the information of the degree of two vertices on the basis of the common neighbor similarity. It considers that the similarity between vertices is directly proportional to the number of common neighbors and inversely proportional to the degree of vertices. Therefore, the Salton algorithm is also called the prior similarity algorithm. For Salton similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{D_i \times D_j}} + u(v_i, v_j) \tag{14}$$

3) Jaccard similarity measure (Jaccard) [50]: It considers that the similarity between vertices is proportional to the proportion of the number of common neighbors to the total number of all their neighbors. That is to say, compared with the vertices with more neighbors, the vertices with fewer neighbors are more vulnerable to neighbors. For Jaccard similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|} + u(v_i, v_j) \tag{15}$$

4) Leicht Holmem Newman similarity measure (LHN) [51]. It is similar to the Salton algorithm. The algorithm is mainly to reduce the influence of vertex degree on similarity. For Leicht Holme Newman similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|D_i \cup D_j|} + u(v_i, v_j) \tag{16}$$

5) Preferential Attachment similarity measure (PA) [35]: The PA algorithm removes the influence of common neighbor vertices. The probability that it defines a link is directly proportional to the degree of the vertex. That is, it considers that the more vertices that the vertex linked, the easier it is to do the link-generating behavior. The network information used by the algorithm is only the degree of the vertex, so it has better efficiency, and more suitable for large-scale network data. For Preferential-Attachment similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = D_i \times D_j + u(v_i, v_j) \tag{17}$$

6) Hub-Promoted similarity measure (HP) [28]: Compared with the Salton algorithm, the numerator of HP function is also the number of common neighbors of two vertices, but its denominator is a smaller degree value of the two vertices degree. This improvement makes the algorithm better applied in the metabolic network. The vertices with lower degree determine the similarity value. For Hub-Promoted similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\min\{D_i, D_j\}} + u(v_i, v_j) \tag{18}$$

7) Hub-Depressed similarity measure (HD) [31] : The HD algorithm is similar to HP, but the denominator of HD function is opposite to HP. It considers that the vertex with large degree has a greater influence on the similarity value. For Hub-Depressed similarity measure, the definition of the weight between vertices is as follows.

$$w'(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\max\{D_i, D_j\}} + u(v_i, v_j) \tag{19}$$

The experimental results are shown in Figure 13 and Figure 14. As shown the structural similarity measures

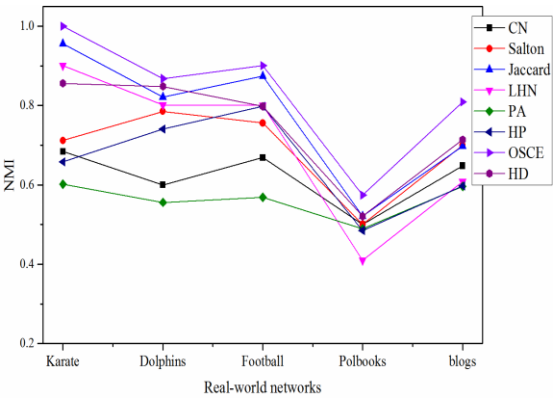


FIGURE 13. NMI on real-world networks, obtained by our community detection methods with different structural similarity measures.

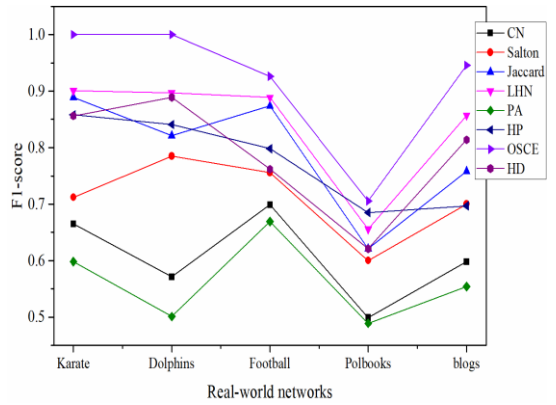


FIGURE 14. F1-score on real-world networks, obtained by our community detection methods with different structural similarity measures.

proposed in this paper, it obviously offers better performance than other similarity measures. The result of using PA is the worst. When HP and HD are used separately, its performance is general and some results less than the other similarity measures. However when using use a combination of HP and HD, its performance is apparently improved.

V. CONCLUSION

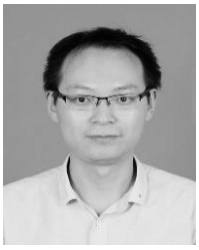
The community reflects the local characteristics of individual behavior in the network and relationships. The study of the community in the network plays a vital role in understanding the structure and function of the whole network, and can help us analyze and predict the interaction between the elements of the network. In fact, community detection plays an important role in the analysis of metabolic network, analysis of genetic regulatory networks and master gene recognition. For example, the complex and community modules in the predicted protein interaction network are important for understanding the organization and function of the biological system and the prediction of the function of the unknown protein.

In this study, we aim to develop a community detection method that can extract the overlapping community structure of real-world networks. In specific, based on the existing studies on community detection only considering the network topology or the edge weight, we developed an overlapping community detection algorithm based on the seed expansion strategy (OCSE), which is a conceptual model of network community structure according to the edge information and the network topology information. The experimental findings showed that OSCE can get a better performance in terms of accuracy compared to existing algorithms.

REFERENCES

[1] Z. Zhao, S. Zheng, J. Sun, L. Chang, and F. Chiclana, "A comparative study on community detection methods in complex networks," *J. Intell., Fuzzy Syst.*, vol. 35, no. 1, pp. 1077–1086, 2018.
[2] X. Wang and X. Qin, "Asymmetric intimacy and algorithm for detecting communities in bipartite networks," *Phys. A, Stat. Mech., Appl.*, vol. 462, pp. 569–578, Nov. 2016.
[3] J. Li, X. Wang, and J. Eustace, "Detecting overlapping communities by seed community in weighted complex networks," *Phys. A, Stat. Mech., Appl.*, vol. 392, no. 23, pp. 6125–6134, 2013.
[4] T. Heuer and S. Schlag, "Improving coarsening schemes for hypergraph partitioning by exploiting community structure," in *Proc. Int. Symp. Exp. Algorithms*, 2017, pp. 21:1–21:19.
[5] Y. Wang, H. Huang, C. Feng, and Z. Liu, "Community detection based on minimum-cut graph partitioning," in *Proc. Int. Conf. Web-Age Inf. Manage.*, 2015, pp. 57–69.
[6] J. Cheng, L. Li, M. Leng, W. Lu, Y. Yao, and X. Chen, "A divisive spectral method for network community detection," *J. Stat. Mech. Theory, Exp.*, vol. 2016, p. 033403, Mar. 2016.
[7] Z. Zhou and A. A. Amini, "Analysis of spectral clustering algorithms for community detection: The general bipartite setting," to be published.
[8] B. Yang, J. Di, J. Liu, and D. Liu, "Hierarchical community detection with applications to real-world network analysis," *Data, Knowl. Eng.*, vol. 83, pp. 20–38, Jan. 2013.
[9] T. Falkowski, A. Barth, and M. Spiliopoulou, "DENGRAPH: A density-based community detection algorithm," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Nov. 2007, pp. 112–115.
[10] F. Meng, F. Zhang, M. Zhu, Y. Xing, Z. Wang, and J. Shi, "Incremental density-based link clustering algorithm for community detection in dynamic networks," *Math. Problems Eng.*, vol. 2016, Dec. 2016, Art. no. 1873504.
[11] J. Eustace, X. Wang, and Y. Cui, "Community detection using local neighborhood in complex networks," *Phys. A Stat. Mech., Appl.*, vol. 436, pp. 665–677, Oct. 2015.
[12] J. Eustace, X. Wang, and J. Li, "Approximating Web communities using subspace decomposition," *Knowl.-Based Syst.*, vol. 70, pp. 118–127, Nov. 2014.
[13] J. Li, X. Wang, and Y. Cui, "Uncovering the overlapping community structure of complex networks by maximal cliques," *Phys. A, Stat. Mech., Appl.*, vol. 415, pp. 398–406, Dec. 2014.
[14] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
[15] J. Eustace, X. Wang, and Y. Cui, "Overlapping community detection using neighborhood ratio matrix," *Phys. A, Stat. Mech. Appl.*, vol. 421, pp. 510–521, Mar. 2015.
[16] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, p. 036106, 2007.
[17] X. Wang and J. Li, "Detecting communities by the core-vertex and intimate degree in complex networks," *Phys. A, Stat. Mech., Appl.*, vol. 392, no. 10, pp. 2555–2563, 2013.
[18] Q. Wang and C. Zhang, "Improvement of overlapping community detection based on local expansion and optimization," *Appl. Res. Comput.*, to be published.
[19] P. Deshpande and B. Ravindran, "MCEIL: An improved scoring function for overlapping community detection using seed expansion methods," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2017, pp. 652–659.
[20] J. P. Bagrow and E. M. Bollt, "Local method for detecting communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 4, p. 046108, 2005.
[21] A. Clauset, "Finding local community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, p. 026132, 2005.
[22] J. Huang, H. Sun, Y. Liu, Q. Song, and T. Weninger, "Towards online multiresolution community detection in large-scale networks," *PLoS ONE*, vol. 6, no. 8, p. e23829, 2011.
[23] Z. Wang, Y. Zhao, Z. Chen, and Q. Niu, "An improved topology-potential-based community detection algorithm for complex network," *Sci. World J.*, vol. 2014, Jan. 2014, Art. no. 121609.
[24] W. Zhao, F. Zhang, and J. Liu, *Local Community Detection Via Edge Weighting*. Springer, 2016.
[25] B. Wang, H. B. Zheng, Y. J. Fang, and J. J. Wei, "Overlapping community detection algorithm based on seed diffusion," *Appl. Mech., Mater.*, vols. 556–562, pp. 3300–3304, May 2014.
[26] J. Xiang *et al.*, "Enhancing community detection by using local structural information," *J. Stat. Mech., Theory Exp.*, vol. 2016, p. 033405, Mar. 2016.
[27] A. Khadivi, R. A. Ajdari, and M. Hasler, "Network community-detection enhancement by proper weighting," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 4, p. 046104, 2011.
[28] S. Asur, D. Ucar, and S. Parthasarathy, *An Ensemble Framework for Clustering Protein-Protein Interaction Networks*. London, U.K.: Oxford Univ. Press, 2007.
[29] P. De Meo, E. Ferrara, G. Fiumara, G. Provetti, "Enhancing community detection using a network weighting strategy," *Inf. Sci.*, vol. 222, pp. 648–668, Feb. 2013.
[30] J. W. Berry, B. Hendrickson, R. A. Laviolette, and C. A. Phillips, "Tolerating the community detection resolution limit with edge weighting," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 5, p. 056119, 2011.
[31] D. Lai, H. Lu, and C. Nardini, "Enhanced modularity-based community detection by random walk network preprocessing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, p. 066118, Jun. 2010.
[32] E. Ravasz, A. L. Somera, and D. A. Mongru, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.
[33] Z.-X. Wang, Z.-C. Li, X.-F. Ding, and J.-H. Tang, "Overlapping community detection based on node location analysis," *Knowl.-Based Syst.*, vol. 105, pp. 225–235, Aug. 2016.
[34] G. J. Qi, C. C. Aggarwal, and T. Huang, "Community detection with edge content in social media networks," in *Proc. IEEE Int. Conf. Data Eng.*, Apr. 2012, pp. 534–545.

[35] T. Zhou, L. Lü, and Y. C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.
[36] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.
[37] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, pp. 2011–2024, Oct. 2009.
[38] Y. Cui, X. Wang, and J. Eustace, "Detecting community structure via the maximal sub-graphs and belonging degrees in complex networks," *Phys. A, Stat. Mech., Appl.*, vol. 416, pp. 198–207, Dec. 2014.
[39] M. Hajiabadi, H. Zare, and H. Bobarshad, "IEDC: An integrated approach for overlapping and non-overlapping community detection," *Knowl.-Based Syst.*, vol. 123, pp. 188–199, May 2017.
[40] Y. Cui, X. Wang, and J. Li, "Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient," *Phys. A, Stat. Mech., Appl.*, vol. 405, pp. 85–91, Jul. 2014.
[41] M. Newman. (2015). *Network Dataset*. [Online]. Available: http://www-personal.umich.edu/?mejn/netdata/
[42] Q. Cai, L. Ma, and M. Gong, "A survey on network community detection based on evolutionary computation," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 2, pp. 84–98, 2014.
[43] P. Zhang, "Evaluating accuracy of community detection using the relative normalized mutual information," *Comput. Sci.*, vol. 2015, no. 11, p. P11006, 2015.
[44] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *J. Stat. Mech., Theory Exp.*, vol. 2009, p. 03024, Mar. 2009.
[45] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 066133, 2004.
[46] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, p. 016118, 2009.
[47] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, p. 43, 2013.
[48] KONECT. (2015). *Network Dataset*. [Online]. Available: http://konect.uni-koblenz.de/networks
[49] J. Crnic, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.
[50] P. Jaccard, "Etude de la distribution florale dans une portion des Alpes et du Jura," *Bull. de la Soc. Vaudoise des Sci. Naturelles*, vol. 37, pp. 547–579, Jan. 1901.
[51] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 73, no. 2, p. 026120, 2006.



QI LI received the M.S. degree from the College of Information Engineering, Shanghai Maritime University, in 2015. He is currently pursuing the Ph.D. degree with the College of Computer Science, Chongqing University, Chongqing. His research interests include data mining and graph computing.

JIANG ZHONG received the Ph.D. degree from Chongqing University, Chongqing, China, in 2005. He is currently a Professor with Chongqing University. His main research interests include data mining, management information system, trusted computer system, and service computing.

QING LI is currently pursuing the Ph.D. with the College of Computer Science, Chongqing University. She is the author/co–author of papers in conferences, book chapters, and journals. Her main research interests include data mining, big data analytics, and machine learning.

CHEN WANG received the bachelor's and master's degrees from the Chongqing University of Posts and Telecommunications, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in software engineering at Chongqing University. His main research interests include machine learning, computer vision, and data mining.

ZEHONG CAO (M'18) received the B.E. degree in electronic and information engineering from Northeastern University, Shenyang, China, in 2012, the M.S. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 2013, the Ph.D. degree in information technology from the University of Technology Sydney (UTS), Ultimo, NSW, Australia, and the Ph.D. degree in electrical and control engineering from National Chiao Tung University (NCTU), Hsinchu, China, in 2017. He is currently a Research Fellow with the Centre for Artificial Intelligence, UTS. His current research interests include data science, human–machine interfaces, computational intelligence, pattern recognition, machine learning, and clinical applications. He was a recipient of the NCTU & Songshanhu Scholarship in 2013, the UTS President Scholarship in 2015, the UTS CAI Best Paper Award in 2017, the UTS FEIT Publication Award in 2017, and the CAMP Award in 2018. He serves as an Associate Editor for the IEEE Access and an Editorial Board Members of SCI-journals, including *Advances in Robotics and Automation* and the *International Journal of Sensor Networks and Data Communications*. He was invited to give oral presentations at IEEE-FUZZY in 2017, IJCNN in 2015, and BME Annual Conference of Taiwan in 2015.