IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 18, 2020, accepted December 20, 2020, date of publication December 30, 2020,
date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048187

RefineDCN: An Improved Community Detection Algorithm Based on Center Finding

YING TANG, BIN WANG, AND PING WANG
Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310000, China

Corresponding author: Ying Tang (tangying@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972355, and in part by the Basic Public Welfare Research Project of Zhejiang Province, China, under Grant LGG19F020012.

ABSTRACT Detecting community structure is an important problem in complex networks. Recently, the community detection method based on centers and neighbors (DCN) has been proposed, which is divided into two stages: community center point detection and label propagation. It has a better result on community detection of simple undirected graph than other algorithms. However, when there are two community centers connected directly, the method of DCN fails to find both centers. Based on the DCN algorithm, this paper proposes an optimization method based on visually-aided user interactions. By showing the local structure of the discovered center point through the force-directed layout, the potential community centers that are missing in DCN can be detected. We further propose the hierarchical visual clustering to assist users to detect more community centers easier. In addition, to make the propagation of labels more stable, we propose the multi-label propagation strategy based on importance which also preserves the labels proportion during propagation. The experimental results on both artificial and real-world networks demonstrate that our improved algorithm RefineDCN obtains better community detection results than the DCN algorithm.

INDEX TERMS Community detection, density peak, visualization assistance, multi-label propagation.

I. INTRODUCTION

The graph structure is an effective way to represent complex network in reality, and community structure [1] is the main feature of the network. A community [2] is a set of closely-related nodes in the network. The nodes between different communities are only sparsely connected. The process of finding a community from the graph is called community detection, namely graph clustering. The purpose of community detection is to assign nodes to different communities. The correct division of node communities in the network can be used for recommendation, such as recommending friends or predicting what kind of movies a new user might like based on the tastes of members in the same community.

A popular way of community detection is to detect the community centers first, and then assign the remaining points to different communities, so as to achieve community detection [3], [4]. The algorithm of FCC [3] uses the DPC algorithm [5] to detect the community centers for network data and propagate labels to the remaining points according to the label coverage rate of neighbor points. The method of DCN [4] improves FCC with more reasonable definition of node degree and uses the label coverage rate to propagate labels to unlabeled nodes. These two methods [3], [4] can detect communities effectively for many graph structures. However, there are some cases that these two methods have difficulties in dealing with. We find two specific situations in which the above two methods are unable to detect community centers correctly or propagate the labels to nodes properly. In the first situation, we find that two directly-connected nodes of large densities can not be detected as center points since they have low distances values according to DPC. So the above algorithms based on DPC would fail to detect such directly-connected centers. However, for some real graph data such nodes need to be identified as centers. For the second situation about label propagation, the propagation order is based on the label coverage rate of neighbor points. The node is assigned the most frequently-appeared label of neighbor points. However, such strategy does not consider the label distribution of neighbor points. It is possible that there is no dominant label in the distribution of neighbor point labels and multiple labels are equally the most frequent. Under such

The associate editor coordinating the review of this manuscript and approving it for publication was Resul Das.

circumstance, the label is randomly selected to be assigned to the node. This makes the propagation process unstable and the result inaccurate.

In this paper, we propose **refineDCN**, which improves DCN in terms of both community center points detection and label propagation. For the directly-connected community center points which can not be detected in DCN, we propose an interaction-based approach to enable users find the missed center points. Specifically, we propose to visually layout the local graph structure of the nodes and enable users to interactively specify the community centers based on visually-encoded graph features. For the second problem of unstable label propagation in DCN, we propose the importance-based multi-label propagation strategy to improve the stability and accuracy of label propagation.

In summary, the main contributions of this paper include:

1) **More accurate community center points detected through user interaction:**The visually-encoded interactive interface helps users identify the community center points that can not be detected by the original DCN algorithm.
2) **More stable network clustering results by importance-based multi-label propagation algorithm:**The multi-label propagation algorithm based on node importance improves both the stability and the accuracy of label propagation to achieve more stable clustering results.
3) **Comprehensive experiments performed to demonstrate the effectiveness of our method:** We perform comprehensive experiments on artificial benchmarks and real-world network data to demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. In Section 2, we introduce the related work about graph community detection methods and label propagation algorithms. In Section 3, we review DCN and discuss its problems. In Section 4, we describe our method to optimize the DCN algorithm in detail. We perform experiments on artificial and real-world network data and compare the results in Section 5. In Section 6, we combine hierarchical ideas with our algorithm. And the summary and discussion are given in Section 6.

II. RELATED WORK

In this section, we introduce the work related to our method, including graph community detection and label propagation.

A. GRAPH COMMUNITY DETECTION

There are many clustering algorithms for graph community detection. The work of KL algorithm [6], FM algorithm [7], and spectral partitioning method [8] directly divide the graph into several parts based on the graph partitioning method. GN algorithm [1] obtains graph partition by continuously removing the relatively important edges in the graph and the concept of modularity is introduced in the GN algorithm. In the work [2], [9] after GN algorithm, modularity is also used as a criterion to evaluate the quality of graph partitioning. Besides, general clustering algorithms can also be used for community detection, such as distance-based k-means clustering [10], density-based DBSCAN [11], and so on.

Recently, the graph clustering approaches have become a hot research point which firstly identify the cluster number and cluster centers and then assign other nodes to clusters [3], [4]. Li *et al.* [12] have reported a new community detection method to estimate the optimal cluster number. A novel algorithm was proposed by Li *et al.* [13] for detecting the leaders in dynamical network based on game theory. Rodriguez *et al.* proposed the density-based clustering algorithm DPC [5] to identify the cluster centers, which is based on two assumptions: (1) the local density of the cluster center is greater than the local density of its neighbors; (2) the distance between the centers of different cluster is relatively larger. The DPC regards the points satisfying these two conditions as the cluster center points and then assigns the remaining points according to the idea similar to k-means clustering.

FCC [3] and DCN [4] use the DPC method for graph clustering. FCC directly regards node degree as the node density or considers the tightness of the relationship between the node neighbors. DCN regards the sum of the node degree and all its neighbors' degrees as the node density. Both FCC and DCN use the minimum graph distance as the distance between the nodes. Then label propagation is performed from the identified community center points by a multi-strategy label propagation algorithm. However, if there is a direct connection between two high-density nodes belonging to different communities, they would not be identified as community center points (The reason is explained in Section III). The missing of proper community centers would greatly affect the clustering effect since the subsequent label propagation algorithm is based on the identified community center points. In this paper, we propose a visually-aided method to show the local structure of detected points for user to identify potential community center points which are undetected in DCN.

B. LABEL PROPAGATION

Raghavan *et al.* use the Label Propagation Algorithm (LPA) [14] for community detection. Each network node is assigned a unique label. After the algorithm is finished, nodes of the same label are assigned to the same community. According to the label updating rule, the most frequent label is selected from the node's neighbors' to be assigned to this node. However, when there are multiple labels which are of the same most frequent uses in its neighbor nodes, only one is selected randomly according to LPA. This leads to the problem of unstable label propagation and low accuracy of community detection.

Recently-proposed methods tend to use the idea of multi-label, that is, not to immediately reject other labels with weaker attribution, but to preserve the label distribution of each node during label propagation. LPANNI [15] considers the similarity between the current node and its neighbors

when updating the label, and the node inherits the label of its neighbor according to the similarity. However, LPANNI does not detect the community center points and it is unknown how many communities there should be. LPANNI has to update all nodes in each iteration, which increase the time complexity.

By detecting the community center points and then allocating the remaining nodes, many unnecessary label updating operations can be saved. CLBLPA [16] calculates the influence value of each node according to the weighted Leader Rank algorithm [17], and then randomly selects k local maximum influence nodes as the community center points. The label update order of CLBLPA is chosen randomly without considering the impact of different update sequences, which results in unstable results.

In this paper, we propose a multi-label propagation algorithm, which considers the density value and distance value of each node calculated in DPC as the importance of the node. We perform label propagation according to the importance order of the nodes, which maintains the stability of the label propagation result.

III. DCN ALGORITHM AND ITS PROBLEMS

In this section, we describe DCN algorithm in detail and analyze its inherent problems.

A. DCN ALGORITHM

The DCN algorithm contains two parts, the first part is to identify the community center points, and the second is multi-strategy label propagation.

1) IDENTIFY COMMUNITY CENTER POINTS

DCN adopts DPC algorithm to select the community center points, in which the densities and the distances of the nodes are calculated to determine cluster centers. The density of a node in DCN is defined as the sum of the degree of itself and its neighbors' degrees.

The density of node i is calculated as:

$$\rho_i = \sum_j \eta_j + \eta_i \tag{1}$$

where η_j is the degree of the neighboring node j, and η_i is the degree of node i itself.

The distance of node i is defined as the minimum of the shortest path distance between node i and all other nodes of higher densities in Equation (2). The path distance is calculated according to Eta-reach-distance(ERD)

$$\delta_i = \begin{cases} 1\ \min(d_{ij}) = 1 \\ 2\ \min(d_{ij}) = 2 \\ 3\ \min(d_{ij}) \geq 3 \end{cases} \tag{2}$$

d_{ij} is the shortest path length between node i and node j and node j represents any node of the higher density than that of node i.

Based on ρ_i and δ_i, γ is calculated to represent the normalized density-distance value, which is defined as:

$$\gamma_i = \rho_i^* \times \delta_i^* \tag{3}$$

ρ_i^* and δ_i^* are the normalized values of ρ_i and δ_i after standard deviation normalization.

Chebyshev inequality [18] is adopted to set an upper bound for selecting the nodes with abnormally large γ as community center points, and unique labels are assigned to the center points for label propagation in the next step.

2) MULTI-STRATEGY LABEL PROPAGATION

First the seed region is formed by traversing all nodes in the network and assigning the same label to the nodes directly connected to the center points. Then we compute the labeled rate for all nodes as:

$$\psi_i = \frac{n_i}{N_i} \tag{4}$$

where n_i is the number of labeled neighbor nodes of node i and is N_i the number of all neighbor nodes of node i.

The node of highest labeled rate is chosen to be assigned the label which is of the highest occurrences in the neighbor nodes. Then the labeled rate of the neighbor nodes are recalculate.

B. PROBLEMS

There are two main problems in DCN.

First, the density-distance value γ involves both the density value and the distance value. When there are two community center points (the groundtruth center points that have not been all identified) directly connected in the network, the distance value of the lower-density node is set to 1 according to the definition of the distance. Therefore, it is highly possible that the lower-density node will be not be detected due to its low distance value. This leads to the missing of one community center point and affects the subsequent multi-strategy label propagation result. For example, in Figure 4 we show the network containing two directly-connected nodes which are the dark blue node and the node of yellow circle. Although these two nodes are directly-connected, we can see clearly there are two clusters corresponding to them and they should both be detected as the center nodes. However, only the dark blue node is identified as the center point by DCN and the yellow circle one is not due to its distance value being 1. So we need to address this situation where both directly-connected nodes are center points. We find the visual layout result of network structure provides helps for users to select the center points, so we propose the visual exploration for new center points in the next section.

Second, in the part of multi-strategy label propagation, the DCN calculates the labeled rate of each unlabeled node, and choose the node of the highest labeled rate to update its label. The larger the labeled rate is, the richer neighbor information the node contains. The node is updated to be assigned the label of the most occurrences in its neighbors. However, the most frequently-occurred label in the neighbors of the

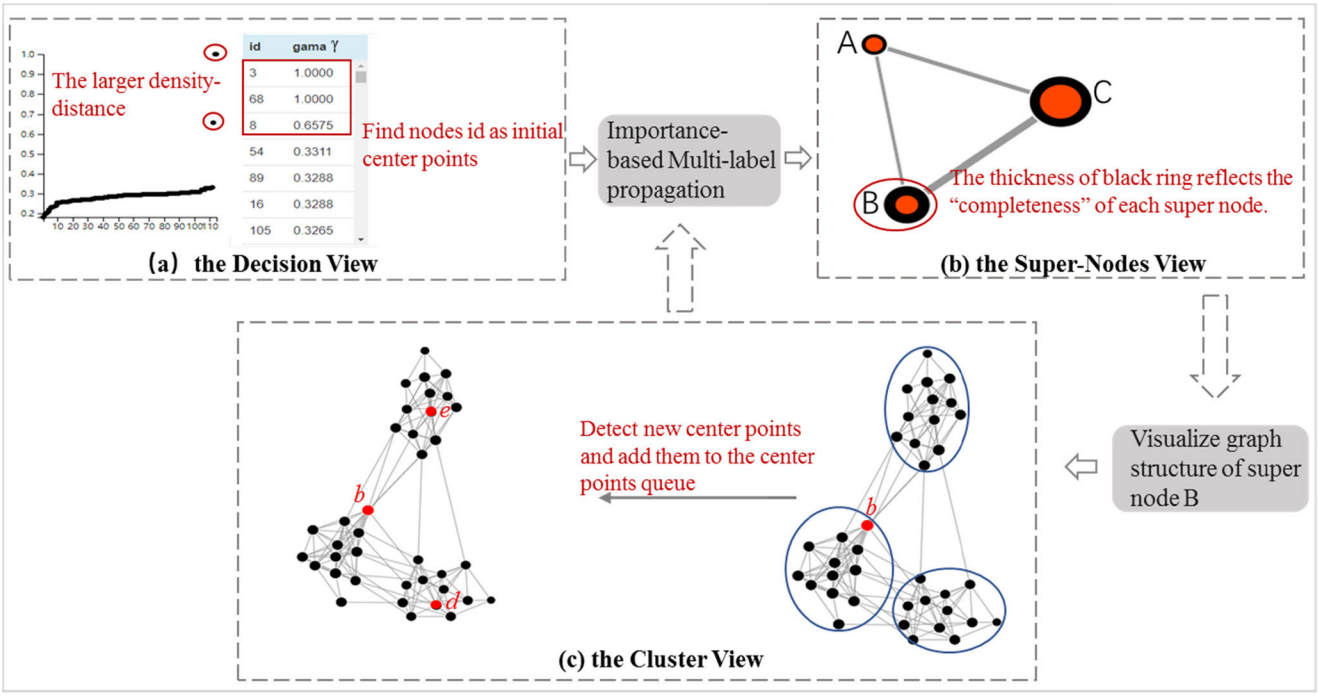

FIGURE 1. The Hierarchical Visual Clustering model has three views to assist users to participate in iterative clustering. (a) The Decision View contains decision graph and density-distance list, which help users identify the initial center points; (b) The Super-Nodes View shows the clusters represented as nodes obtained by importance-based multi-label propagation. According to the ratio of the inner and outer radius of the super nodes, the users find the cluster that need to be further improved by detecting more center nodes within it; (c) The Cluster View shows the node-link structure of the cluster selected by users to be improved. After observing the local structures of the cluster, nodes of large densities in the sub-cluster are added as the new center points.

largest-labelled-rate node is not necessarily the label that occupies the dominant position of the neighbors of all nodes. For example, if the node of largest labelled rate has many different labels of close proportions for its neighbor nodes, the most-occurred label may not be dominant compared with other nodes' labelled neighbors. Besides, in some cases there will be multiple labels of the same highest occurrences in the neighbor nodes. Under current multi-strategy label propagation algorithm of DCN, one of these multiple labels is randomly selected, which affects the stability and accuracy of the label propagation result. So specifying a certain label for a node during label propagation is not good in some situations as we analyzed above. It is better to keep the label algorithm.

IV. THE PROPOSED ALGORITHM

We improve the above two problems in DCN. For missing community center points, we visualize the local structure of the identified community center point for users to interactively search for potential center points of high density. For the problem in the multi-strategy label propagation, we propose a propagation algorithm based on node importance. The above two parts are integrated to form our hierarchical visual clustering model for network data. In the following of this section, we first give the overview of our hierarchical visual clustering, then the details of each part are described in the following sub-sections.

A. OVERVIEW OF THE HIERARCHICAL VISUAL CLUSTERING

The flowchart of the proposed Hierarchical Visual Clustering model is shown in Figure 1. We design three views to assist users to participate in clustering, which are **the Decision View(a), the Super-Nodes View(b) and the Cluster View(c).** The users interact with these three views to get the final clustering results.

The Decision View contains a decision graph and density-distance (γ) list as shown in Figure 1(a). From the Decision View, users can identify initial center points for clustering. We explain the details of the Decision View in the sub-section ***B.***

After we get the initial center points, the importance-based multi-label propagation is invoked to obtain the clustering result as shown in the **Super-Nodes View** in Figure 1(b). The multi-label propagation for class labeling is described in the sub-section ***C.***

The three nodes in Figure 1(b) represent the three clusters after propagation, called **Super-Nodes**. We visually encode each super node as a black ring containing a red circle. The size of each super-node is proportional to the number of nodes in the cluster. The thickness of the black ring is inversely proportional to the Aggregation Coefficient (AC), which is defined as:

$$\mathrm{AC} = \frac{2E}{K(K-1)} \tag{5}$$

where E is the number of edges in the cluster, K is the number of nodes in the cluster. The closer AC is to 1 (i.e. the thinner the black ring is), the closer the cluster is to the complete subgraph. For example, for the super-node A in Fig 1(b), the thin black ring indicates that the cluster structure of super-node A is close to a complete graph shown in Figure 2.

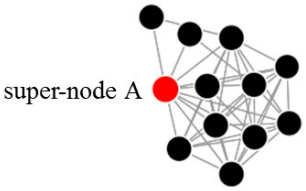

FIGURE 2. The cluster structure of super-node A. All nodes in the cluster are closely related, which is close to a complete graph.

By observing the thickness of the black ring in the Super-Nodes View, we detect clusters less tightly connected and that are need to be further divided, such as the super-node B in Figure 1(b). To further decompose super-node B, we design **the Cluster View** (Figure 1(c)), which shows the node-link graphs of the super-node B where users find new center points. The details of new center point detection are shown in sub-section ***D***.

After new center points detected, the label propagation is re-used to get the new Super-Nodes View. Users again find the super-node to be improved in the Super-Nodes View and detect the new potential center points in the Cluster View. The above process is iteratively executed to achieve a hierarchical network clustering, in which users are actively involved by interacting with the three views to get the final clustering results.

Algorithm 1 Hierarchical Visual Clustering

Require: Center_points C
For node in density-distance list:
  If node's $\gamma >$ *Threshold* ε
    $C+ =$ *node*
  End if
End for

Super-nodes View $\leftarrow$ *Multi-label propagation (C)*

For each Super node in Super-nodes View:
  s_node $\leftarrow$ *Super node*
  If s_node's black circular ring $>$ *threshold* θ
    $C+ =$ *Detect new center points in the Cluster View of the s_node*
  End if
  Update Super-nodes View $\leftarrow$ *Multi-label propagation (C)*
End for

Ensure: each Super node's black circular ring $> \delta$

The whole process of hierarchical visual clustering in shown in **Algorithm 1**. The interactive algorithm stops when users find there is no need to divide the super-node. Alternatively, we can define a threshold value θ to make our algorithm stop when the thicknesses of all super-nodes' black ring meet the threshold. To do this, we compute the value of the thickness of the black ring as 1 minus the ratio of radii of red circle to that of the black circle, i.e. 1 minus AC coefficient. When the value is below θ, our algorithm indicates to users that they can stop finding new center points. The value of θ is set to be 0.1. The details of **Algorithm 1** including functions and parameters are explained in the following sub-sections.

To explain our method conveniently without losing generality, we describe our method on the synthetic network LFR2. LFR2 is an artificial data set generated by the Lancichinetti-Fortunato-Radicch (LFR) benchmark network [19]. LFR2 contains 3 communities, 1200 nodes and 3576 edges. The relevant parameters of LFR2 are shown in Table 1.

TABLE 1. Data sets used in the experiments.

Artificial data sets							
Networks	n	k_m	μ	t_1	t_2	c_{min}	c_{max}
LFR1	200	3	0.1	2	1	50	50
LFR2	1200	5	0.2	2	1	400	400
Real-world data sets							
Networks	n	k_m	description				
Karate [1]	34	4.6	A social network of friendships between 34 members of a karate club at a US university in the 1970s.				
Dolphins [21]	62	5.1	An undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound				
Polbooks [22]	105	8.4	A directed network for hyperlinks between weblogs on US politics, recorded in 2005 by Adamic and Glance.				
Football [1]	115	10.7	A network of American football games between Division IA colleges during regular season Fall 2000.				

B. IDENTIFYING COMMUNITY CENTER POINTS

For LFR2, two community center points can be identified by the original DCN algorithm, and the decision graph is shown in Figure 3. The decision graph is computed by DPC algorithm to help users find the center points with obviously higher density-distance value. Here, the x-axis shows the node index, and the y-axis represents the density-distance value γ. The two scatter points above the red solid line which is computed by the Chebyshev inequality are two points that are of significantly larger density-distance values than other points. In our algorithm, the parameter ε of the Chebyshev inequality is set to be 2. The corresponding nodes in the network (node 1198 and 1199) are the identified community center points, each one is assigned a unique label.

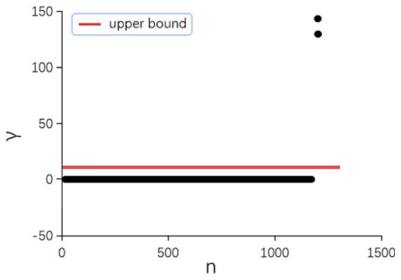

FIGURE 3. Decision graph for LFR2.

Sometimes multi-points of close density-distance values would be so close in the view that it becomes difficult for users to discern all these points. So in the Decision View the decision graph combined with the density distance list provide the interfaces for users to choose the initial center points. For example, although it is difficult to find three center points in the decision graph in Figure 1(a), combined with the density-distance list, we know that there are actually three nodes that can be used as the initial center points.

C. CLASS LABELING BY MULTI-LABEL PROPAGATION BASED ON IMPORTANCE

In order to avoid randomly selecting the label of the node needs from multiple most frequent labels of the neighbor nodes, we use the multi-label idea to preserve the influence proportions of multiple labels of its neighbors. Different from the original propagation algorithm, we consider the proportions of multiple labels during the propagation process. In our algorithm, we sort the nodes according to their importance which is defined by the density-distance value γ of the node. We propagate the labels according to the order of the sorted nodes from the highest importance to the lowest.

Firstly, we set an initial multi-label l for each node in the network. The initial multi-label of node i is defined as a c-dimensional vector:

$$l_i = \langle 0, 0, ..., 0 \rangle \tag{6}$$

where c is the number of all center points. Every center point has a unique serial number from 1 to c. The i th center point's multi-label is: $\langle 0, \ldots 1, \ldots, 0 \rangle$, where the ith item of the vector is set to 1 and all the other items are 0. The nodes directly connected to only one of the center points are assigned the same label with the center point to form the seed region.

Then for each iteration, the unlabeled node with the highest importance is chosen to assign the calculated label. The new label of node i is calculated as:

$$l_i = \sum_{m=1}^{n} J_{i,m} \times l_m \tag{7}$$

where n is the number of labeled neighbor nodes of node i, and the $J_{i,m}$is the Jaccard similarity [20] between node i and the neighbor node m. The similarity is calculated as:

$$J_{i,m} = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

A and B are the sets of neighbor nodes of two points. The more have both sets the same neighbor nodes, the greater the similarity is.

From Equation (7) and (8), we can see that more similar nodes contribute more to the computation of the label of current node.

The multi-label l_i is normalized to get the normalized multi-label l_i^* which is used as the final multi-label of node i. The normalization is computed as:

$$l_i^* = \frac{l_i}{\sum_{k=1}^{c} l_{i,k}} \tag{9}$$

where $l_{i,k}$ is the kth item of normalized multi-label of node i. The normalized multi-label l_i^* is the final multi-label of node i.

The propagation stops when the number of unlabeled node is zero. For each node, the label of the maximum proportion is chosen as the final label. The algorithm of multi-label propagation based on importance is shown in **Algorithm 2.**

Algorithm 2 Multi-Label Propagation

Require: Network $G = (V, E)$, $|V| = n$, $|E| = m$, *community centers* C, $|C| = v$.

For $i = 1{:}v$ *do*
  If the ith center's neighbors only connect to one center then Propagate the center's label to them
  End if
End for

Sorting the remained unlabeled nodes in a descending order by their importance

While the number of unlabeled node is larger than zero Do

1. *Select the node with highest importance*
2. *Assign the label according to its neighbors' labels and the structural similar between them*
3. *Normalize the multi-label of the node*

End while

For $i = 1{:}$ n *do*
  Select the label of the maximum proportion as the final label of the ith node
End for

Ensure: the partition result

D. VISUAL EXPLORATION FOR NEW CENTER POINTS

We adopt the force-directed graph layout method [19] to show the network structure. The force-directed graph layout is a widely-used method to draw graphs in an aesthetically-pleasing way, which provides the overview of the network structure. The users can identify obvious cluster structure from force-directed layout.

From the Cluster View in Figure 1(c), there are three distinct sub-cluster structures after force-directed graph layout, and the center point (the red point b) is located in one of

these sub-clusters. From this cluster view of super-node B, users find new center points and add them to the center points queue by clicking it. In Figure 1(c), users select two red points d and e as new center nodes which are added into the center point queue. The updated center point queue is passed to the procedure of label propagation in the Super-Nodes view for new pass of label propagation to get the updated clustering result. Next, we explain the visualization details of graph-layout of the Cluster View.

First, in order to make users easily find the node of the largest density, we render the nodes in the cluster view in a way that their sizes are proportional to their density values (Figure 4). The larger the nodes are, the more likely they are the center nodes. When the sizes of nodes are too close to tell the difference, there is a list above the Cluster View which shows the concrete density values. For the sake of showing structures clearly, the nodes are of the same size in the Cluster View of Figure 1(C). In real operations, the nodes are of sizes proportional to densities to enable users find proper center pints.

FIGURE 4. The 2nd-order local subgraph of node 1199 in LFR2.

Secondly, when there are too many nodes or edges in the Cluster View, it would be difficult for users to detect potential center points due to visual clutter. In this situation, it is not a good choice to display the whole graph of the super-node for users to explore. Here, we define the 2nd-order local subgraph of a center point. The 2nd-order subgraph contains the set of nodes that are less than or equal to 2 in terms of the shortest path distance to the current center point. The 2nd-order subgraph is visualized as the local structural diagram of a center node. The 2nd-order local subgraph of node 1199 in LFR2 is shown in Figure 4, where the two blue nodes are the two identified community center points (node 1198 and 1199), and the remaining gray nodes represent the nodes that has not been assigned a label.

The 2nd-order local subgraph show the local structure of one detected point for user to determine if there is any cluster with no detected center points, which solves the problem of limited display and reduces the computational complexity of graph layout.

We can clearly see that there are three clusters in Figure 4, and only two clusters have the labeled center nodes (in blue). For the cluster in the lower right part of the figure, there is no identified center point in this cluster since all nodes are gray. So we interactively mark the node with the largest size as the community center point, which is shown by the yellow circle. This node is directly connected to the dark blue node, so its distance value is set to 1. Such distance value results to a relatively small density-distance value γ that cannot be recognized in the DCN algorithm. However, in our method, users can find this center point by visually exploring the local graph structures.

So far, the three center points (Figure 4) are added to the center points queue. Next, the same visual exploration is performed on the remaining center points of super node B in the queue. Other center points found in the exploration process are also added to the center points queue. We keep performing such visually-aided exploration process until all the center points of super node B in the queue have been explored.

E. THE TIME AND SPACE COMPLEXITY OF OUR METHOD

Assume the network has n nodes and m edges, the time complexities to compute all nodes' densities and distances are $O(n)$ and the $O(m)$ respectively. Assume the average degree of each node is k, the time complexity of multi-label propagation is $O(kn)$. So the whole time complexity for the first pass is $O(kn + m)$, which is the same with DCN.

As for each iteration in the following pass, only multi-label propagation is needed to be invoked to get the new clustering results, so the time complexity of each later iteration is $O(kn)$. Compared with DCN, our model has more computing time in terms of hierarchical refine passes to find more new center points. This additional computing time is dependent on specific network. For some network we can find all cluster with few passes, yet for others multiple passes are needed to find all clusters. However, the more additional time our method costs, the more new clusters are found which is beneficial to more accurate clustering result.

Next, we analyze the space complexity of our algorithm. The space complexity of computing all nodes' densities and nodes' distances are both $O(n)$. Assume the number of communities is c, the space complexity of label propogating is $O(cn)$. So the whole space complexity of our algorithm is $O(cn)$, which is linearly related to the number of nodes. Our algorithm's space complexity is nearly the same with DCN's space complexity $O(n)$. The procedures to compute nodes' densities and distances are the same for both methods. As for label propagation, our algorithm saves the neighbor labels' distribution which makes it $O(cn)$ compared with $O(n)$ of DCN. Since the number of communities c is far less than n, we consider the space complexities of both methods almost the same.

V. EXPERIMENTS AND RESULTS

In this section, we perform experiments to verify the effectiveness of our improved methods.

A. DATA SETS

The experiment was conducted on six data sets, including two artificial data sets and four real-world data sets. The data sets are shown in Table 1.

LFR1 and LFR2 are generated by the Lancichinetti-Fortunato-Radicch (LFR) benchmark network [19] and the parameters of LFR1 and LFR 2 are shown in Artificial data sets in Table 1. Karate, Dolphins, Polbooks and Football are real-world data sets. Here n denotes the numbers of vertices. k_m is the averaged node degree, μ is the mixing parameters, t_1 is the negative exponent for the degree distribution, t_2 is the negative exponent for the community size distribution, c_{min} and c_{max} is the minimum and maximum size of communities respectively.

B. EVALUATION INDICATIONS AND COMPARISON ALGORITHMS

Four evaluation indicators are used as indicators for evaluating community detection, which are Accuracy (Acc) [23], Standard Mutual Information (NMI) [24], Rand Index (ARI) [25], and Modularity (Q) [1].

Acc is calculated as [23]:

$$\text{Acc} = \frac{\sum_{i=1}^{c} a_i}{n} \tag{10}$$

where a_i is the number of correctly classified nodes which belong to the ith community, c is the number of communities, and n is the number of nodes.

Assume we have the ground-truth communities partition $\boldsymbol{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots \boldsymbol{R}_{\boldsymbol{P}}\}$, where P is the number of communities. We also have $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots \boldsymbol{S}_{\boldsymbol{T}}\}$ which is the partition result obtained from community detection algorithm and T is the number of detected communities.

NMI is defined as [24]:

$$\text{NMI} = \frac{MI(R, S)}{\sqrt{H(R)H(S)}} \tag{11}$$

where $MI(R,S)$ is the mutual information between R and S, $H(R)$ and $H(S)$ are the entropy of R and S respectively [24].

ARI is calculated as [25]:

$$ARI = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00}+N_{01})(N_{01}+N_{11}) + (N_{00}+N_{10})(N_{10}+N_{11})} \tag{12}$$

where N_{11} is the number of node pairs belonging to the same community in R and S. N_{00} is the number of node pairs that do not belong to the same community in the community division in R and S. N_{01} is the number of pairs not belonging to the same community in the community division S obtained by the algorithm but belonging to the same community in the actual community division R. N_{10} is the number of pairs belonging to the same community in S but not belonging to the same community in R.

Q is defined as [1]:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{K_i K_j}{2m}) \delta(C_i, C_j) \tag{13}$$

where m is the number of edges. A_{ij} is the element of the adjacency matrix, $K_i(K_j)$ is the degree of node i (j), and $\delta()$ is the Kronecker function, C_i (C_j) is the ith (jth) community.

And the five algorithms used to compare with our algorithm are: Louvain [26], Label Propagation Algorithm (LPA), Infomap [27], Eigenvector [28] and DCN algorithm. The time complexities of the above five compared algorithms are listed in Table 2.

TABLE 2. The time complexities of the five compared algirthms.

Algorithms	Louvain	LPA	Infomap	Eigenvector	DCN
Time Complexity	O((n+m)n)	O(n+m)	O(I^2m)	O(n^2)	O(kn+m)

In Table 2, n is the number of nodes and m is the number of edges in the network. I is the number of iterations executed in Infomap. Compared with these five algorithms our method has linear complexity with regard to the number of nodes and edges, which makes it efficient.

C. EXPERIMENTAL RESULTS

In Table 3-8 we show the experimental results of our method compared with that of the other five algorithms.

TABLE 3. Experimental results of LFR1 dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	4	-	-	-	0.676
DCN	4	0.650	0.612	0.830	0.644
refineDCN	**4**	**0.679**	**0.650**	**0.845**	**0.638**
Infomap	20	0.648	0.450	0.535	0.724
Louvain	10	0.632	0.494	0.355	0.719
LPA	199	0.415	-	0.025	0.341
eigenvector	8	0.691	0.622	0.695	0.687

TABLE 4. Experimental results of LFR2 dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	3	-	-	-	0.493
DCN	2	0.290	0.265	0.563	0.316
refineDCN	**3**	**0.383**	**0.439**	**0.774**	**0.422**
Infomap	132	0.271	0.137	0.302	0.562
Louvain	30	0.297	0.255	0.009	0.570
LPA	1199	0.268	-	0.003	0.250
eigenvector	25	0.240	0.137	0.153	0.493

Table 3-8 list the ground truth and the results of six data sets by six algorithms. Community is the number of communities detected by the algorithm. NMI, ARI, Acc and Q are the four indicators adopted in this paper.

We identify new center points for data sets LFR2 (Table 4) and Football (Table 8) with the hierarchical visual clustering method. Comparing the results of DCN and refineDCN in Table 4 and Table 8, we see the results by refineDCN are better than original DCN for all indicators.

TABLE 5. Experimental results of Karate dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	2	-	-	-	0.410
DCN	2	1.000	1.000	1.000	0.410
refineDCN	**2**	**1.000**	**1.000**	**1.000**	**0.410**
Infomap	3	0.700	0.702	0.824	0.464
Louvain	4	0.587	0.462	0.324	0.510
LPA	26	0.380	0.135	0.177	0.268
eigenvector	4	0.677	0.512	0.559	0.496

TABLE 6. Experimental results of Dolphins dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	2	-	-	-	0.391
DCN	2	0.814	0.872	0.968	0.372
refineDCN	**2**	**0.890**	**0.935**	**0.984**	**0.391**
Infomap	6	0.503	0.367	0.581	0.595
Louvain	7	0.456	0.267	0.194	0.602
LPA	56	0.285	0.019	0.129	0.289
eigenvector	6	0.500	0.281	0.258	0.582

TABLE 7. Experimental results of Polbooks dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	2	-	-	-	0.435
DCN	2	0.598	0.667	0.848	0.456
refineDCN	**2**	**0.731**	**0.670**	**0.867**	**0.457**
Infomap	6	0.493	0.536	0.695	0.560
Louvain	6	0.470	0.372	0.305	0.572
LPA	3	0.075	0.036	0.476	0.010
eigenvector	4	0.516	0.536	0.438	0.517

TABLE 8. Experimental results of Football dataset using different community detection algorithms.

Algorithms	Community	NMI	ARI	Acc	Q
Ground Truth	12	-	-	-	0.585
DCN	3	0.365	0.108	-	0.328
refineDCN	**11**	**0.847**	**0.751**	**-**	**0.598**
Infomap	12	0.924	0.897	-	0.650
Louvain	11	0.911	0.857	-	0.651
LPA	114	0.680	-	-	0.150
eigenvector	8	0.699	0.464	-	0.549

For data sets in Table 3, 5, 6, 7, no new center points are identified by our visually-aided method, which means that refineDCN only worked one pass to get the clustering results. However, all results by refineDCN are still better than that of DCN except for data set Karate in Table 5 where both methods produce the same best result. This is due to our importance-based multi-label propagation algorithm.

From the results in Table 3-8, we find that our method has improvements over original DCN. Besides DCN, our method gets better results for nearly all data sets than other five algorithms. As for the football dataset which has many centers, our method is not as good as that of Infomap. Still our result is comparable with Louvain's and much better than that of DCN.

VI. CONCLUSION

In this paper we propose an improved clustering method based on DCN. The first improvement is the visually-guided interactive approach for users to find potential center points. The local structural diagram of the identified community center point is displayed by the force-directed graph layout. Users can explore the local structural diagram to see if there exist potential community center points. For data set with many clusters, in this paper we adopt the idea of hierarchical clustering. According to visual cues associated with the Aggregation Coefficients of the cluster, users can select the cluster which is not tight enough to detect the potential cluster centers within it. The second is the improved multi-label propagation algorithm based on node importance to determine the clusters. We compare our method to DCN and other algorithms to show that our algorithm generate better results nearly for each indicator on all data set.

However, there are still some limitations for our method in this paper. The potential community is explored through the local 2nd-order structural diagram. Although the number of nodes and edges has been reduced, for some large-scale network, the number of nodes of the local structural diagram may still be so large that the local graph is visually cluttered with force-directed layout.

Since the graph layout is an independent stage of our algorithm, in our future work, we plan to replace it with other scalable dimensionality reduction layout algorithms such as MDS [29] and t-SNE [30] to handle large-scale local graphs to find potential center points.

In the complex networks research community, there are more and more work devoted to multiplex networks [31], [32] and dynamic networks [33] with the development of wireless network or social networks, in the future we will work on to extend our method to deal with multiple correlated networks or dynamic network topology by integrating more advanced local subspaces projection techniques [34] for users to detect the center points.

REFERENCES

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[2] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113, doi: 10.1103/PhysRevE.69.026113.

[3] Y. Chen, P. Zhao, P. Li, K. Zhang, and J. Zhang, "Finding communities by their centers," *Sci. Rep.*, vol. 6, no. 1, pp. 1–8, Apr. 2016, doi: 10.1038/srep24017.

[4] J. Ding, X. He, J. Yuan, Y. Chen, and B. Jiang, "Community detection by propagating the label of center," *Phys. A, Stat. Mech. Appl.*, vol. 503, pp. 675–686, Aug. 2018, doi: 10.1016/j.physa.2018.02.174.

[5] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014, doi: 10.1126/science.1242072.

[6] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, 1970, doi: 10.1002/j.1538-7305.1970.tb01770.x.

[7] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. 25th Electron. Design Automat.* New York, NY, USA: ACM, 1988, pp. 241–247, doi: 10.1145/62882.62910.

[8] E. R. Barnes, "An algorithm for partitioning the nodes of a graph," *SIAM J. Algebr. Discrete Methods*, vol. 3, no. 4, pp. 541–550, Dec. 1982, doi: 10.1137/0603056.

[9] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, Aug. 2005, Art. no. 027104, doi: 10.1103/PhysRevE.72.027104.

[10] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math, Statist., Probab.*, vol. 1. Berkeley, CA, USA: Univ. of California Press, 1967, pp. 281–297.

[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[12] H.-J. Li, L. Wang, Y. Zhang, and M. Perc, "Optimization of identifiability for efficient community detection," *New J. Phys.*, vol. 22, no. 6, Jun. 2020, Art. no. 063035.

[13] H. J. Li, Z. Bu, Z. Wang, and J. Cao, "Dynamical clustering in electronic commerce systems via optimization and leadership expansion," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5327–5334, Aug. 2020.

[14] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[15] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, "LPANNI: Overlapping community detection using label propagation in large-scale complex networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1736–1749, Sep. 2019, doi: 10.1109/TKDE.2018.2866424.

[16] S. Liu, F. Zhu, H. Liu, and Z. Du, "A core leader based label propagation algorithm for community detection," *China Commun.*, vol. 13, no. 12, pp. 97–106, Dec. 2016, doi: 10.1109/CC.2016.7897535.

[17] Q. Li, T. Zhou, L. Lü, and D. Chen, "Identifying influential spreaders by weighted LeaderRank," *Phys. A, Stat. Mech. Appl.*, vol. 404, pp. 47–55, Jun. 2014, doi: 10.1016/j.physa.2014.02.041.

[18] J. Ding, X. He, J. Yuan, and B. Jiang, "Automatic clustering based on density peak detection using generalized extreme value distribution," *Soft Comput.*, vol. 22, no. 9, pp. 2777–2796, May 2018, doi: 10.1007/s00500-017-2748-7.

[19] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw., Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, Nov. 1991, doi: 10.1002/spe.4380211102.

[20] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bull. de la Société Vaudoise des Sci. Naturelles*, vol. 37, no. 142, pp. 547–579, Jan. 1901.

[21] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, Sep. 2003, doi: 10.1007/s00265-003-0651-y.

[22] V. Krebs. *Books About US Politics*. Accessed: 2004. [Online]. Available: http://www.orgnet.com/

[23] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 1997, pp. 21–34.

[24] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, 2003, doi: 10.1162/153244303321897735.

[25] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 2837–2854, Jan. 2010, doi: 10.1007/s10846-010-9415-x.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008, doi: 10.1088/1742-5468/2008/10/p10008.

[27] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008, doi: 10.1073/pnas.0706851105.

[28] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104, doi: 10.1103/PhysRevE.74.036104.

[29] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952, doi: 10.1007/BF02288916.

[30] G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Vigiliae Christianae*, vol. 9, no. 2, pp. 2579–2605, 2008.

[31] H.-J. Li and L. Wang, "Multi-scale asynchronous belief percolation model on multiplex networks," *New J. Phys.*, vol. 21, no. 1, Jan. 2019, Art. no. 015005.

[32] H.-J. Li, Z. Wang, J. Pei, J. Cao, and Y. Shi, "Optimal estimation of low-rank factors via feature level data fusion of multiplex signal systems," *IEEE Trans. Knowl. Data Eng.*, early access, Aug. 13, 2020, doi: 10.1109/TKDE.2020.3015914.

[33] H.-J. Li, Q. Wang, S. Liu, and J. Hu, "Exploring the trust management mechanism in self-organizing complex network based on game theory," *Phys. A, Stat. Mech. Appl.*, vol. 542, Mar. 2020, Art. no. 123514.

[34] R. Bian, Y. Xue, L. Zhou, J. Zhang, B. Chen, D. Weiskopf, and Y. Wang, "Implicit multidimensional projection of local subspaces," *IEEE Trans. Vis. Comput. Graphics*, early access, Oct. 13, 2020, doi: 10.1109/TVCG.2020.3030368.

YING TANG received the Ph.D. degree from Zhejiang University, China, in 2005. She is currently an Associate Professor with the Zhejiang University of Technology, China. Her research interests include data mining and visualization.

BIN WANG received the M.D. degree from the Zhejiang University of Technology, China, in 2019. His research interests include data mining and visualization.

PING WANG is currently pursuing the master's degree with the Zhejiang University of Technology, Hangzhou, China. Her research interests include data mining and data visualization.