JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 499-515 (2021)
DOI: 10.6688/JISE.202105_37(3).0001

Local Community Detection by Local Structure Expansion and Exploring the Local Communities for Target Nodes in Complex Networks

HAO-SHANG MA, SHIOU-CHI LI, ZHI-JIA JIAN,
YOU-HUA KUO AND JEN-WEI HUANG
Department of Electrical Engineering
Institute of Computer and Communication Engineering
National Cheng Kung University
Tainan City, 701 Taiwan
E-mail: {ablove904; sp822543; dream880347; wilson820817}@gmail.com;
jwhuang@mail.ncku.edu.tw

Most algorithms for local community detection select a seed node using a greedy algorithm and then expand it into a community using optimization functions. This paper presents a novel approach to community detection based on local expansion. The proposed Local Community Detection via LOcal Structure Expansion (CLOSE) algorithm features a novel connective function, which identifies a source node in the center of a highly-connected component of a graph. The CLOSE algorithm also selects a group of nodes rather than a single node as a seed for local community expansion, which facilitates the selection of a community suitable for the hub node. We also developed a system by which to identify the most suitable source nodes for given target nodes, referred to as Exploring Local Communities of Target Nodes (ELCTN). The performance of CLOSE and ELCTN was compared with that of state-of-the-art methods in experiments using synthetic networks generated using the Lancichinetti-Fortunato-Radicchi benchmark as well as real-world networks. Both algorithms outperformed previous methods in terms of accuracy and modularity.

Keywords: complex networks, local community detection, source-node selection, neighboring group expansion, communities identification for target nodes

1. INTRODUCTION

There is growing interest in the study of complex networks such as the world-wide web, biological and interaction networks, citation networks, online social networks, and metabolic networks. Complex networks comprise individual elements as well as the relationships between them as well. For instance, a social network is a social structure comprising a group of users as well as the social interactions that occur among users. This type of network can be represented as graph $G = (V, E)$, where node set V corresponds to individuals, and edge set E represents the relationships between individuals.

Within a complex network, the term "community" refers to a group of nodes that form connections of greater density than those outside the community [1]. Community detection is essentially the process of identifying community structures within complex networks [2]. Existing community detection algorithms require global knowledge (*i.e.*,

Received August 16, 2020; revised October 18, 2020; accepted November 18, 2020.
Communicated by Mong-Fong Horng.

pertaining to the entire network) [1, 3, 4], which imposes a heavy computational burden. Thus, researchers have been shifting their focus toward local community detection, which requires only local knowledge (*i.e.*, pertaining to structures adjacent to a particular node) [5, 6, 7, 8, 9]. In situations where the availability of information is limited, local community detection has proven more efficient in detecting communities within complex networks.

Most existing algorithms for local community expansion can be divided into two parts: source-node selection and community extension [10]. In source-node selection, existing methods find the source node at the center of the community to ensure that the analysis is close to the ground truth [5, 11]. The local community related to the source node is expanded to include nodes with a close relationship to the source node. The expansion process is terminated when none of the remaining nodes satisfy conditions for inclusion in the community.

Nonetheless, there remain a number of challenges to local community detection. Previous works [12, 13, 14] used greedy methods to maximize local modularity measurements by incorporating vertices in communities one at a time. Greedy algorithms for local community detection select in each iteration the neighbor of the seed with the highest local modularity. Note that greedy algorithms can only search for local optimal solutions; therefore, selecting a suitable starting node is crucial to the discovery of community structures. Note also that local community detection is based on the assumption that we do not have information related to the entire network. This paper also focuses on a scenario in which we need only determine communities for given target nodes, rather than all of the nodes in the network. Thus, the selection of source nodes and the expansion of communities should be related to the target nodes. To the best of our knowledge, no existing method is able to detect local communities for specific target nodes.

This paper presents two algorithms for local community detection. Local Community Detection by LOcal Structure Expansion (CLOSE) performs local community detection for all nodes. Exploring Local Communities for Target Nodes (ELCTN) performs local community detection for specific target nodes. CLOSE is based on a novel source-node selection scheme in which we define a novel connective function (referred to as center prediction) to identify the node that represents the center of a community. Note that CLOSE selects a group of nodes instead of a single node as the seed for expansion. The CLOSE algorithm prevents the expansion of nodes at the fringes of the community, while simultaneously resolving the hub problem.

The ELCTN algorithm calculates a target center prediction (TCP) score with the aim of identifying the node with the strongest relation to the target node for use as a starting point from which to form a community.

In experiments, the CLOSE algorithm outperformed state-of-the-art algorithms in terms of accuracy and modularity for all types of real-world network and especially large networks involving dense graphs. ELCTN achieved the highest F-score and normalized mutual information (NMI) score in all types of network.

The main contributions of this paper are summarized as follows.

- We developed a source-node selection method in which a novel connective function is used to identify the node that is most strongly connected to its neighbors for use as a source node from which to begin the expansion of a community.

- We originated the idea of including a group of neighboring nodes in order to minimize the risk of including inappropriate nodes within the community.
- We investigated a scenario involving the identification of communities only for a given target node using local network information. We also developed a source-node selection method based on target center prediction.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 outlines the proposed CLOSE and ELCTN algorithms. Section 4 outlines the experiments used to assess the performance of the algorithms using real-world databases. Conclusions are drawn in Section 5.

2. RELATED WORKS

Local community detection is an important topic in the field of complex network analysis. Generally, local community detection is implemented in two phases: seed selection and community expansion. Section 2.1 provides an overview of the seed selection process. Section 2.2 examines previous works focusing on community expansion.

2.1 Seed Selection

The efficacy of local expansion depends heavily on the selection of an appropriate source node. Several source-node selection methods have been proposed. The local maximum degree method detects the node with highest degree from among the neighbors of the initial node [15]. Moradi *et al*. [11] proposed a seeding algorithm based on link prediction, wherein the node with the highest similarity score is selected as the seed node. The characteristics of the seed node depend on the specifics of the link prediction method. Gleich and Seshadhri [16] proposed a method that involves selecting the nodes with the local minimum conductance score within the neighborhood community. Xiaoyu *et al*. [5] proposed a core detection method based on node mass and node relation strength. Their method makes it possible to replace any seed node with the core member of greatest relevance to the target community. Instead of expanding a community from a single source node, Tabarzad [17] and Zhou [9] advocated detecting a community within a subgraph of high density. Tabarzad *et al*. [17] proposed the concept of a minimal cluster, which is the set of nodes that connect most closely to the initial node provided by the user. Zhou *et al*. [9] proposed the concept of an early community, which is a high-density subgraph within a network representing the community in embryonic form.

2.2 Community Expansion

Following seed selection, the nodes around the seed node are merged to form a community. Most existing methods perform community expansion using an objective function, such as the local modularity function [13], the L-metric [12], the M local modularity measurement [14], or the Quasi- Cliques function [18]. However, the one-by-one greedy addition of vertices to a community does not necessarily lead to a favorable solution. The local community metric L proposed by Chen *et al*. [12] was shown to outperform local modularity R. Tao *et al*. [14] advocated merging a node with the community as long as

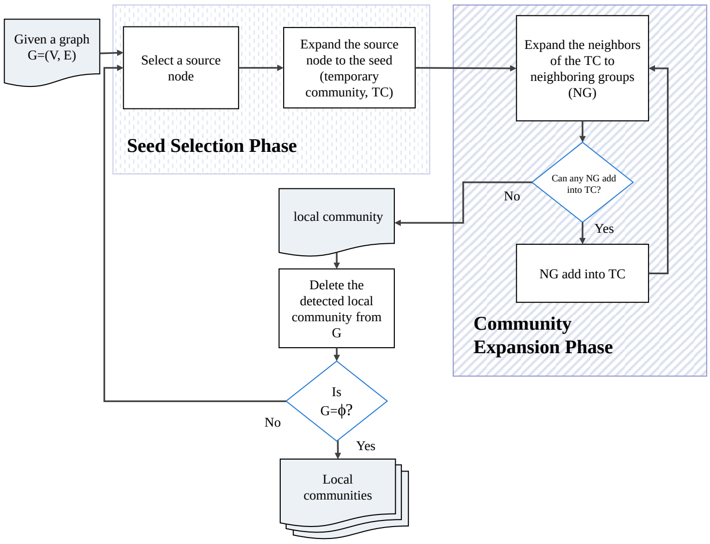


Fig. 1. System flow.

it connects more comprehensively to nodes within the community (*i.e.*, with more edges) than to nodes outside the community. Ding *et al*. [5] based community extension on node relational strength by measuring the degree of similarity between adjacent nodes (*i.e.*, between the community and its neighbors), where nodes with relational strength surpassing a given threshold are automatically merged into the community.

3. METHODOLOGY

In this section, we introduce the proposed algorithm local Community detection by LOcal Structure Expansion (CLOSE) based on our previous work [19]. A system flow for CLOSE is detailed in Section 3.1. We then discuss implementation of the proposed scheme for Exploring Local Communities of the Target Nodes (ELCTN).

3.1 System Architecture

Fig. 1 presents the system flow of CLOSE for network $G = (V, E)$. The seed selection phase employs a novel connective function, which identifies the source node in the center of a community and assembles a seed set through the expansion of source nodes. The implementation of community expansion involves two simultaneous actions: 1) combining neighbors of the temporary community to form neighboring groups and 2) combining the neighboring groups into a temporary community. These actions are repeated iteratively until the temporary community can no longer be expanded. CLOSE then selects another source node and forms a corresponding community. The overall process is repeated until all of the communities in the graph are discovered.

3.2 Seed Selection Phase

Community expansion involves 1) selecting a source node and 2) expanding the source node to form a seed set.

3.2.1 Select a source node

In this study, we developed a novel connective function referred to as center prediction (CP), which finds a suitable node to form the center of a community. *CP* is defined as follows:

$$CP(v) = \sum_{m \in \Gamma(v), u \in \Gamma(v), (m,u) \in E} Sim(m,u), \tag{1}$$

where $\Gamma(v)$ denotes the neighbors of v and $Sim(m,u)$ represents the expected connection value between m and u. $Sim(m,u)$ can be defined as $Sim(m,u) = \sum_{w \in \Gamma(m) \cap \Gamma(u)} \frac{1}{k_w}$, where w is a common neighbor of m and u, and k_w is the degree of w. Node v with a high *CP* score indicates that the neighbors of node v are tightly connected to one another. Among all of the neighbors, the node with the highest *CP* is the node with the highest connection density. This node (referred to as the local maximum *CP*) forms the center of the community.

3.2.2 Expand the source node to a seed set

The seed is defined as a set of nodes located in the vicinity of the selected source node. The neighbors that are common to node v and source node v_s are also recruited into the seed. Thus, seed expansion from the source node v_s can be defined as

$$Seed(v_s) = (\Gamma(v_s) \cap \Gamma(v)) \cup \{v_s\} \cup \{v\}, \tag{2}$$

where $\Gamma(v_s)$ is the set containing all of the neighbors of v_s. Following seed expansion, the resulting seed set is denoted as a temporary community (TC).

3.3 Community Expansion Phase

Community expansion is implemented in two simultaneous steps: 1) expanding neighbors of the temporary community to form neighboring groups and 2) recruiting suitable neighboring groups into the temporary community. Each of these steps is detailed in the following sub-sections.

3.3.1 Expanding neighbors of temporary community to form neighboring groups

Nodes adjacent to the TC are expanded to form neighboring groups. Note that we do not expand all of the neighbors as this could lead to the aggregation of multiple communities. We first assess each neighboring node n of TC to determine whether they are suitable for expansion.

In Fig. 2, we consider the following example: Node 5 is a neighbor n of TC. The neighbors of n can be partitioned into three cases: 1) nodes belonging to TC; 2) nodes belonging to first-level neighbors of TC; and 3) nodes that are neither in TC nor first-level neighbors of TC. Nodes q and r belong to Case 1 (neighbors of node 5 and belonging to TC). Nodes 1, 2, and 7 represent Case 2 (neighbors of node 5 and first-level neighbors of

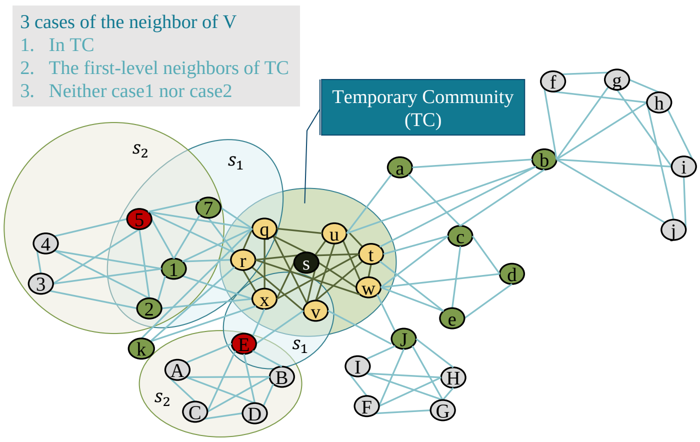


Fig. 2. Checking the expansion of each neighboring node.

TC). Nodes 3 and 4 are Case 3 (neighbors of node 5 but not in TC nor first-level neighbors of TC).

In most situations, a community is denser near the center and sparser near the fringes. We define the nodes in Cases 1 and 2 as set s_1, and the nodes in Cases 2 and 3 as set s_2. Set s_1 represents the group of nodes close to TC. Set s_2 represents the group of nodes with one or two degrees of separation from TC. In cases where the connection density in s_1 is higher than in s_2, we recruit the neighbor of TC into the neighboring group. Otherwise, we leave the node alone. Follow [19], we use the *Quasi-Clique Ratio*, *QCR*, to identify the density extent and propose the 2-*level Quasi-Clique Ratio*, $2\_QCR(n)$, to identify the density extent between two different sets of nodes. The formula is defined as $2\_QCR(n) = \frac{QC(s_1)}{QC(s_2)}$. The node n is a neighbor of TC. If s_1 is denser than s_2, then $2\_QCR(n)$ is larger than 1. We then expand node n into the neighboring group ng_n with all neighbors except those in TC. The neighboring group expanded by n is denoted as $ng_n = \{\{n\} \cup \{u|u \in \Gamma(n), MAX(|\Gamma(u) \cap \Gamma(n)|)\} \cup \{\Gamma(u) \cap \Gamma(n)\}\} \setminus TC$. Otherwise, the neighboring group of n contains only n. In Fig. 2, $QCR(s_1)$ of node 5 is larger than $QCR(s_2)$ of node 5. Thus, node 5 is recruited into neighboring group ng_5 with node 1 and their common neighbors $1,2,3,4$ and 7 (see Fig. 3). By contrast, $2\_QCR(E)$ is less than 1; therefore, node E is not recruited with its neighbors into the neighboring group (see Fig. 3). The results of the expansion of neighboring nodes of this example are presented in Fig. 3.

3.3.2 Inclusion of suitable neighboring groups within temporary community

In this step, a label propagation algorithm [20] is used to determine whether the neighboring groups should be merged into the TC. Note that this approach makes it possible to evaluate all of the neighboring groups simultaneously. Initially, each node carries the label that denotes the community to which it belongs. In each iteration of label diffusion, all of the nodes pass their labels to their neighbors. After receiving all of the labels from all of their neighbors, each node changes its own label to the one that it encountered

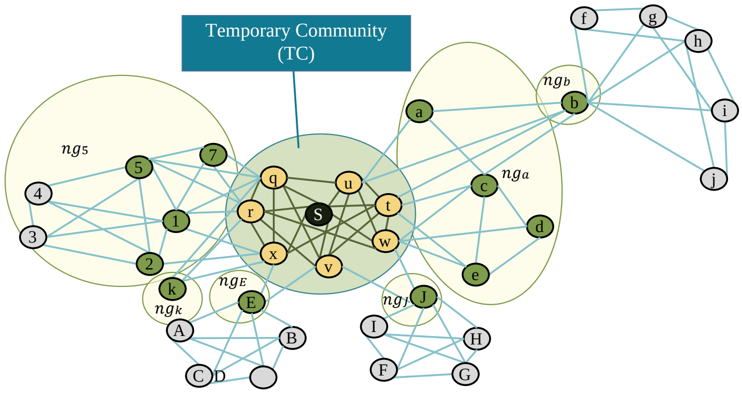


Fig. 3. Neighboring group expansion.

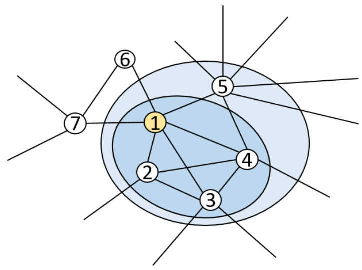


Fig. 4. The example of neighborhood cluster.

most frequently. After several iterations, the label carried by a node indicates the community to which it belongs. Label propagation is highly efficient in terms of execution time; however, it does not necessarily produce consistent solutions. We sought to overcome this by performing label propagation several times and accumulating all of the labels gathered from each run as the final result. We applied label propagation to the subgraph that contains nodes in the TC and all of the neighboring groups of the TC. Initially, each node carries the label denoting the neighboring group to which the node belongs.

In our implementation, the nodes in the TC carry the label -1. Following several iterations, the label that occurs most frequently in each neighboring group is selected to represent the group. Running the process several times generates several results for each group. We then identify the label that occurs most frequently in the neighboring groups to function as its final label. The final label of a neighboring group is -1, if most of the labels are gathered from the TC. In this situation, the neighboring group is then merged into the TC.

Following Step 2, community expansion continues until there are no neighboring groups to be merged into the TC. The local community expanded from the source node is then output as the solution. If the user wishes to find other local communities, then seed selection and community expansion are repeated accordingly.

3.4 Exploring Local Communities associated with Target Nodes

In this section, we examine the scenario in which it is necessary to detect communities for designated target nodes rather than all of the nodes in the network. We developed ELCTN algorithm to solve this problem. For given target nodes, it is preferable to identify the most strongly-related source nodes from which to form the community structure. We first define a neighborhood cluster structure (*i.e.*, a set of nodes neighboring the target node). Note that the nodes in a neighborhood cluster should also be closely related to the target nodes. A strong relationship is identified by a high local modularity [7]. For example, node 5 in Fig. 4 is not closely related to target node 1, based on the fact that the out-degree of node 5 (relative to the neighborhood cluster of node 1) is five compared to an in-degree of two. We developed a function referred to as target center prediction (TCP) to quantify the probability of a given node becoming the center of the community containing the target node. TCP employs neighborhood cluster density (NCD) and the neighborhood cluster similarity relation (NCSR). The NCD value indicates the density of a given cluster neighboring the target node; it is formulated as follows:

$$NCD(m) = \frac{2 \mid \{(u,v) \mid u,v \in NC(m), (u,v) \in E_{NC(m)}\} \mid}{\mid NC(m) \mid (\mid NC(m) \mid -1)}, \tag{3}$$

where $NC(m)$ is the neighborhood cluster of node m and $E_{NC(m)}$ indicates the edges in neighborhood cluster of node m. A higher density indicates that the node in question has an elevated likelihood of becoming the center of a community. The NCSR value is another criterion used to establish the similarity of nodes in the neighborhood cluster. A higher NCSR score indicates that the nodes in the neighborhood cluster are strongly related to one another. NCSR is formulated as follows:

$$NCSR(m) = \sum_{u,v \in NC(m), (u,v) \in E_{NC(m)}} NS(u,v) \tag{4}$$

where $NC(m)$ is a neighborhood cluster of node m and $E_{NC(m)}$ refers to edges in node m of neighborhood cluster $NC(m)$. We can calculate $NS(u,v) = \sum_{c \in N(u) \cap N(v)} \frac{1}{|N(c)|}$, where $N(c)$ is the number of nodes neighboring node c. The TCP score is obtained as follows: $TCP(m) = NCD(m) \times NCSR(m)$.

After calculating the TCP for the neighbors of target node, we traverse the neighbors of the target node. ELCTN selects the source node from which to form a community based on three conditions. The conditions are as follows: 1) The TCP score of the neighbor must exceed the TCP score of the original node. 2) The neighborhood cluster must contain the original node. 3) The overlapping rate between the neighborhood cluster of neighbor and the neighborhood cluster of original node, their neighborhood cluster overlapping rate (NCOR) score must satisfy the user defined threshold θ. NCOR is defined as follows,

$$NCOR(NC(u), NC(v)) = \frac{\mid NC(u) \mid \cap \mid NC(v) \mid}{\mid NC(u) \mid \cup \mid NC(v) \mid}. \tag{5}$$

The first condition is used to identify a node suitable for a community center. The second and third condition ensures the resulting community contains the target node. ELCTN iteratively searches for new source nodes until none of the remaining neighbors can be changed. Finally, ELCTN obtains source nodes with a strong relation to the target node near the center of the community.

4. EXPERIMENTS

In this section, we discuss the experiments conducted to verify the proposed approaches. A description of the data is presented in Section 4.1. The measurements used to verify the results are presented in Section 4.2. The results obtained in the current study are compared with those obtained in previous works in Section 4.3.

4.1 Data Description

The proposed algorithms were evaluated in experiments using synthetic and real-world networks.

4.1.1 Synthetic networks

We use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark [21] to generate synthetic complex networks. The advantage of LFR is the fact that it accounts for heterogeneity in the distributions of node degrees and community sizes.

We generated networks with the number of nodes set at 5,000, 10,000, and 50,000. Regardless of the number of nodes, LFR was configured to generate three types of graphs: sparse (**S**), medium (**M**) and dense (**D**).

4.1.2 Real world data

Real-world networks are available from UCI Machine Learning Repository[1] and Stanford large network dataset collection[2]. Table 1 lists the statistics of five networks: Karate, Club Football, Polbooks, Amazon, and DBLP.

Karate [22]: refers to a social network formed from the members of a karate club over a three-year period (1970 – 1972). The edges between pairs of members indicate interactions outside the club.

American college football [23]: refers to a network of American football games played between Division IA colleges during the fall season in 2000. The vertices in the network represent teams, and the edges represent regular-season games between two teams.

Polbooks[3]: refers to a network of books related to United States (US) politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com. Edges between books represent the frequent co-purchasing of books by the same buyer.

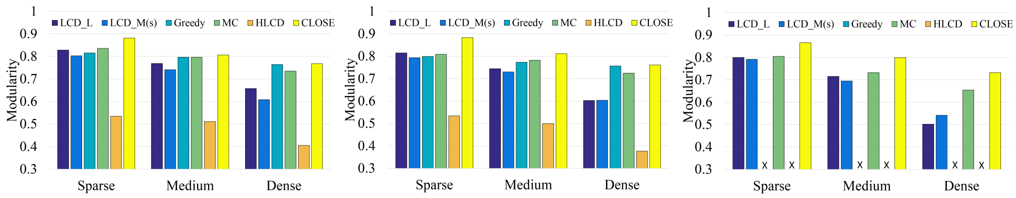
Amazon [24]: refers to products and the co-purchasing relationship between pairs of products.

DBLP [25]: refers to a co-authorship network in which two authors are deemed to be connected if they publish at least one paper together.

[1] https://archive.ics.uci.edu/ml/datasets.php
[2] http://snap.stanford.edu/data/
[3] http://www.orgnet.com

Table 1. Real world dataset, where AD: average degree; CC: clustering coefficient.

Data Sets	|V|	|E|	#Class	AD	CC
Karate Club	34	78	2	4.588	0.255
FootBall	115	613	11	2.665	0.403
Polbooks	105	441	3	8.400	0.488
Amazon	334863	925872	75149	1.382	0.3967
DBLP	317080	1049866	13477	6.622	0.632



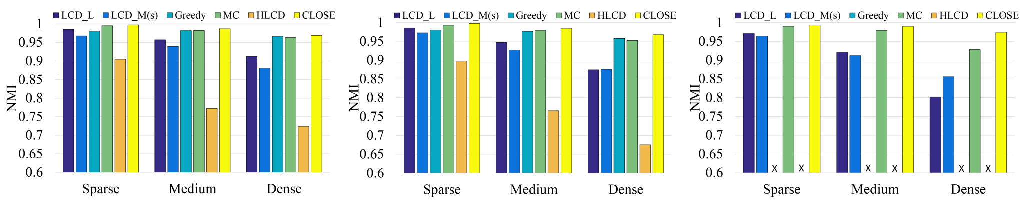
(a) 5,000 nodes. (b) 10,000 nodes. (c) 50,000 nodes.

Fig. 5. Modularity for different algorithms in synthetic networks.

4.2 Measurements

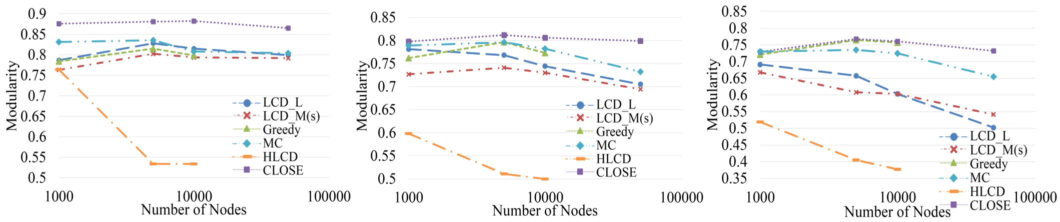
In the experiments, the performance of the CLOSE algorithm was evaluated using modularity and normalized mutual information (NMI). The quality of communities detected from source nodes selected by ELCTN was evaluated using NMI. We also used the F-score and cover-rate to evaluate the correctness of detected communities based on target nodes. Note that the assessment of ELCTN using NMI involved a comparison of the detected community and a real-world community that included the target node. Cases where the target node did not appear in the detected community were severely penalized in terms of NMI score. We believe this is a practical approach to performance verification.

Modularity [26] Modularity is used to measure the accuracy with which a network is partitioned into communities without the need for ground-truth data Modularity deals only with connections within a community and connections between communities. Networks with high modularity present dense connections between the nodes within communities, but sparse connections between the nodes in different communities. Modularity refers to



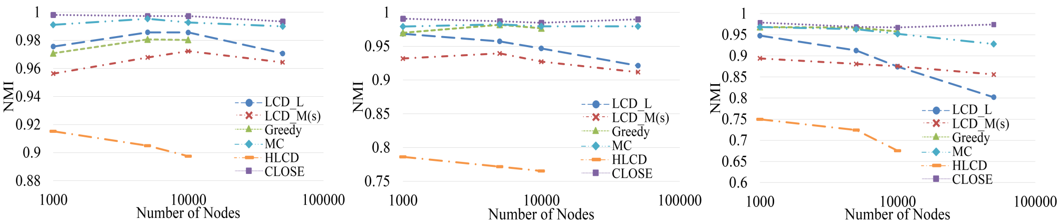
(a) 5,000 nodes. (b) 10,000 nodes. (c) 50,000 nodes.

Fig. 6. NMI for different algorithms in synthetic networks.



(a) Sparse graph. (b) Medium graph. (c) Dense graph.

Fig. 7. Modularity of different densities in synthetic networks.



(a) Sparse graph. (b) Medium graph. (c) Dense graph.

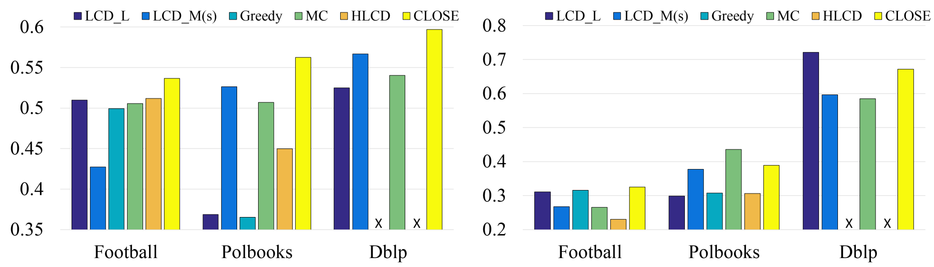
Fig. 8. NMI of different densities in synthetic networks.

the fraction of edges that fall within given groups minus the expected fraction that would fall within the groups if the edges were randomly distributed. The modularity metric Q proposed by Newman [26] is defined as $Q = \frac{1}{2m}\sum_{uv}[A_{uv} - \frac{k_v * k_u}{2m}]\delta(c_u, c_v)$. The value in the adjacency matrix A_{uv} is 1 if vertices u and v are connected and 0 otherwise; m indicates the number of links in the graph; k_v is the degree of node v; and $\delta(c_u, c_v)$ is 1 if u and v are in the same community and 0 otherwise. $A_{uv} - \frac{k_v * k_u}{2m}$ indicates the difference between the actual number and the expected number of edges between vertices v and w. The equation sums over all vertex pairs. Modularity metric Q is used to obtain the average difference between the actual number and the expected number of edges between two vertices in the same community. Thus, edge density is proportional to Q.

Normalized Mutual Information (NMI) [27] scores ranged from 0 to 1, where 1 indicates that two sets provide a perfect match. Local community detection algorithms can create a binary partition of the network to differentiate the detected local community from the rest of the network. We take the simplified version of NMI. [28]

F-score [29] is a measure of precision and recall, where a higher score is indicative of superior performance. An F-score of 1 indicates that the detected community is precisely the same as the real community.

Cover-rate is a score determined by whether the target node of a community belongs to the final community. In the following, C_f is the number of target nodes that belong to the final community and C is the number of target nodes.



(a) Modularity. (b) NMI.

Fig. 9. The performance of different algorithms on real-world networks.

4.3 Performance Evaluation

4.3.1 Performance comparisons of CLOSE on synthetic networks

The efficiency of the CLOSE algorithm was compared with that of a local community detection algorithm based on local modularity M(s) (denoted as LCD M(s)) [14], and an algorithm based on local modularity L (denoted as LCD L) [12]. We also conducted comparisons using methods based on greedy optimization (denoted as Greedy) [30] and on expanding communities from minimal clusters (denoted as MC) [9], and a heuristic method (denoted as HLCD) [17]. Each of the algorithms was assessed in terms of accuracy (NMI) and modularity using various types of network with various numbers of nodes. The results of Greedy and HCDL for 50,000 nodes are excluded from the figures due to the fact that those algorithms are unable to deal with graphs comprising so many nodes. The HCDL results are strongly affected by the values for α and β. We have considerable experience with this algorithm [17], and found that $\alpha = 0.8$ and $\beta = 0.8$ generally provide the best performance.

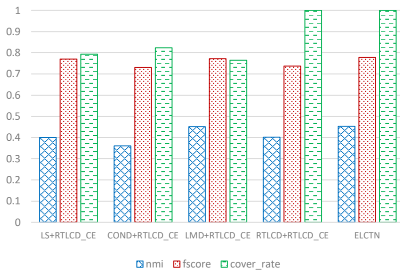
Fig. 5 compares the methods in terms of modularity in sparse, medium, and dense cases. Each subfigure in Fig. 5 presents the results obtained using a different number of nodes. Irrespective of the type of graph, CLOSE outperformed the other methods in terms of modularity and accuracy. As shown in Fig. 6, CLOSE achieved the highest accuracy regardless of the numbers of nodes. Greedy also achieved high NMI values; however, the execution time was several times longer. Figs. 7 and 8 present the variations in modularity as a function of the number of nodes. It appears that the performance of CLOSE is unaffected by the number of nodes, regardless of the type of graph.

4.3.2 Performance comparison of CLOSE using real-world data

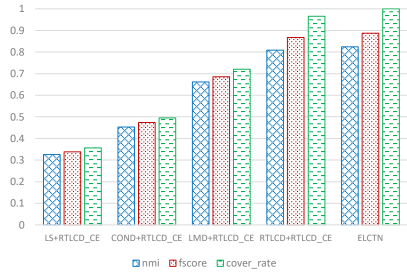
The Football, Polbooks, and DBLP datasets were selected to test the performance of the algorithms (see Fig. 9). CLOSE outperformed the other algorithms in terms of modularity and NMI on all three datasets.

4.4 Performance Comparison of ELCTN

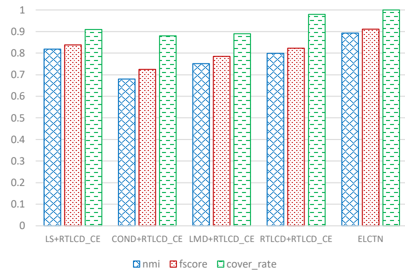
The performance of the ELCTN algorithm was evaluated using four real-world datasets (Karate Club, Football, Amazon, and DBLP) with four seed selection strategies: LS [11], COND [16], LMD [15], RTLCD [5].



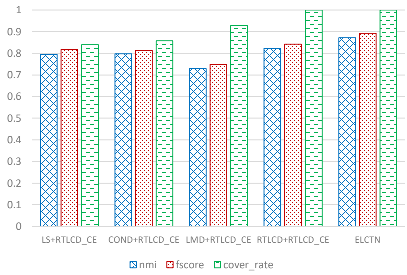
(a) Karate Club



(b) Football



(c) Amazon



(d) DBLP

Fig. 10. NMI, F-score and Cover-rate of source selection algorithms.

Fig. 10 presents the F-score, NMI, and cover-rate of the seed selection algorithms when applied to real-world networks using the same community extension methods. As shown in Fig. 10, ELCTN can have a profound influence on performance when applied to real-world networks. It appears that failure to consider the relationship between the target node and seed node increases the probability of a seed node being transferred to another community. This means that LS, COND, and LMD are ill-suited to this type of problem in real-world networks. Overall, ELCTN outperformed existing state-of-the-art algorithms in detecting communities that contain the target node.

5. CONCLUSIONS

This paper presents an algorithm for the detection of communities in complex networks based on local expansion. The proposed CLOSE algorithm performs source-node selection using a novel connective function to identify the node with the largest number of connections to its neighbors. CLOSE also selects a group of nodes rather than a single node as the seed for the expansion of a local community. Our use of neighboring groups helps to identify suitable communities for hub nodes. We developed a label propagation scheme to estimate the degree of connectivity with neighboring groups and whether they should be included in the temporary community. We also discuss the scenario in which we need determine communities for given target nodes, rather than all of the nodes in the network. The proposed ELCTN algorithm ensures that the selected source nodes are strongly related to the target node, and that they provide suitable starting points from which to form communities. In experiments, CLOSE outperformed state-of-the-art local community detection algorithms in nearly every case. ELCTN also proved highly effective in detecting communities for specific target nodes from selected source nodes.

REFERENCES

1. A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, Vol. 70, 2004, No. 0661111.
2. S. Fortunato, "Community detection in graphs," *Physics Reports*, Vol. 486, 2010, pp. 75-174.
3. V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2008, 2008, pp. P10008.
4. S. Li, H. Lou, W. Jiang, and J. Tang, "Detecting community structure via synchronous label propagation," *Neurocomputing*, Vol. 151, 2015, pp. 1063-1075.
5. X. Ding, J. Zhang, and J. Yang, "A robust two-stage algorithm for local community detection," *Knowledge-based Systems*, Vol. 152, 2018, pp. 188-199.
6. J. P. Bagrow and E. M. Bollt, "Local method for detecting communities," *Physical Review E*, Vol. 72, 2005, No. 046108.
7. F. Luo, J. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelligence and Agent Systems*, Vol. 6, 2008, pp. 387-400.
8. Y. Wu, H. Huang, Z. Hao, and F. Chen, "Local community detection using link similarity." *Journal of Computer Science and Technology*, Vol. 27, 2012, pp. 1261-1268.
9. Y. Zhou, G. Sun, Y. Xing, R. Zhou, and Z. Wang, "Local community detection algorithm based on minimal cluster," *Applied Computational Intelligence and Soft Computing*, Vol. 46, 2017, pp. 62-78.
10. C. Shang, S. Feng, Z. Zhao, and J. Fan, "Efficiently detecting overlapping communities using seeding and semi-supervised learning." *International Journal of Machine Learning and Cybernetics*, Vol. 8, 2017, pp. 455-468.
11. F. Moradi, T. Olovsson, and P. Tsigas, "A local seed selection algorithm for overlapping community detection," in *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, 2014, pp. 1-8.
12. J. Chen, O. Zaïane, and R. Goebel, "Local community identification in social networks," in *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, 2009, pp. 237-242.
13. A. Clauset, "Finding local community structure in networks," *Physical Review E*, Vol. 72, 2005, No. 026132.
14. L. L. Tao, S. Bing, Y. Y. Xiang, and T. Fang, "A local complex network communities' discovery algorithm," in *Proceedings of International Conference on Software Engineering and Service Science*, 2013, pp. 787-790.
15. Q. Chen, T.-T. Wu, and M. Fang, "Detecting local community structures in complex networks based on local degree central nodes," *Physica A: Statistical Mechanics and its Applications*, Vol. 392, 2013, pp. 529-537.

16. D. F. Gleich and C. Seshadhri, "Vertex neighborhoods, low conductance cuts and good seeds for local community methods," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 597-605.
17. M. A. Tabarzad and A. Hamzeh, "A heuristic local community detection method (hlcd)," *Applied Intelligence*, Vol. 2016, 2016, pp. 1-11.
18. C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 104-112.
19. Z.-J. Jian, H.-S. Ma, and J.-W. Huang, "Close: Local community detection by local structure expansion in a complex network." in *Proceedings of the 25th International Conference on Technologies and Applications of Artificial Intelligence*, 2019, pp. 1-6.
20. U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large scale networks," *Physical Review E*, Vol. 76, 2007, No. 036106.
21. A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, Vol. 80, 2009, No. 016118.
22. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, Vol. 33, 1977, pp. 452-473.
23. M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proceedings of National Academy of Sciences*, 2002, pp. 7821-7826.
24. J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, Vol. 42, 2015, pp. 181-213.
25. J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of International Conference on Data Mining*, 2012, pp. 745-754.
26. M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, 2004.
27. L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics*, Vol. 9, 2005, p. 8.
28. J. P. Bagrow, "Evaluating local community methods in networks," *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2008, 2008, p. P05001.
29. W. P. Li Jianhua, Wang Xiaofeng, "Review on community detection methods based on local optimization," *Bulletin of Chinese Academy of Sciences*, 2015, pp. 238-247.
30. R. Kanawati, "Community detection in social networks: the power of ensemble methods," in *Proceedings of International Conference on Data Science and Advanced Analytics*, 2014, pp. 46-52.

Hao-Shang Ma is a PhD. candidate in Institute of Computer and Communication Engineering at National Cheng Kung University, Taiwan. His research interests include data mining, social network analysis, recommendation system and deep neural network.

Shiou-Chi Li received the Master degree in Institute of Computer and Communication Engineering at National Cheng Kung University, Taiwan. He is currently a PhD. student in Institute of Computer and Communication Engineering from National Cheng Kung University. His research interests include data mining, social network analysis and graph theory.

Zhi-Jia Jian received the Master degree in Institute of Computer and Communication Engineering at National Cheng Kung University, Taiwan. Her research interests include data mining and social network analysis.

You-Hua Kuo received the Master degree in Institute of Computer and Communication Engineering at National Cheng Kung University, Taiwan. His research interests include data mining and social network analysis.

Jen-Wei Huang is an Associate Professor of Department of Electrical Engineer at National Cheng Kung University, Taiwan. He received the Ph.D. degree in Department of Electrical Engineering at National Taiwan University, Taiwan. His works focus on spatio-temporal data mining, social network analysis, multimedia information retrieval, fintech, and bioinformatics.