International Journal of Data Science and Analytics (2022) 14:27–44
https://doi.org/10.1007/s41060-022-00314-3

REGULAR PAPER

An interaction-based method for detecting overlapping community structure in real-world networks

Pawan Kumar[1] · Ravins Dohare[2]

Received: 9 April 2021 / Accepted: 27 January 2022 / Published online: 11 March 2022

Abstract
A central theme of network analysis, these days, is the detection of community structure as it offers a coarse-grained view of the network at hand. A more interesting and challenging task in network analysis involves the detection of overlapping community structure due to its wide-spread applications in synthesising and interpreting the data arising from social, biological and other diverse fields. Certain real-world networks possess a large number of nodes whose memberships are spread through multiple groups. This phenomenon called community structure with pervasive overlaps has been addressed partially by the development of a few well-known algorithms. In this paper, we presented an algorithm called Interaction Coefficient-based Local Community Detection (IC-LCD) that not only uncovers the community structures with pervasive overlaps but do so efficiently. The algorithm extracted communities through a local expansion strategy which underlie the notion of interaction coefficient. We evaluated the performance of IC-LCD on different parameters such as speed, accuracy and stability on a number of synthetic and real-world networks, and compared the results with well-known baseline algorithms, namely DEMON, OSLOM, SLPA and COPRA. The results give a clear indication that IC-LCD gives competitive performance with the chosen baseline algorithms in uncovering the community structures with pervasive overlaps. The time complexity of IC-LCD is $\mathcal{O}(nc_{\max})$, where n is the number of nodes, and $c_{\max}$ is the maximum size of a community detected in a network.

Keywords Node interaction coefficient · Edge interaction coefficient · Pervasive overlap · Community detection algorithm

1 Introduction

Networks have become an indispensable tool to study complex systems emerging from diverse areas including biological, social and computational sciences. Network techniques are widely used in data structure, data mining and artificial intelligence too. The emergence of giant social networking sites such as facebook and twitter has made the human–human interaction possible through complex social networks at an unprecedented level. The internet itself is a huge complex network where nodes represent the webpages and edges the hyperlinks given by one to others [9]. Molecular interactions in living organisms control mechanism of cellular functions [41]. Understanding the nature and dynamics of such large networks, specifically at mesoscale level, depends on their successful decomposition into smaller modules consisting largely of related nodes called communities [6]. Most of the previous studies have mainly focused on the disjoint community structures where each node belongs to no more than one community. However, in many real networks nodes possess the multiple groups simultaneously. Many networks such as co-authorship networks and the internet are likely to have large number of nodes whose interactions are spread across several groups. For example, influential scientists contribute to different disciplines of interdisciplinary nature by collaborating with other influential scientists. Likewise most often searched webpages on internet are well connected with several other websites. An individual in a social network may have interactions with multiple groups such as family, friends, peers, teachers, etc. Such nodes belong to multiple groups. This phenomenon is called *community structure with pervasive overlaps* [28,31,56], and it cannot be detected by the other traditional methods of overlapping community

✉ Ravins Dohare
ravinsdohare@gmail.com

Pawan Kumar
pkumariitd@gmail.com

[1] School of Sciences, Indira Gandhi National Open University, New Delhi, India

[2] Centre for Interdisciplinary Research in Basic Sciences, Jamia Millia Islamia, New Delhi 110025, India

detection. Although, there are few algorithms based on pervasive overlap phenomenon such as CPM [40], LinkCom [2], SPLA [54], OSLOM [26], COPRA [17] but these are not satisfactory upto the mark and suffer with slow speed. A detailed discussion on these and few more algorithms is given in Sect. 2.

To overcome these limitations, we have proposed a novel algorithm called IC-LCD—Interaction Coefficient based Local Community Detection. As the name suggests, IC-LCD algorithm is based on node and edge interaction coefficient, where edge interaction coefficient is a newly defined parameter. Community detection through IC-LCD takes place in two phases. Phase-1 (*Expansion Phase*) expands communities starting from seeds containing single nodes, and Phase-2 (*Refinement Phase*) involves the merging of unstable or smaller communities into stable and larger communities. We have tested the proposed algorithm on LFR benchmarks and a range of real-world networks including some large networks. To compare its performance we choose the algorithms, namely DEMON, SPLA, OSLOM and COPRA as the baseline algorithms due to their wide-spread use in the literature. The results indicate that IC-LCD gives competitive performance with all these algorithms in terms of speed, accuracy and stability.

2 Related work

The overlapping community detection algorithms can be classified on the aspect whether they employ a *local* strategy or a *global* one for clustering nodes. Usually a global algorithm uses overall topological information about the network to optimise a global quality function such as modularity [34], or overlapping modularity [22,39]. Generally, this process of community detection uses an agglomerative [35] or divisive [52] approach. Some examples of global methods are FCM by Zhang et al. [59], CONGO (Cluster-Overlap Newman Girvan Algorithm) by Gregory [16], and MOSES by McDaid and Hurley [31]. On the other hand, the algorithms such as CPM [40] and [43] employ local optimisation strategies for producing overlapping communities. The algorithm FCM is based on a combination of three concepts namely, modularity [38], spectral relaxation [52] and fuzzy c-means clustering method [7]. But, the computational complexity is a key problem of FCM (See Table 1 for more details.). CONGA provides good results but it is slow and has approximately cubic time complexity, so it cannot cope with large size networks. CONGA inherits its low speed from the GN algorithm [13]. Both CONGA and GN algorithms rely on betweenness, which is a global centrality measure: at each step, it counts the number of shortest paths between all pairs of vertices in the network. For a fast and scalable algorithm, there was an urgent need that a measure can be computed locally. Hence, Gregory [16] introduced the concept of local betweenness, and then described the new algorithm: CONGO (CONGA Optimized) algorithm. Although, CONGO is good for sparse networks, and is slightly faster than CONGA, but its accuracy is decreased (See Table 1 for complexity.). McDaid and Hurley [31] proposed MOSES algorithm which is based on the same global objective function (modularity) but with the incorporation of greedy maximisation strategy in which communities are created and deleted, nodes are added and deleted from community in order to maximise the objective function. It is learnt that MOSES is not able to attain high NMI scores on LFR benchmarks, despite being a fast algorithm.

On the other hand, local strategy-based algorithms such as EAGLE [45] and OSLOM [26] utilise local expansion and optimization-based strategies for growing a seed community (Other details can be seen in Table 1). In these methods, a seed community is grown only through a local structure such as neighbourhood around the seed. In some cases, the seed community can be started with a single node such as in LFM [25] and MONC [19]. However, a k-clique (complete subgraph on k vertices) is seed community for GCE [28] algorithm. Most of the local algorithms rely on a local benefit function that characterises the quality of a densely connected group of nodes. However, problems arise with both the approaches. For instance, global methods usually suffer with resolution limits [11], whereas local methods are unable to scale with network size.

Node versus link is another classification of the community detection algorithms. Structural information of networks lies in node degrees, clustering coefficients, etc. Usually the node clustering methods exploit the node degree or node centrality to extract the community structure [25]. A typical limitation of node-based algorithms is that they tend to produce smaller overlapping regions among communities [53]. The notion of link clustering was proposed by Palla et al. [40] with the presumption that overlapping community structure can be effectively recovered if the clustering is done through links rather than the nodes.

They offered the Clique Percolation Method (CPM) which is based on the definition that a community, or more specifically a k-clique community, is the union of all k-cliques that can be reached to each other through a series of adjacent k-cliques. The output of CPM is, however, sensitive to the parameter k. Another interesting algorithm for link clustering is LinkCom [2] proposed by Ahn et al. which works on the concept of link similarity. It has been argued that, in general, the link clustering methods perform no better than node clustering methods [12]. To resolve some of the drawbacks of node and link clusterings, recently an algorithm was proposed that can extract node, link and hybrid node-link communities through a probabilistic model [20]. Several other algorithms have been developed based on probabilistic modelling for community detection and these are

Table 1 Details of some state-of-the-art overlapping community detection algorithms

Algorithm with references	Algorithm type	Algorithm complexity	Remarks
FCM [59]	Global quality function based	$\mathcal{O}(nK^2)$, where n is no. of nodes, K is no. of eigenvectors	High computational complexity
CONGA [15]	Global quality function based	$\mathcal{O}(m^3)$ in worst case, where m is no. of edges	High computational complexity
CONGO [16]	Global quality function based	$\mathcal{O}(m \log m + \frac{m^{2h+2}}{n^{2h+1}})$ or $\mathcal{O}(n \log n)$ for a sparse network	Good for sparse networks, less accurate than CONGA
MOSES [31]	Global quality function based with greedy strategy	Not calculated	Fast speed but low NMI scores on LFR benchmarks, good for highly overlapped communities
EAGLE [45]	Modularity based local expansion	$\mathcal{O}(n^2 s)$, where n is no. of nodes, s is no. of maximal cliques	Good for overlapping and hierarchical community structure, slow speed, not scalable
OSLOM [26]	Local expansion based on the homogeneous random graph	not measured	Good for hierarchical community in directed, weighted, and dynamic networks
COPRA [17]	Label propagation based	$\mathcal{O}(v^3 n)$, where n is no. of nodes, v is the maximum no. of node memberships	Good for overlapping and weighted communities, but not hierarchical
SLPA [54]	Speaker-Listener based label propagation	$\mathcal{O}(tn)$, where n is number nodes, t is number of iterations	
SIMGT [49]	Game theory based framework	$\mathcal{O}(mkt)$, where m is the no. of edges in the rewired network, k is the number of communities and t is the number of iterations	Tested on networks having upto 5000 nodes
DEMON [4]	Label propagation on ego-minus-ego network	$\mathcal{O}(nk^{3-\alpha})$, where n is the number of nodes, k is the maximum degree, α the scale-free exponent	Difficulty in parameter selection and highly unstable

highly applicable for the set of social media, target events, story line detection, protein function predictions, etc. DIM3 (*dynamic infinite mixed-membership stochastic blockmodel*) [8] is one such algorithm based on stochastic mechanism, and it is highly applicable on dynamical sets of data where data is changing day by day. NOODLES (*overlappiNg cOmmunities and rOles from noDe Links and messagES*) [5] is another algorithm for community detection based on the probabilistic approach and it can be applied to detect certain attributes from real data. This algorithm has extensively been experimented on real-world social media data for community detection and link prediction. However, alike many other algorithms it also needs further scope of improvement, for instance in eliminating the unrealistic communities.

Many overlapping community detection algorithms use the idea of label propagation introduced by Raghavan et al. [42]. It has been extended to incorporate multiple memberships of vertices in the methods such as COPRA [17], SLPA [54] and DEMON [4]. In COPRA each node updates its label and the belonging coefficients average out from the coefficients of all its neighbours in a synchronous manner. SLPA is a general speaker–listener–based information propagation process. It spreads label information between nodes according to the pairwise interaction rules. In SLPA, each node has a memory space to store the received information. The probability of observing a label in the memory of a node is perceived as the membership strength. So it needs a post-processing parameter to generate communities. However, there are no effective strategies for the proper parameter setting. The algorithm DEMON first extracts the ego-minus-ego network from the given network and then it applies the label propagation to discover communities. Recently, Sun et al. [47] has proposed a link propagation-based algorithm called LinkLPA for overlapping community detection. But LinkLPA also performs poorly on synthetic networks as well as on ground truth networks such as DBLP in terms of overlapping modularity [39]. Apart from all above discussion on overlapping community detection algorithms, few more attempts have been made for community detection in general or specific network [46,48,51,58,60].

To be specific we are concerned about the pervasive overlaps where "nodes possess large number of memberships" rather than communities with large overlapping regions. The task of node-community assignment in such networks is often hard. The algorithms such as DEMON, CPM and LinkCom can detect pervasive overlaps somehow satisfactorily but they suffer with slow speed [10,55,57]. Also they need reliable quality measures for proper evaluation. On LFR networks SLPA, OSLOM and COPRA show better performance but in most real networks they have detected less than 20% overlapping nodes [53]. In this direction, the performance of GCE and MOSES are noteworthy, especially on LFR benchmarks. In real networks, however, both fail to detect more than 10 memberships per node.

Accordingly, in this article, we have chosen to study the problem of community detection with "pervasive overlaps". Our efforts have led us to develop an algorithm called IC-LCD (Interaction Coefficient based Local Community Detection). Interaction coefficient, the notion underlying IC-LCD can be defined for nodes as well as for edges with a subgraph. IC-LCD exploits the node and edge interaction coefficients combinedly under the framework of local seed expansion to reveal the community structure of real networks. We experiment with IC-LCD on a number of synthetic and real-world networks and compare the results with the algorithms DEMON, SLPA, OSLOM, and COPRA. The results indicate that IC-LCD gives competitive performance with the baseline algorithms in uncovering the community structure with pervasive overlaps.

3 Methodology

3.1 Definitions and terminologies

We denote a graph with G, its vertex (node) set with V_G, and edge set with E_G. We shall write $C \subseteq G$ to indicate that C is an induced subgraph of G. Then the sets V_C and E_C are the vertex set, and the edge set of C, respectively. The notation N_v indicates the set of neighbours of a vertex v in G, and d_v is the degree of v. Let $C \subseteq G$. Then N_C is the *neighbourhood* of C defined as the set of all neighbours of the nodes of the subgraph C excluding V_C, i.e.,

$$N_C = \{v \in V_G \backslash V_C : v \in N_u \text{ for some } u \in C\}.$$

We shall write N_{uC} to denote the *common neighbourhood* of C and u, i.e., $N_{uC} = N_u \cap N_C$. Traditionally, seed expansion-based algorithms use coefficients such as the one defined below.

Definition 1 (*Node interaction coefficient*) Given a subgraph $C \subseteq G$ and a node $v \in N_C$, the *node interaction coefficient* of v with C is the quantity:

$$\xi_{\text{node}}(v, C) = \frac{|N_v \cap V_C|}{d_v} \tag{1}$$

Some authors define a node interaction coefficient as a weighted average of the quantities $|N_v \cap V_C|$ and $|N_v \cap N_C|$, see [23] for instance. Another approach is to define the interaction coefficient of an *edge* with a subgraph. So, consider a subgraph C of a graph G. Let e_{uv} be an edge of G such that its endpoints u and v are in N_C. Note that

$$|N_u \cap V_C| \leq d_u - 1, \quad |N_v \cap V_C| \leq d_v - 1,$$

which implies,

$$\min\{|N_u \cap V_C|, |N_v \cap V_C|\} \leq \min\{d_u - 1, d_v - 1\}$$

Hence, $1 + \min\{|N_u \cap V_C|, |N_v \cap V_C|\} \leq \min\{d_u, d_v\}$, which is smaller than or equal to $\max\{d_u, d_v\}$. This leads to the following definition.

Definition 2 (*Edge interaction coefficient*) Let G be a graph and $C \subseteq G$. Let e_{uv} be an edge of G with endpoints $u, v \in N_C$. Then the *edge interaction coefficient* of e_{uv} with C is defined as

$$\xi_{\text{edge}}(e_{uv}, C) = \frac{1 + \min\{|N_u \cap V_C|, |N_v \cap V_C|\}}{\max\{d_u, d_v\}} \qquad (2)$$

3.2 Seed expansion criterion

As described in the introduction, Phase-1 of IC-LCD is involved in the expansion of seed communities, where each seed community starts from a single node. To expand a seed community, one essentially needs to know which nodes are to be included in it. This task is performed by the procedure GET-NEW-NODES() of our algorithm, given in Algorithm 1. The arguments of the procedure GET-NEW-NODES() are the graph G, a subgrapgh C of G, the neighbourhood N_C of C, and the parameters ξ_0 and $n_{\min}$. The subgraph C is known as a seed community, and its local expansion solely depends on how it interacts with its neighbourhood N_C. The interaction between C and N_C can be measured in two ways—using node interaction coefficient (ξ_{node}) and edge interaction coefficient (ξ_{edge}). The threshold value for both these coefficients is taken as ξ_0. How this is done is explained in "Appendix A.1." The parameter $n_{\min}$ indicates the minimum size of the detected communities.

Let G be a graph where the nodes in V_G are, for the sake of simplicity, assumed to be positive integers. Let $C \subseteq G$. In order to expand C into a community, we must have some criterion to pick new nodes from the neighbourhood of C. For this task, we select a set V_{new} which is empty, initially. Then, for each $u \in N_C$, if $\xi_{\text{node}}(u, C) \geq \xi_0$, u is added to V_{new}, otherwise if $|N_C| \geq |V_C|$, then for each $v \in N_{uC}$ with $d_v > d_u$, if $\xi_{\text{edge}}((u, v), C) \geq \xi_0$, both u and v are added to V_{new}. Note that $v > u$ is taken to avoid computing ξ_{edge} twice for the same edge.

If the steps above yield $V_{\text{new}} = \varnothing$ and the size of C is still smaller than $n_{\min}$, then all those nodes $u \in N_C$ that satisfy $\xi_{\text{node}}(u, C \cup N_C) \geq \xi_0$ are added to V_{new}. This last step which we call *augmentation step* plays a pivotal role during the expansion phase of a community. The method for computing the new nodes is listed in Procedure GET-NEW-NODES (See Appendix A for illustration.).

Procedure GET-NEW-NODES($G, C, N_C, \xi_0, n_{\min}$)

```
Result: $V_{\text{new}}$
1  $V_{\text{new}} \leftarrow \varnothing$
2  for $u \in N_C$ do
3      if $\xi_{node}(u, C) \geq \xi_0$ then
4          $V_{\text{new}} \leftarrow V_{\text{new}} \cup \{u\}$
5      else
6          if $|N_C| \geq |C|$ then
7              for $v \in N_{uC}$ do
8                  if $v > u$ then
9                      if $\xi_{edge}((u, v), C) \geq \xi_0$ then
10                         $V_{\text{new}} \leftarrow V_{\text{new}} \cup \{u, v\}$
11                     end
12                 end
13             end
14         end
15     end
16 end
   /* Augmentation step                                   */
17 if $V_{new} = \varnothing$ then
18     if $|C| < n_{min}$ then
19         for $u \in N_C$ do
20             if $\xi_{node}(u, C \cup N_C) > \xi_0$ then
21                 $V_{\text{new}} \leftarrow V_{\text{new}} \cup \{u\}$
22             end
23         end
24     end
25 end
```

3.3 Overlap and merging criterion

The strategies of determining the overlapping function and the consequent merging criterion is the next important aspect of a seed expansion based algorithm. A simple distance measure between two communities C_1 and C_2 is given below [28]:

$$\delta(C_1, C_2) = 1 - \frac{|V_{C_1} \cap V_{C_2}|}{\min\{|V_{C_1}|, |V_{C_2}|\}}.$$

Note that $1 - \delta(C_1, C_2)$ can be taken as the overlap between C_1 and C_2. However, this overlap simply depends on the number of nodes in the common region of C_1 and C_2. It does not take into account the edges between C_1 and C_2, and the edges within C_1 and C_2. It is quite possible that $V_{C_1} \cap V_{C_2} = \varnothing$, but there are a large number of edges with one endpoint in C_1 and the other endpoint in C_2. In such cases $1 - \delta(C_1, C_2)$ would be zero, still they could be merged to form a stable community.

To arrive at a more sensitive overlap function let us concentrate on the quantity $|\text{cut}(C_1, C_2)|$, which represents the number of edges with one endpoint in C_1 and the other endpoint in C_2. To normalise it we divide it by the minimum of the external degrees of C_1 and C_2, i.e., by the minimum of $|N_{C_1}|$ and $|N_{C_2}|$. Thus we have

$$0 \leq \frac{|\text{cut}(C_1, C_2)|}{\min\{|N_{C_1}|, |N_{C_2}|\}} \leq 1 \tag{3}$$

This indicates that C_1 and C_2 could be merged if either of them shares a large fraction of its external edges with another. However, this argument may fail in certain cases. For example, consider two cliques C_1 and C_2 joined by a single edge, and let they have no other external edges. Then merging C_1 and C_2 would not be a good idea. So, we need to look at some other factor that makes the merging of two communities necessary. In fact, it is the internal density (the ratio of the number of internal edges present to the maximum possible number of internal edges) of a community that makes it stable or unstable. So, consider the quantity

$$\alpha(C_1, C_2) = 1 - \frac{\min\{\|C_1\|, \|C_2\|\}}{\max\{\|C_1\|_{\max}, \|C_2\|_{\max}\}}, \tag{4}$$

which takes into account the combined effect of the internal densities of both C_1 and C_2. Here $\|C\|$ denotes the number of edges in C, and $\|C\|_{\max}$ the maximum possible number of edges in C. That is, $\|C\|_{\max} = \binom{|C|}{2}$. Note that $\alpha(C_1, C_2)$ lies between 0 and 1, where 1 is achieved when either $\|C_1\|$ or $\|C_2\|$ is zero. When $\|C_1\|$ and $\|C_2\|$ increase, which means that both C_1 and C_2 have large number of internal edges, $\alpha(C_1, C_2)$ decreases. So, α can serve as a factor responsible for amalgamation of C_1 and C_2.

Now we are ready to define our overlap function. Given a cover $\mathcal{K}$ containing the initial communities, we define the overlap function between any $C_1, C_2 \in \mathcal{K}$ as

$$\theta(C_1, C_2) = \frac{\alpha(C_1, C_2)\,|\text{cut}(C_1, C_2)|}{\min\{|N_{C_1}|, |N_{C_2}|\}} \tag{5}$$

We now present the criterion for merging two communities that have overlap greater than a certain threshold, say θ_0. The method for merging communities is described in Procedure MERGE-COMS(), which takes as input a graph G, the initial communities of G stored in a container $\mathcal{K}$, and the corresponding membership container m, and the parameters θ_0 and $n_{\min}$. (See Table 2 for details.) The parameter $n_{\min}$ is included just to facilitate the user to get the communities smaller than a specific size merged into another suitable and stable community.

The method works as follows. All the labels of the initial communities in $\mathcal{K}$ are stored in a variable L. Then in a repeat-until loop (lines 2–21), a label ℓ is chosen from L randomly. Let C be the community corresponding to ℓ. The labels of the communities that are larger than or equal to C in size, and that share at least one edge with C are stored in L_N (See line 5.).

Now a variable L_H is required to store the labels of the communities into which C may be merged. There are two cases depending on whether $|C| < n_{\min}$ or $|C| \geq n_{\min}$. In the first case, L_H stores the labels of those $C' \in \mathcal{K}$ for which $\theta(C, C')$ is the maximum (see line 7). In the second case, on the other hand, L_H stores the labels of all those $C' \in \mathcal{K}$ that satisfy $\theta(C, C') \geq \theta_0$. (See line 9.)

The lines 11–16 simply perform the task of merging C into each of the communities with labels L_H and updating the memberships of all the vertices of C accordingly. Then, at line 17, ℓ is removed from L. The lines 18–20 remove C from $\mathcal{K}$ whenever L_H is nonempty. This means, when $L_H = \varnothing$, C will not be removed from $\mathcal{K}$ even if the size of C is smaller than $n_{\min}$.

Procedure MERGE-COMS($G, \mathcal{K}, m, \theta_0, n_{\min}$)

```
Result: $\mathcal{K}, m$
1  $L \leftarrow \{1, 2, \ldots, |\mathcal{K}|\}$
2  repeat
3      Pick $\ell \in L$ at random.
4      $C \leftarrow C_\ell$
5      $L_N \leftarrow \{i \in L : i \neq \ell, |C_i| \geq |C|, \text{cut}(C, C_i) \neq \varnothing\}$
6      if $|C| < n_{min}$ then
7          $L_H \leftarrow \{i \in L_N : \theta(C_i, C) = \max_{j \in L_N} \theta(C_j, C)\}$
8      else
9          $L_H \leftarrow \{i \in L_N : \theta(C_i, C) \geq \theta_0\}$
10     end
11     for $i \in L_H$ do
12         for $v \in C$ do
13             $C_i \leftarrow C_i \cup \{v\}$
14             $m_v \leftarrow m_v \cup \{i\} \setminus \{\ell\}$
15         end
16     end
17     $L \leftarrow L \setminus \{\ell\}$
18     if $L_H \neq \varnothing$ then
19         $\mathcal{K} \leftarrow \mathcal{K} \setminus C$
20     end
21 until $L = \varnothing$
```

3.4 Main algorithm and parameter selection

Looking at the different parts of IC-LCD we are now ready to describe the full algorithm which is given in Algorithm 1. The algorithm maintains primarily three containers—$\mathcal{K}$, U, and m as described in Table 2.

Initially, $\mathcal{K}$ is empty, and U contains all the vertices of G, that is, $U = V_G$. In the beginning $m_u = \varnothing$, for each u, so m is also empty. The algorithm now works as follows. The counter i is initialised with 1 (line 5). Since in the beginning U is nonempty, a seed, say u, is selected from U randomly (line 7). Let C be the set initialised with u (line 8). Lines 9 through 18 update C as follows. V_{new} is computed for C through the procedure GET-NEW-NODES(). Next, for each $v \in V_{\text{new}}$, v is added to C, v is removed from N_C, and any neighbour of v not in C, is added to N_C. When C can no more be expanded, C is added to $\mathcal{K}$ at the i^{th} position (line 19).

Table 2 Containers and parameters of IC-LCD

Containers and parameters	Description
G	The input graph
$\mathcal{K}$	The container holding the communities. We write $\mathcal{K} = \{C_1, C_2, \ldots, C_k\}$, where C_i is the community stored at the ith position in $\mathcal{K}$
U	The set of unexplored vertices
m	The container holding the memberships of all the vertices of G. Indeed, $m = \{m_u : u \in V_G\}$, where m_u stores the memberships of the vertex u
$n_{\min}$	The minimum size of a community the user is interested in
θ_0	The threshold value for the overlap function θ
ξ_0	The threshold value for both ξ_{node} and ξ_{edge}

Algorithm 1: IC-LCD

```
Input: $G, \theta_0, n_{\min}$
Result: $\mathcal{K}, m$
1  $\mathcal{K} \leftarrow \varnothing$
2  $U \leftarrow V_G$
3  $m_u \leftarrow \varnothing, \forall u \in V_G$
4  $m \leftarrow \{m_u : u \in V_G\}$
5  $i \leftarrow 1$
6  while $U \neq \varnothing$ do
7      Pick $u \in U$ at random.
8      $C \leftarrow \{u\}$
9      repeat
10         $V_{\text{new}} \leftarrow$ GET-NEW-NODES($G, C, N_C, 0.5, n_{\min}$)
11         for $v \in V_{new}$ do
12             $C \leftarrow C \cup \{v\}$
13             $N_C \leftarrow N_C \backslash \{v\}$
14             for $w \in N_v \backslash C$ do
15                 $N_C \leftarrow N_C \cup \{w\}$
16             end
17         end
18     until $V_{new} = \varnothing$
19     $C_i \leftarrow C$
20     for $v \in C$ do
21         $m_v \leftarrow m_v \cup \{i\}$
22         $U \leftarrow U \backslash \{v\}$
23     end
24     $i \leftarrow i + 1$
25 end
26 MERGE-COMS($G, \mathcal{K}, m, \theta_0, n_{\min}$)
```

Then, the lines 20–23, update the memberships of each of the vertices in C, and C is removed from U. Line 24 increments i with 1. As long as U remains nonempty, lines 7 through 24 get repeated. At the end of line 25, $\mathcal{K}$ contains the list of all the initial communities generated in the previous steps. Finally, the procedure MERGE-COMS() is called to refine the communities. Finally, the algorithm prints $\mathcal{K}$ containing the communities of G, and the set m of the memberships of the vertices of G.

We have run our algorithm several times on different networks for a range of parameter values. We found 0.5 to be the best choice for ξ_0. So, it is internally fixed at $\xi_0 = 0.5$ meaning that it is not available for user manipulation (See line 10 of Algorithm 1.). For $n_{\min}$ we tried the values 2, 3, 4 and 5, and found that 5 is the most suitable choice. The next parameter is θ_0, which determines the maximum overlap allowed between any two communities, and it gives best results for 0.4. Both θ_0 and $n_{\min}$ can be set by the user as per his/her choice. The default values for them are as $\theta_0 = 0.4$ and $n_{\min} = 5$. At the same settings we have compared IC-LCD with other algorithms for all the networks considered in this article (See Table 3.).

4 Evaluation on synthetic networks

The most often used synthetic networks for testing community detection algorithms are Lancichinetti–Fortunato–Radicchi (LFR) networks [24]. This is because the LFR networks possess a number of parameters which allows users to control many network characteristics. The most prominent parameter is the mixing parameter (μ) which represents the fraction of the neighbours of a node lying outside the community of the node. Other important parameters are: N—the number of nodes, k—the average degree, $k_{\max}$—the maximum degree, $c_{\min}$—the minimum size for a community, $c_{\max}$—the maximum size of a community, O_n—the number of overlapping nodes, O_m—the number of memberships for each overlapping node. Based on the LFR benchmarks, we first talk about the accuracy measures.

4.1 Accuracy measures

Our criterion for measuring the accuracy of an algorithm is three-fold. The first measure is the widely used normalized mutual information (NMI) given by Lancichinetti et al. [25]. The NMI score varying between 0 and 1 indicates how much similar or dissimilar two community covers are. Second, we study how well community detection algorithms reproduce the number of communities. To measure this we consider the ratio $\overline{C}/C$, where C is the average number of communities present in LFR networks over 100 different realizations and $\overline{C}$ is the average number of communities detected by an algorithm in the same 100 networks. A value of $\overline{C}/C$ closer to 1 indicates the higher accuracy. As a third measure we study how well the algorithms reproduce the overlapping nodes. To this end we compute the ratio $\overline{OV}/OV$, where OV is

Table 3 Algorithms parameters

Algorithms	Parameters
IC-LCD	$n_{\min} = 5, \theta_0 = 0.4$
COPRA	$v = 10$
OSLOM	$hr = 0$
SLPA	$r = 0.05, t = 50$
DEMON	$\epsilon \in [0.1, 0.4]$

fixed at 100 which is the average number of overlapping nodes present in LFR networks over 100 realizations and $\overline{OV}$ is the average number of overlapping nodes detected by an algorithm in the same 100 networks.

We compute each of the three measures—NMI, $\overline{C}/C$ and $\overline{OV}/OV$, as independent functions of O_m and μ. For O_m we pick the range $\{2, 3, 4, \ldots, 15\}$ and fix other LFR parameters as $N = 2000, O_n = 100, \mu = 0.2, k = 50, k_{\max} = 100, c_{\min} = 10, c_{\max} = 150$. On the other hand for μ, we choose the range $\{0.05, 0.10, \ldots, 0.50\}$ and take $O_m = 8$ and other LFR parameters as above.

4.2 Accuracy results and comparison with competitors

To compare the results of IC-LCD, we have selected DEMON, SLPA, OSLOM, and COPRA as the baseline algorithms. The parameters for these algorithms are set as in Table 3. Note that the parameter ϵ of DEMON is not fixed at a particular value. Instead, we have taken some random discrete values of ϵ in the range $[0.1, 0.4]$, and recorded the output for the best choice. It is important to note that the algorithms SLPA and DEMON are implemented in the CDLib [44], which is a Community Discovery Library available for the python language. We have used this library for running the algorithms DEMON and SLPA, and also for computing the NMI measure. The results for all the three accuracy measures are shown in Fig. 1.

First we plot the NMI scores between the produced and actual covers of the five algorithms as a function of O_m (see Fig. 1a) and μ (see Fig. 1b). It can be observed that as O_m varies, IC-LCD delivers more stable NMI scores than the rest of the algorithms. When NMI is seen as a function of μ, among all the algorithms IC-LCD delivers the highest scores till μ reaches 0.3, and thereafter it starts degrading whereas OSLOM and COPRA maintain the high scores till $\mu = 0.5$. The worst performance on NMI is shown by DEMON, which indicates that it produces highly unstable covers.

Next we look at $\overline{C}/C$ as functions of O_m and μ (see Fig. 1c, d). The algorithm COPRA delivers the correct number of communities till $O_m = 13$, where as OSLOM does so only upto $O_m = 6$. IC-LCD on the other hand, underestimates the number of communities until O_m crosses 4 and thereafter it delivers the correct number of communities.

The algorithm SLPA underestimates the number of communities for the whole range of O_m. When μ varies till 0.35, the scores of $\overline{C}/C$ indicate that the number of communities are estimated correctly by both the COPRA and IC-LCD, whereas they are overestimated by OSLOM and underestimated by SLPA. In this case also DEMON exhibits the worst performance.

Finally, we consider how O_m and μ affect $\overline{OV}/OV$ (see Fig. 1e, f). As O_m increases, IC-LCD delivers scores of $\overline{OV}/OV$ that monotonically increase and stay above 0.8 when O_m crosses 7. The opposite behavior is shown by SLPA which begins with high scores of $\overline{OV}/OV$ and monotonically degrades as O_m increases. For OSLOM, the scores of $\overline{OV}/OV$ are closer to 1 until $O_m = 5$ and afterwards they start oscillating with wide variations. Both the algorithms COPRA and DEMON fail to detect the overlapping nodes as soon as O_m crosses 4. On the other hand, when $\overline{OV}/OV$ is measured as a function of μ, SLPA produces slightly better estimates for overlapping nodes than IC-LCD till μ reaches 0.2, and thereafter IC-LCD takes the lead. The other three algorithms deliver unrealistic estimates.

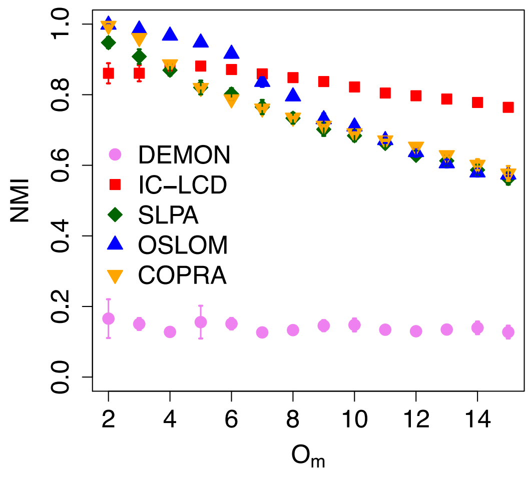
Overall, we find that IC-LCD delivers moderate performance when the overlapping nodes have fewer memberships (the number of communities per node). As their membership increases IC-LCD's performance starts improving while that of the others deteriorate. The transition point usually occurs at $O_m = 6$. This indicates that IC-LCD favors community structures with pervasive overlaps, *i.e.*, where the nodes possess memberships of several communities.

5 Evaluation on real-world networks

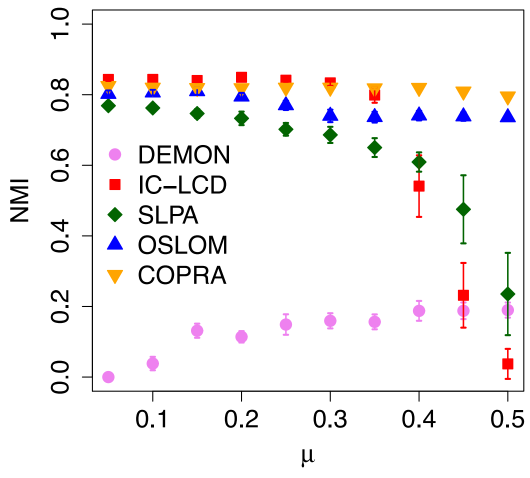
After understanding how IC-LCD performs on synthetic networks, we proceed to test it on real-world networks. The list of real-world networks we have chosen to experiment with IC-LCD is given in Table 4. The list includes a range of networks with the number of nodes varying from very small to about 335 thousand, and the number of edges varying upto about 926 thousand. The networks DBLP and AMAZON are quite large and are well suited for testing the scalability of algorithms. Since in real networks the prior information about the community structure is usually absent, we must employ other measures to understand how well an algorithm performs on them.

5.1 Quality measures

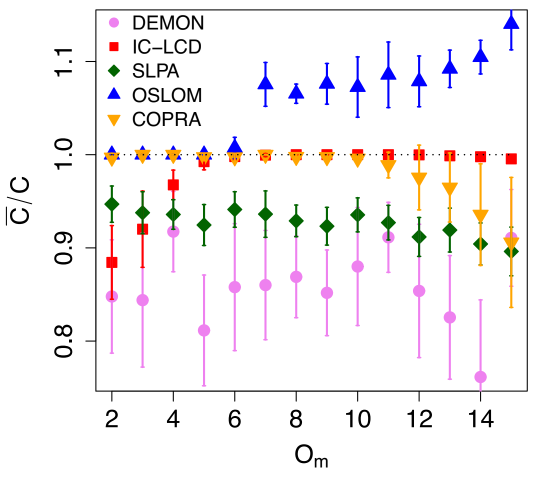
There are many quality measures for this purpose [27,30,39] which are motivated in some or other way by the *modularity* of Newman [37]. We select two such measures. The first



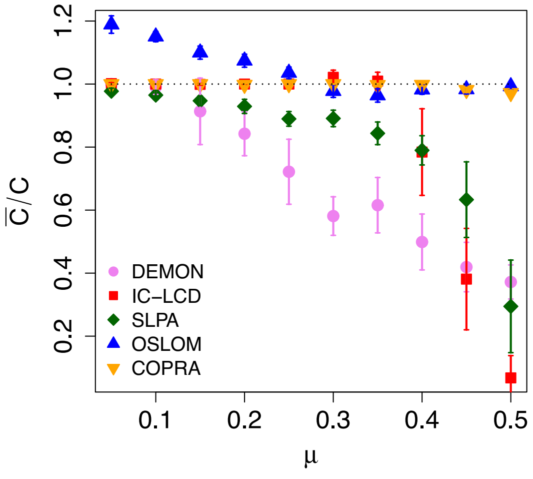
(a) Accuracy measure NMI versus parameter O_m



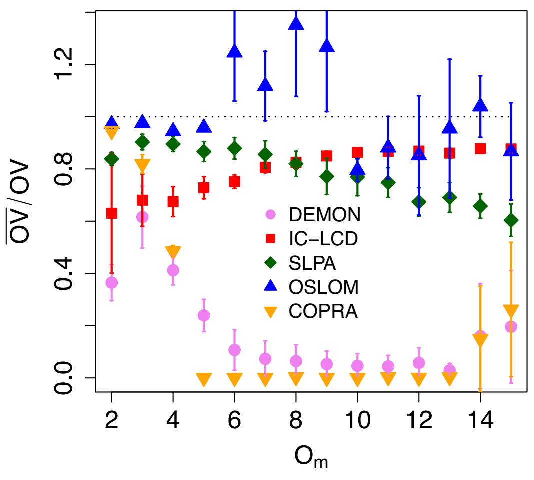
(b) Accuracy measure NMI versus parameter μ



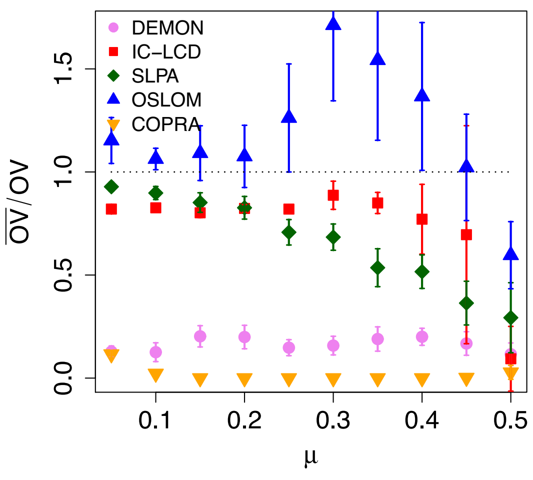
(c) Measure $\overline{C}/C$ versus parameter O_m



(d) Measure $\overline{C}/C$ versus parameter μ



(e) Measure $\overline{OV}/OV$ versus parameter O_m



(f) Measure $\overline{OV}/OV$ versus parameter μ

Fig. 1 Accuracy measures versus LFR parameters for the algorithms DEMON, IC-LCD, SLPA, OSLOM and COPRA

Table 4 Details of networks studied in this paper

Network	Nodes	Edges	References
LES-MISERABLES	77	254	[21]
JAZZ-MUSIC	198	2742	[14]
EMAIL-URV	1133	5451	[18]
POLBLOGS	1490	16715	[1]
NETSCIENCE	1589	2742	[36]
YEAST-PPI	2361	6646	[3]
POWER-GRID	4941	6594	[50]
HEP-THEORY	5835	13,815	[33]
ASTRO-PHYSICS	14,845	119,652	[33]
INTERNET	22,963	48436	[32]
DBLP	317,080	1,049,866	[29]
AMAZON	334,863	925,872	[29]

one is the overlapping modularity measure (Q_{ov}) given by Nicosia et al. [39] as it has been used quite extensively in the literature on overlapping community detection. The Q_{ov} score depends on a function $f(x) = 2px - p$, where p is any real number. The code for computing Q_{ov} implements the function f with $p = 30$. To compute the Q_{ov} scores we have used the CDLib (Community Discovery Library). The second measure is the weighted overlapping modularity measure Q_{wo} [22] which is equally applicable for unweighted networks. Unlike the traditional versions of modularity, Q_{wo} has a very simple formulation and it can be computed very fast.

We take the parameters of the algorithms again as given in the Table 3 except the parameter r of SLPA which we set now as $r = 0.45$. This value of r is chosen by the authors of SLPA for experiments on real networks. We run each of the five algorithms 10 times and record the average values $\overline{Q}_{ov}$, $\overline{Q}_{wo}$, $\overline{C}$ and $\overline{OV}_{frac}$ of the corresponding parameters over 10 runs.

5.2 Results and comparison with competitors

On all the 12 networks listed in Table 4, we applied IC-LCD along with the four baseline algorithms. The results for the selected measures for all chosen algorithms can be seen in Table 5. First, let us analyse how the five algorithms perform on the modularity measures. We find that the score $\overline{Q}_{ov} \geq 0.60$ is achieved on 7 networks by IC-LCD, on 5 networks by both OSLOM and COPRA, on 3 networks by SLPA, and unfortunately on zero networks by DEMON. Likewise, the score $\overline{Q}_{wo} \geq 0.60$ is achieved on 11 networks by IC-LCD, on 4 networks by OSLOM and DEMON both, on 6 networks by COPRA, and on 7 networks by SLPA.

This clearly indicates in terms of quality IC-LCD gives better performance than the baseline algorithms. On the other two parameters, i.e., $\overline{C}$ and $\overline{OV}_{frac}$, IC-LCD gives almost the similar results with the baseline algorithms.

Community Structure in LES-MISERABLES Network After giving a quantitative tests of the results on all the real-world networks listed in Table 4, we specifically select the network LES-MISERABLES for demonstration of its overlapping community structure found by IC-LCD and compare the results with those of the baseline algorithms. It is the weighted network of co-appearances of characters in Victor Hugo's novel "Les Miserables" [21]. It contains 77 nodes and 254 edges. Nodes represent characters as indicated by the labels and edges connect any pair of characters that appear in the same chapter of the book. The values on the edges are the number of such co-appearances. Since IC-LCD does not work for weighted networks, we have ignored the weights.

We ran IC-LCD on this network 10 times. The best cover produced by IC-LCD has 8 communities (See Fig. 2.). We can see that the character 'Valjean' has a central role in the novel. It has occurred with almost all the groups, although prominently with 4 groups. Other characters that co-appear with two or more groups are 'Marius', 'Cosette', 'Gavroche', 'Javert', 'Marguerite', and 'Simplice'. The modularity scores corresponding to this cover are $\overline{Q}_{ov} = 0.66$ and $\overline{Q}_{wo} = 0.52$.

When we ran OSLOM 10 times on this network we find that the best cover produced by it contains 6 communities with only two overlapping nodes namely, 'Valjean', and 'Marius'. Both of these nodes possess the memberships of just two communities. However, the cover gets a quality scores as $\overline{Q}_{ov} = 0.69$ and $\overline{Q}_{wo} = 0.55$. Next we run COPRA 10 times on this network. The best cover produced by COPRA contains only 3 communities (see Fig. 2c). Further, we can see that no node has membership more than 2. The modularity scores for this cover are $\overline{Q}_{ov} = 0.53$ and $\overline{Q}_{wo} = 0.59$. Finally, we look at (Fig. 2d) the cover produced by SLPA. It detects 6 communities, but with no overlapping node. The quality scores for this cover are $\overline{Q}_{ov} = 0.73$ and $\overline{Q}_{wo} = 0.58$. It may be noted that DEMON does not produce more than 1 communities in this network in 10 consecutive runs for different values of its parameter ϵ in the range $[0.1, 0.4]$.

On this network, our algorithm produced the maximum number of communities, with node membership reaching as high as 4. On the other hand, rest of the three algorithms produce upto a maximum of 6 communities, and node memberships no more than 2.

5.3 Node membership distribution

To understand how node memberships are spread across multiple communities we select three real-world networks namely, INTERNET, DBLP and AMAZON. The network INTERNET represents a symmetrised snapshot of the structure of the Internet at the level of autonomous systems, recon-

Table 5 Quality measure comparison of the algorithms IC-LCD, OSLOM, COPRA, SLPA and DEMON on real-world networks

	IC-LCD				OSLOM				COPRA			
	$\overline{Q}_{\text{ov}}$	$\overline{C}$	$\overline{OV}_{\text{frac}}$	$\overline{Q}_{\text{wo}}$	$\overline{Q}_{\text{ov}}$	$\overline{C}$	$\overline{OV}_{\text{frac}}$	$\overline{Q}_{\text{wo}}$	$\overline{Q}_{\text{ov}}$	$\overline{C}$	$\overline{OV}_{\text{frac}}$	$\overline{Q}_{\text{wo}}$
LES-MISERABLES	0.66	8	0.09	0.52	0.69	6	0.03	0.55	0.53	3	0.35	0.59
JAZZ-MUSIC	0.69	3	0.17	0.79	0.45	11	0.13	0.44	0.56	2	0.09	0.52
EMAIL-URV	0.54	154	0.27	0.88	0.49	36	0.06	0.50	0.20	3	0.03	0.45
POLBLOGS	0.80	8	0.03	0.63	0.42	15	0.05	0.32	0.76	5	0.10	0.68
NETSCIENCE	0.75	47	0.08	0.84	0.84	29	0.01	0.43	0.82	29	0.05	0.70
YEAST-PPI	0.51	287	0.18	0.74	0.54	60	0.03	0.35	0.84	55	0.00	0.78
POWER-GRID	0.72	692	0.15	0.68	0.89	156	0.01	0.88	0.64	1308	0.14	0.59
HEP-THEORY	0.63	731	0.25	0.60	0.65	598	0.04	0.34	0.78	489	0.16	0.58
ASTRO-PHYSICS	0.58	931	0.30	0.60	0.41	1218	0.15	0.42	0.58	256	0.06	0.63
INTERNET	0.55	937	0.46	0.79	0.53	176	0.04	0.63	0.26	2634	0.00	0.62
DBLP	0.58	31,629	0.15	0.69	0.54	17525	0.09	0.64	0.49	2565	0.04	0.58
AMAZON	0.68	24,418	0.17	0.72	0.73	17041	0.05	0.78	0.59	7105	0.20	0.60

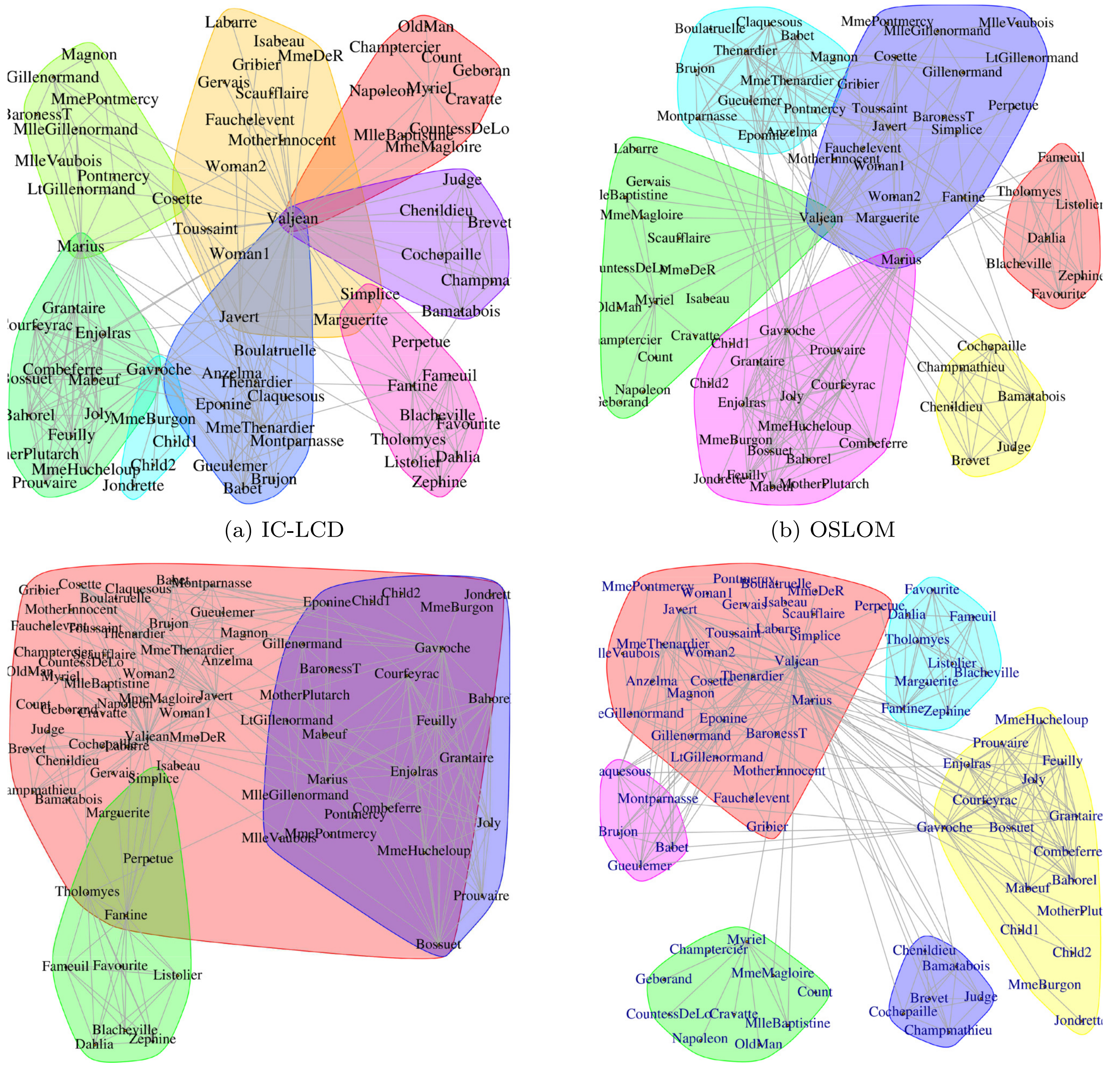
	SLPA				DEMON			
	$\overline{Q}_{\text{ov}}$	$\overline{C}$	$\overline{OV}_{\text{frac}}$	$\overline{Q}_{\text{wo}}$	$\overline{Q}_{\text{ov}}$	$\overline{C}$	$\overline{OV}_{\text{frac}}$	$\overline{Q}_{\text{wo}}$
LES-MISERABLES	0.65	6	0.00	0.54	0.13	1	0	0.10
JAZZ-MUSIC	0.45	11	0.35	0.52	0.14	1.1	0.1	0
EMAIL-URV	0.37	3	0.59	0.40	0.16	11.9	0.70	0.37
POLBLOGS	0.39	18	0.41	0.59	0.14	1	0	0
NETSCIENCE	0.72	24	0.04	0.96	0.11	106	0.14	0.92
YEAST-PPI	0.56	63	0.00	0.65	0.20	130	0.19	0.40
POWER-GRID	0.65	666	0.11	0.70	0.01	75	0.01	0.59
HEP-THEORY	0.52	460	0.25	0.83	0.10	298	0.33	0.78
ASTRO-PHYSICS	0.29	743	0.54	0.69	0.42	258	0.72	0.71
INTERNET	0.41	9	0.53	0.58	0.07	56	0.23	0.48
DBLP	0.50	19,059	0.59	0.68	0.30	9164	0.28	0.58
AMAZON	0.52	30,316	0.62	0.68	0.38	14701	0.40	0.74

structed from BGP tables posted at archive.routeviews.org. This snapshot was created by Mark Newman from data for July 22, 2006. From Fig. 3a which depicts the vertex membership distribution, we find that DEMON detects the highest number of memberships for the overlapping nodes. The rest of the algorithms including IC-LCD are not able to detect more than 5 memberships for any node.

The network DBLP represents a co-authorship network taken from the computer science bibliography database DBLP where two authors are connected if they have published at least one paper together. This is a large network with 317,080 nodes and 1,049,866 edges. The distribution of the node memberships of the community structures produced by IC-LCD and the baseline algorithms is shown in Fig. 3b. In this network also DEMON takes the lead and IC-LCD achieves the second place.

The AMAZON network was collected by crawling Amazon website. It is based on *Customers Who Bought This Item Also Bought* feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an undirected edge joining i and j. On this network also we find that DEMON detects the highest number of memberships for the overlapping nodes, whereas IC-LCD remains at the second position.

Apparently, DEMON is able to detect highest number of memberships for the overlapping nodes. However, its performance on LFR networks does not make it a reliable candidate. Moreover, its output significantly depends on the selection of its parameter ϵ. On the other hand, despite remaining at the second position, IC-LCD is more reliable in terms of the stability and quality of the covers produced by it. Thus in a sense, IC-LCD is capable of revealing the highly overlapping structure of networks.

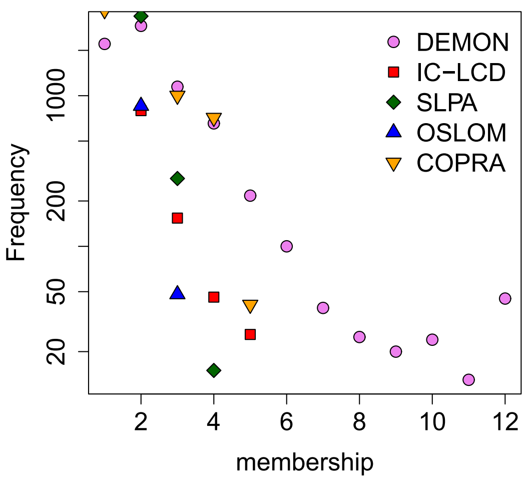

(a) IC-LCD (b) OSLOM

(c) COPRA (d) SLPA

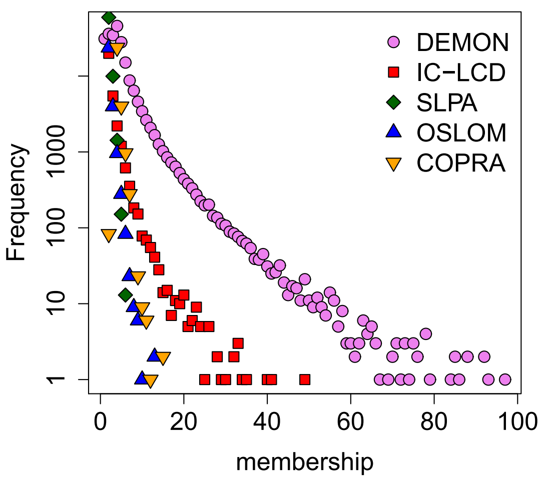
Fig. 2 Community structure in LES-MISERABLES networks

6 Scalability and time complexity

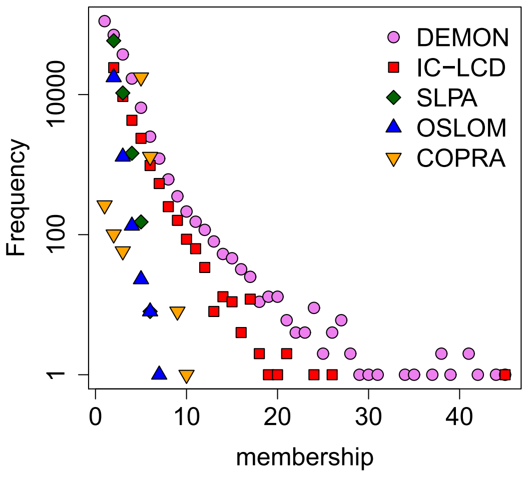
Finally, we study the most important aspect for any algorithm its scalability and time complexity. Scalability of an algorithm tells us how the run time of the algorithm varies with the number of nodes in the network. For comparing the actual run times of IC-LCD with the baseline algorithms we select random networks with the number of nodes ranging from 1000 to 100,000. These networks are generated using the *IGRAPH* Package available for the *R* language. We simply used the command `sample_gnp(n, p)` with $p = 5/n$, where n lies in $\{5000, 10{,}000, \ldots, 100{,}000\}$. The parameters set for the algorithms are again as in Table 3. The plot showing the run time of the algorithms DEMON, IC-LCD, SLPA, OSLOM and COPRA on random networks is shown in Fig. 4a. Here COPRA is the fastest algorithm, whereas IC-LCD is the third fastest algorithm. However, as the network size reaches 100,000 mark all the three algorithms COPRA, DEMON and IC-LCD seem to be on the same scale.



(a) INTERNET

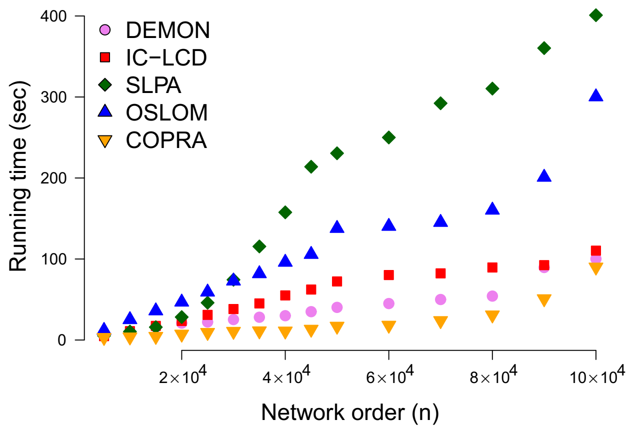


(b) DBPL

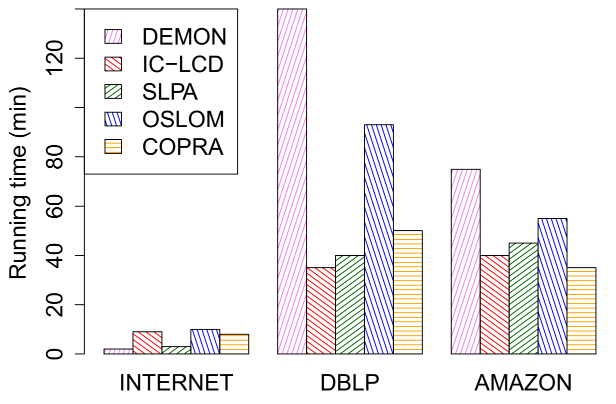


(c) AMAZON

Fig. 3 Relationships between node degree and shared communities in INTERNET,DBLP and AMAZON networks



(a) Synthetic networks



(b) Real Networks

Fig. 4 **a** Running time (in seconds) plot of IC-LCD along with the baseline algorithms on a class of random networks generated by the command `sample_gnp(n, p)` of igraph package, with n lying in $\{5000, 10{,}000, \ldots, 100{,}000\}$ and $p = 5/n$ **b** Running time (in minutes) plot of IC-LCD along with the baseline algorithms on the real networks INTERNET, DBLP and AMAZON

To observe the difference in running time of IC-LCD, and the baseline algorithms, we again considered the three real networks INTERNET, DBLP and AMAZON. While on INTERNET, the running times are not much distinguishable, the difference is clear on DBLP and AMAZON. On DBLP we find that IC-LCD takes the least time which is less than 40 min, whereas DEMON takes more than 120 min. On AMAZON, the least time is taken by COPRA, whereas IC-LCD takes the second least time again around 40 min. DEMON takes again the highest time on this network.

We shall now estimate the theoretical complexity of IC-LCD. First we compute the complexity of the Expansion Phase. For an arbitrary subgraph C of the input graph, assume that $c = |C|$ and $n_c = |N_C|$. First observe that both node and edge interaction coefficients of a node or an edge with C can be computed in $\mathcal{O}(c)$ time. The procedure GET-NEW-NODES takes a subgraph C and its neighbourhood N_C, and returns new nodes for the expansion of C. This can be done in $\mathcal{O}(cn_c)$ time. Now we come to the **while** loop (Lines 6–25) of Algorithm 1, which runs k times, where k is the number of (initial) communities. Note that the final communities are never more than k. Given a subgraph C, the **repeat-until** loop (Lines 9–18) runs at most $|V_{\text{new}}| \leq n_c$ times. So, the complexity of the **repeat-until** loop is $\mathcal{O}(cn_c^2)$. The assignments at Lines 19–24 take $\mathcal{O}(c)$ time. Thus the complexity of the **while** loop is $\mathcal{O}(kcn_c^2 + kc) = \mathcal{O}(kcn_c^2)$. Note that c and n_c are not constants, but they keep on changing as C expands. Normally, C and N_C do not grow simultaneously throughout the expansion phase. Indeed, for every C, we have $1 = c \leq n_c$ at the beginning. At the intermediate steps it is possible that $c \leq n_c$, but towards the end of the expansion phase two situations might emerge. In the first case, C does not grow after acquiring a few nodes, and so in this case it is reasonable to assume that $cn_c \leq n_{\max}$, where $c_{\max}$ is the maximum size of the final communities. In the second case, C grows much faster than N_C so that c becomes much larger than n_c, i.e., $n_c \ll c$. In this case, we get $n_c^2 \leq c_{\max}$. As a result, $cn_c^2 \leq c_{\max}^2$, and hence the complexity of the Expansion Phase becomes $\mathcal{O}(kc_{\max}^2)$.

Now we compute the complexity of the Refinement Phase, that is, of the MERGE-COMS() procedure. Observe that the **repeat-until** loop runs k times in the worst case. Now look at the Lines 5, 7 and 9, each of which requires cc_i steps, where $c_i = |C_i|$. Clearly, $cc_i \leq c_{\max}^2$. Now observe that $|L_H| \leq |L_N| \leq |N_C| = n_c$. So, the nested **for** loop (Lines 11–16) runs cn_c times. Finally, the operation at Line 19 takes c steps. Thus the complexity of the Expansion Phase is $\mathcal{O}(kc_{\max}^2 + kcn_c + c) = \mathcal{O}(kc_{\max}^2)$. Consequently, the complexity of IC-LCD is also $\mathcal{O}(kc_{\max}^2)$. Further, if the input network has n nodes, then it can be seen that $\mathcal{O}(kc_{\max}) = \mathcal{O}(n)$. Consequently, the complexity of IC-LCD reduces to $\mathcal{O}(nc_{\max})$.

7 Discussion

In the previous sections we have seen the performance of IC-LCD on synthetic as well as real-world networks. On synthetic networks, IC-LCD gives results more accurate than the other algorithms when the node memberships are high. In real-world networks, the performance of IC-LCD is measured by the quality indicators Q_{ov} and Q_{wo}. We find that IC-LCD gives competitive performance with the baseline algorithms on almost all the quality indicators. If we assess from the point of view of quality and stability, IC-LCD is a suitable choice for overlapping community detection. The running time plots on synthetic networks show that IC-LCD is slightly slower than COPRA and DEMON, but much faster than both OSLOM and SLPA. Surprisingly, DEMON is the slowest algorithm on the real networks DBLP and AMAZON. (See Fig. 4.) Our theoretical estimate of the complexity of IC-LCD is thus in line with practical situations, and hence the algorithm scales for large networks.

Another interesting fact about IC-LCD is that its MERGE-COMS() procedure is stand-alone, and therefore it can be used as a refinement step for many seed expansion based algorithms which are fast but produce unstable covers. However, this requires a systematic study.

8 Conclusion

We have proposed the algorithm IC-LCD which employs the notion of node and edge interaction coefficient to reveal the community structure with pervasive overlaps in complex networks. IC-LCD has time complexity $\mathcal{O}(nc_{\max})$, where n is the number of nodes in the input network, and $c_{\max}$ is the maximum size of the communities detected. We have analysed the performance of IC-LCD on a number of quality indicators, and compared the results with a few well-known algorithms for overlapping community detection. The results on synthetic networks suggest that when the node memberships are low IC-LCD performs moderately and as the node memberships increase it begins outperforming the other algorithms. Tests on real networks suggest that IC-LCD gives competitive performance in terms of quality with other algorithms. From the point of view of detecting the community structure with pervasive overlaps, IC-LCD gives desirable outcomes. Additionally its high speed makes it appropriate for application on large networks.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Here we shall illustrate the concepts of node and edge interaction coefficients, with the help of examples. Then we shall see how the seed expansion takes place using these coefficients.

Appendix A.1: Illustration of node and edge interaction coefficients

Consider the graph given below (Fig. 5).

Take $C_1 = \{u_0\}$, and $C_2 = \{v_0\}$. Let us expand C_1, and C_2 using the node interaction coefficient ξ_{node}, taking $\xi_0 = 0.5$.

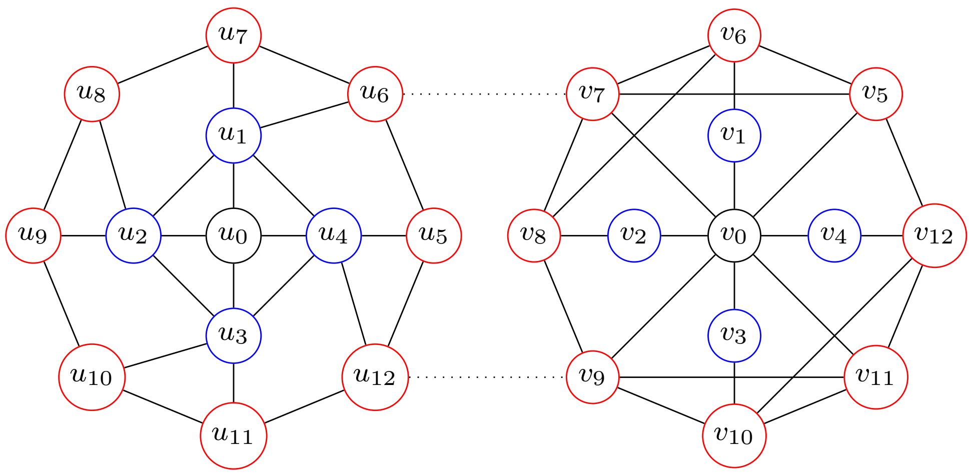


Fig. 5 Expansion of the seeds u_0 and v_0

Note that $N_{C_1} = \{u_1, u_2, u_3, u_4\}$ and $N_{C_2} = \{v_1, v_2, v_3, v_4\}$. For each $1 \leq i \leq 4$, we have

$$\xi_{\text{node}}(u_i, C_1) = \frac{1}{5} < \xi_0.$$

This means C_1 would not expand. However, for each $1 \leq i \leq 4$ we have

$$\xi_{\text{node}}(v_i, C_2) = \frac{1}{2} = \xi_0,$$

which means C_2 can expand to all its neighbours, and becomes $C_2 = \{v_0, v_1, v_2, v_3, v_4\}$. So, $N_{C_2} = \{v_5, v_6, \ldots, v_{12}\}$. Now for each $5 \leq i \leq 12$, we have

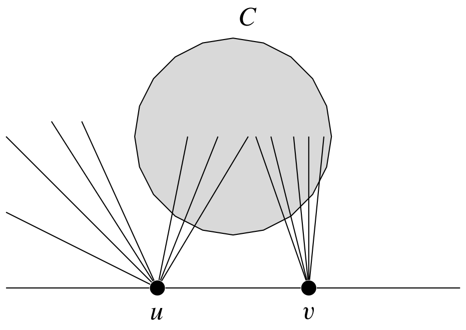
$$\xi_{\text{node}}(v_i, C_2) = \frac{1}{4} < \xi_0,$$

which means C_2 cannot be expanded further. The case we have considered is specific. But, it captures the two important types of seeds which are—highly clustered, and lowly clustered. The same strategy, such as the one based on node interaction coefficient, will not work for the expansion of both the kinds of seeds.

Therefore, we have introduced the concept of edge interaction coefficient ξ_{edge}. An edge e_{uv} essentially interacts with a subgraph C through its endpoints u and v. To arrive at a formula for $\xi_{\text{edge}}(e_{uv}, C)$, we use the following assumption: If both u and v have more neighbours in C, then e_{uv} interacts with C highly. So, look at the quantity

$$\min\big\{|N_u \cap V_C|, |N_v \cap V_C|\big\}.$$

To normalise it we can divide it by the minimum or the maximum of the degrees of u and v. Moreover, we wish $\xi_{\text{edge}}(e_{uv}, C)$ to be highest when $N_u \subseteq V_C \setminus \{v\}$, $N_v \subseteq V_C \setminus \{u\}$, and $d_u = d_v$. Keeping, all these requirements, we get Eq. (2). It is apparent that $0 \leq \xi_{\text{edge}} \leq 1$. It can be seen that $\xi_{\text{edge}}(e_{uv}, C) = 1$ iff $d_u = d_v$ and $|N_u \cap V_C| = |N_v \cap V_C|$. In the denominator of Eq. (2) we have taken $\max\{d_u, d_v\}$ instead of $\min\{d_u, d_v\}$. To see why let us look at the case given in the picture below.



The node u has 3 neighbours in C, and 6 neighbours outside C. So, $\xi_{\text{node}}(u, C) = 1/3$ which is much smaller than the threshold ξ_0. Consequently, u must not join C in any case. However, v has 5 neighbours in C and just 2 neighbours outside C. So, v would surely join C. Now let us compute the node interaction coefficient of u with $C \cup \{v\}$. We have

$$\xi_{\text{node}}(u, C \cup \{v\}) = \frac{4}{9} < \xi_0$$

Thus u would not join $C \cup \{v\}$ too. Consider, now the case when $\min\{d_u, d_v\}$ is the numerator in Eq. (2). Then

$$\xi_{\text{edge}}(e_{uv}, C) = \frac{1+3}{7} = \frac{4}{7} > \xi_0$$

In this case u joins C. Thus $\min\{d_u, d_v\}$ is *not an appropriate choice* for the denominator in Eq. (2).

Appendix A.2: Illustration of seed expansion phase

We illustrate the GET-NEW-NODES() procedure through examples. Note that we do not specify any criterion for select-

Fig. 6 A network with community structure

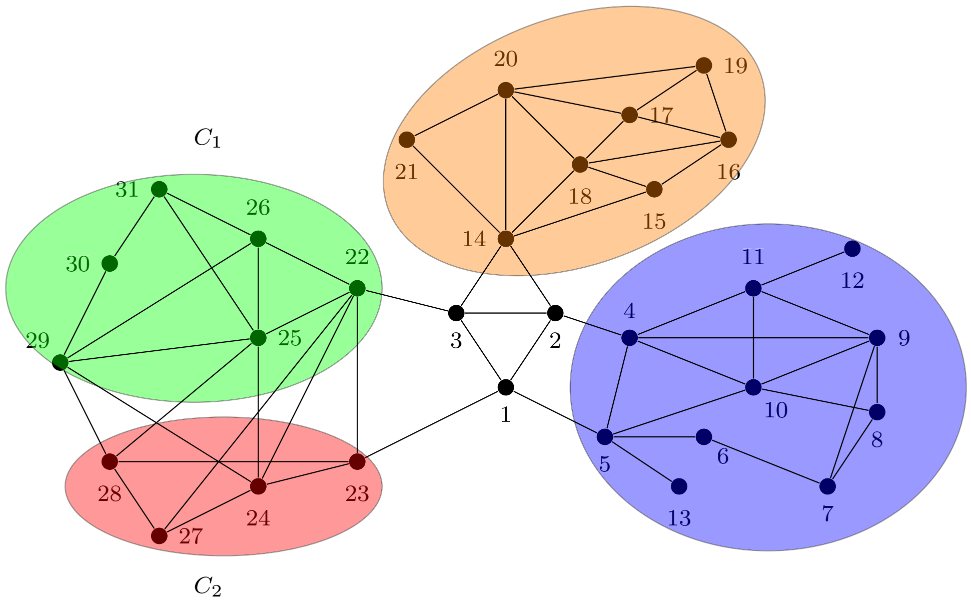


Table 6 Before augmentation step, $V_{\text{new}} = \varnothing$ for $C = \{30, 31\}$ in Fig. 6

$u \in N_C$	$\xi_{\text{node}}(u, C)$	$v \in N_u \cap N_C, v > u$	$\xi_{\text{edge}}((u, v), C)$
25	$\frac{1}{6}$	26	$\frac{1}{3}$
		29	$\frac{1}{3}$
26	$\frac{1}{4}$	29	$\frac{2}{5}$
29	$\frac{1}{5}$		

Table 7 Before augmentation step, $V_{\text{new}} = \varnothing$ for $C = \{1, 2, 3\}$ in Fig. 6

$u \in N_C$	$\xi_{\text{node}}(u, C)$	$v \in N_u \cap N_C, v > u$	$\xi_{\text{edge}}((u, v), C)$
4	$\frac{1}{5}$	5	$\frac{2}{5}$
5	$\frac{1}{5}$		
14	$\frac{1}{3}$		
22	$\frac{1}{6}$	23	$\frac{1}{3}$
23	$\frac{1}{4}$		

ing seeds, so any node may serve as a seed. Then it may well happen that certain seeds, especially the low degree nodes, stop expanding after growing to few nodes, or do not expand at all. Let us consider a few examples assuming that $\xi_0 = 0.5$ and $n_{\min} = 4$.

Example 1 Consider the graph given in Fig. 6.

Let $C = \{30, 31\}$. Then $N_C = \{25, 26, 29\}$. In order to compute V_{new}, the steps followed before the augmentation step are listed in Table 6. No node of N_C is added to V_{new}, leaving V_{new} empty. On the other hand, during the augmentation step, we find that $\xi_{\text{node}}(25, C \cup N_C) = 1/2$, $\xi_{\text{node}}(26, C \cup N_C) = 3/4$ and $\xi_{\text{node}}(29, C \cup N_C) = 3/5$, which makes $V_{\text{new}} = \{25, 26, 29\}$.

Example 2 This time consider the subgraph $C = \{1, 2, 3\}$ in the graph given in Fig. 6. Here $N_C = \{4, 5, 14, 22, 23\}$. Then before the augmentation step V_{new} remains empty as shown in Table 7. However, in this case even the augmentation step does not help, as $\xi_{\text{node}}(u, C \cup N_C) < \xi_0$ for all $u \in N_C$, and so, $V_{\text{new}} = \varnothing$. Thus the subgraph C is not expandable to a full community. Such groups of nodes are likely to join multiple communities and form the basis for pervasive overlaps.

References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 U.S. election: divided they blog. In: Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD'05, pp. 36–43. ACM, New York (2005). https://doi.org/10.1145/1134271.1134277
2. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature **466**(7307), 761–764 (2010). https://doi.org/10.1038/nature09182
3. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., Chen, R.: Topological structure analysis of the protein-protein interaction network in budding yeast. Nucleic Acids Res. **31**(9), 2443–2450 (2003)
4. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: DEMON: a local-first discovery method for overlapping communities. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'12, pp. 615–623. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2339530.2339630
5. Costa, G., Ortale, R.: Topic-aware joint analysis of overlapping communities and roles in social media. Int. J. Data Sci. Anal. **9**(4), 415–429 (2020)

6. Ding, Z., Zhang, X., Sun, D., Luo, B.: Overlapping community detection based on network decomposition. Sci. Rep. **6**, 24115 (2016). https://doi.org/10.1038/srep24115
7. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters (1973)
8. Fan, X., Cao, L., Da Xu, R.Y.: Dynamic infinite mixed-membership stochastic blockmodel. IEEE Trans. Neural Netw. Learn. Syst. **26**(9), 2072–2085 (2015). https://doi.org/10.1109/TNNLS.2014.2369374
9. Flake, G.W., Lawrence, S., Giles, C.L.: Efficient identification of web communities. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'00, pp. 150–160. ACM, New York (2000). https://doi.org/10.1145/347090.347121
10. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010). https://doi.org/10.1016/j.physrep.2009.11.002
11. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. PNAS **104**(1), 36–41 (2007). https://doi.org/10.1073/pnas.0605965104
12. Fortunato, S., Hric, D.: Community detection in networks: a user guide. Phys. Rep. **659**, 1–44 (2016). https://doi.org/10.1016/j.physrep.2016.09.002. Community detection in networks: A user guide
13. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002). https://doi.org/10.1073/pnas.122653799
14. Gleiser, P.M., Danon, L.: Community structure in jazz. Advs. Complex Syst. **06**(04), 565–573 (2003). https://doi.org/10.1142/S0219525903001067
15. Gregory, S.: An algorithm to find overlapping community structure in networks. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 91–102. Springer, Berlin (2007)
16. Gregory, S.: A fast algorithm to find overlapping communities in networks. In: Daelemans, W., Goethals, B., Morik, K. (eds.) Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Science, pp. 408–423. Springer, Berlin (2008)
17. Gregory, S.: Finding overlapping communities in networks by label propagation. New J. Phys. **12**(10), 103018 (2010). https://doi.org/10.1088/1367-2630/12/10/103018
18. Guimerá, R., Amaral, L.A.N.: Cartography of complex networks: modules and universal roles. J. Stat. Mech. **2005**(P02001), P02001-1–P02001-13 (2005). https://doi.org/10.1088/1742-5468/2005/02/P02001
19. Havemann, F., Heinz, M., Struck, A., Gläser, J.: Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. J. Stat. Mech. **2011**(01), P01023 (2011). https://doi.org/10.1088/1742-5468/2011/01/P01023
20. He, D., Jin, D., Chen, Z., Zhang, W.: Identification of hybrid node and link communities in complex networks. Sci. Rep. **5**, 8638 (2015). https://doi.org/10.1038/srep08638
21. Knuth, D.E.: The Standford Graph-Base: A Platform for Combinatorial Computing. Addition-Wesley, Reading (1993)
22. Kumar, P., Dohare, R.: A neighborhood proximity based algorithm for overlapping community structure detection in weighted networks. Front. Comput. Sci. (2019). https://doi.org/10.1007/s11704-019-8098-0
23. Kumar, P., Dohare, R.: Formalising and detecting community structures in real world complex networks. J. Syst. Sci. Complex. **34**, 180–205 (2021). https://doi.org/10.1007/s11424-020-9252-3
24. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E **80**(1), 016118 (2009). https://doi.org/10.1103/PhysRevE.80.016118
25. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys. **11**(3), 033015 (2009). https://doi.org/10.1088/1367-2630/11/3/033015
26. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. PLoS ONE **6**(4) (2011). https://doi.org/10.1371/journal.pone.0018961
27. Lázár, A., Abel, D., Vicsek, T.: Modularity measure of networks with overlapping communities. EPL (Europhys. Lett.) **90**(1), 18001 (2010). https://doi.org/10.1209/0295-5075/90/18001
28. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion. arXiv:1002.1827 [physics] (2010)
29. Leskovec, J., Krevl, A.: SNAP Datasets: stanford large network dataset collection. http://snap.stanford.edu/data (2014)
30. Lu, Z., Sun, X., Wen, Y., Cao, G., Porta, T.L.: Algorithms and applications for community detection in weighted networks. IEEE Trans. Parallel Distrib. Syst. **26**(11), 2916–2926 (2015). https://doi.org/10.1109/TPDS.2014.2370031
31. McDaid, A., Hurley, N.: Detecting highly overlapping communities with model-based overlapping seed expansion. In: 2010 International Conference on Advances in Social Networks Analysis and Mining, pp. 112–119 (2010). https://doi.org/10.1109/ASONAM.2010.77
32. Newman, M.E.J.: Network datasets from Newman. http://www-personal.umich.edu/~mejn/netdata/
33. Newman, M.E.J.: The structure of scientific collaboration networks. PNAS **98**(2), 404–409 (2001). https://doi.org/10.1073/pnas.98.2.404
34. Newman, M.E.J.: Detecting community structure in networks. Eur. Phys. J. B **38**(2), 321–330 (2004). https://doi.org/10.1140/epjb/e2004-00124-y
35. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004). https://doi.org/10.1103/PhysRevE.69.066133
36. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006). https://doi.org/10.1103/PhysRevE.74.036104
37. Newman, M.E.J.: Modularity and community structure in networks. PNAS **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103
38. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004). https://doi.org/10.1103/PhysRevE.69.026113
39. Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M.: Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech. **2009**(03), P03024 (2009). https://doi.org/10.1088/1742-5468/2009/03/P03024
40. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814–818 (2005). https://doi.org/10.1038/nature03607
41. Qi, Y., Ge, H.: Modularity and dynamics of cellular networks. PLoS Comput. Biol. **2**(12), e174 (2006). https://doi.org/10.1371/journal.pcbi.0020174
42. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007). https://doi.org/10.1103/PhysRevE.76.036106
43. Reichardt, J., Bornholdt, S.: Detecting fuzzy community structures in complex networks with a Potts model. Phys. Rev. Lett. **93**(21), 218701 (2004)
44. Rossetti, G., Milli, L., Cazabet, R.: CDLIB: a python library to extract, compare and evaluate communities from complex networks. Appl. Netw. Sci. **4**(1), 52 (2019). https://doi.org/10.1007/s41109-019-0165-9

45. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A **388**(8), 1706–1712 (2009). https://doi.org/10.1016/j.physa.2008.12.021
46. Sun, H., Jia, X., Huang, R., Wang, P., Wang, C., Huang, J.: Distance dynamics based overlapping semantic community detection for node-attributed networks. Comput. Intell. (2020)
47. Sun, H., Liu, J., Huang, J., Wang, G., Jia, X., Song, Q.: LinkLPA: a link-based label propagation algorithm for overlapping community detection in networks. Comput. Intell. **33**(2), 308–331 (2017). https://doi.org/10.1111/coin.12087
48. Tripathi, B., Parthasarathy, S., Sinha, H., Raman, K., Ravindran, B.: Adapting community detection algorithms for disease module identification in heterogeneous biological networks. Front. Genet. **10**, 164 (2019)
49. Wang, Y., Bu, Z., Yang, H., Li, H.J., Cao, J.: An effective and scalable overlapping community detection approach: integrating social identity model and game theory. Appl. Math. Comput. **390**, 125601 (2021). https://doi.org/10.1016/j.amc.2020.125601
50. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature **393**(6684), 440–442 (1998). https://doi.org/10.1038/30918
51. Wei, Y., Singh, L., Gallagher, B., Buttler, D.: Overlapping target event and story line detection of online newspaper articles. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 222–232 (2016). https://doi.org/10.1109/DSAA.2016.30
52. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proceedings of the 2005 SIAM International Conference on Data Mining, Proceedings, pp. 274–285. Society for Industrial and Applied Mathematics (2005)
53. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: the state-of-the-art and comparative study. ACM Comput. Surv. **45**(4), 43:1–43:35 (2013). https://doi.org/10.1145/2501654.2501657
54. Xie, J., Szymanski, B.K., Liu, X.: SLPA: uncovering overlapping communities in social networks via a speaker–listener interaction dynamic process. pp. 344–349. IEEE (2011). https://doi.org/10.1109/ICDMW.2011.154
55. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM'13, pp. 587–596. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2433396.2433471
56. Yang, J., Leskovec, J.: Overlapping communities explain core–periphery organization of networks. Proc. IEEE **102**(12), 1892–1902 (2014). https://doi.org/10.1109/JPROC.2014.2364018
57. Yang, J., McAuley, J.J., Leskovec, J.: Community Detection in Networks with Node Attributes (2013)
58. Zhang, F., Ma, A., Wang, Z., Ma, Q., Liu, B., Huang, L., Wang, Y.: A central edge selection based overlapping community detection algorithm for the detection of overlapping structures in protein–protein interaction networks. Molecules **23**(10), 2633 (2018)
59. Zhang, S., Wang, R.S., Zhang, X.S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A **374**(1), 483–490 (2007). https://doi.org/10.1016/j.physa.2006.07.023
60. Zhang, Y., Yin, D., Wu, B., Long, F., Cui, Y., Bian, X.: Plinkshrink: a parallel overlapping community detection algorithm with link-graph for large networks. Soc. Netw. Anal. Min. **9**(1), 66 (2019)