Community Detection Algorithm Based on Nonnegative Matrix Factorization and Improved Density Peak Clustering

HONG LU\textsuperscript{1}, XIAOSHUANG SANG\textsuperscript{1}, QINGHUA ZHAO\textsuperscript{2}, AND JIANFENG LU\textsuperscript{1}\textsuperscript{1}\textsuperscript{1}\textsuperscript{1}

\textsuperscript{1}School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China
\textsuperscript{2}College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China

Corresponding author: Jianfeng Lu (ljf@njust.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004, and in part by the 111 Project under Grant B13022.

\textbf{ABSTRACT} Community detection is a critical issue in the field of complex networks. Recently, the nonnegative matrix factorization (NMF) method has successfully uncovered the community structure in the complex networks. However, this method has a significant drawback; most of community detection methods using NMF require the number of communities to be preassigned or determined the number of communities by searching for the best community structure among all candidates. To address this problem, in this paper, we use density peak clustering (DPC) to obtain the number of centers as the pre-defined parameter for nonnegative matrix factorization. However, due to sparse and high dimensional characteristics of complex networks, DPC cannot be used to detect community directly. To overcome this issue, we employ degree and hop of nodes as the density and distance indexes, respectively; we use NMF and Symmetric NMF to deal with linearly separable data and non-linearly separable data, respectively. Experimental results show that the proposed methods exhibit excellent performance on artificial and real-world networks and superior to the state-of-the-art methods which are the most common method for community detection of complex networks.

\textbf{INDEX TERMS} Community detection, complex networks, density peak clustering, nonnegative matrix factorization.

\section{I. INTRODUCTION}
Community structure is ubiquitous in the form of networks, such as social networks\cite{1}, biological networks\cite{2}, citation networks\cite{3}, etc, which makes community detection especially crucial for better understanding the organization of networks and extracting useful information. Although the common definition has not yet been agreed, a community, also called a module or a cluster, is typically regarded as a group of nodes that are densely interconnected but loosely connected to the other communities of the network\cite{4}.

Since the seminal work by Girvan and Newman\cite{1}, a number of algorithms for community detection in complex networks have been proposed, such as modularity based algorithms\cite{5}, clustering based algorithms\cite{6,39}, random walk based algorithms\cite{7} and matrix decomposition based algorithms\cite{8,9,10,11}. Readers who are interested can refer to comprehensive surveys of community detection\cite{12,13,14}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{The community detection process.}
\end{figure}

As one of the hot research topics, nonnegative matrix factorization (NMF) has been broadly adopted in community detection, to name a few,\cite{10,15,16,17} etc. Despite a lot of methods proposed, determining the number of communities is an important and thorny issue in practice. Most of methods adopted maximum modularity (Q) as the criterion to determine the number of communities, such as SNMF-SS\cite{18} proposed by Ma et al. and SBMF\cite{17} proposed by Zhang et al., which is very time-consuming. Other methods rely on external algorithms such as Shi et al.\cite{19} or need the number of communities as priori information such as\cite{10,15}. However, determining the number of communities in advance is often unfeasible for concrete application and it is also very inefficient to decide the number of communities (K) by searching all possible candidates.

To address the above issues, we, inspired by DPC algorithm\cite{20} and chen’s method\cite{21}, propose DPCNMF (Density Peak Clustering with Nonnegative Matrix Factorization) for community detection on linearly separable data and DPCSNMF (Density Peak Clustering with Symmetric
Nonnegative Matrix Factorization) for non-linearly separable data. The main advantage of this scheme is: it identifies the number of communities (K) by a single run and can find hub points accurately. Unfortunately, in view of the sparse and high dimensional characteristics of complex networks, DPC algorithm cannot be used on these datasets directly; the similarity between nodes becomes meaningless due to the curse of dimensionality. Therefore, we adopt the degree of a node as the density index and the length of the shortest path between two nodes as the distance index in this paper. The advantage of applying the node degree as density index is that it can avoid the influence of the cut-off parameter d_c to clustering results. Then, we utilize the number of density peaks as the number of communities instead of input in advance as usual for NMF.

In summary, the key contributions of this paper are:

1) we propose parameter-free algorithm which is only based on topology structure to implement unsupervised community detection method.

2) we provide a method to derive appropriate number of communities with improved density peak clustering, and combine NMF based method to get communities structure. To the best of our knowledge, this is the first work to address NMF’s input parameter K by the density peak clustering.

3) The advantages of adopting improved density peak clustering lie in that: it enables us not only to obtain hubs of the network, but also to analyze the different scale community via choosing the number of centers. In addition, unlike DPC which selects cluster centers manually, our method can choose centers automatically.

The rest of the paper is organized as follows. Section 2 introduces the related work on NMF based algorithms to identify community and briefly reviews DPC and several improved DPC methods. Section 3 elaborates our DPCNMF and DPCSNNMF methods. Section 4 provides several experimental evaluations between our approaches and some representative algorithms on synthetic and real-world networks. Finally, Section 5 summarizes the paper.

II. RELATED WORK
A. RELATED NMF BASED COMMUNITY DETECTION ALGORITHMS

Owing to its innate interpretability and good performance in practice, NMF has received extensive attention in community detection. Many researchers are devoted to improving the performance of NMF in community detection, and some of their representative work includes:

Psorakis et al. [22] proposed an approach for finding overlapping communities using a Bayesian NMF model. This method has the advantages of automatically determining the number of communities and does not have the resolution limit. Unfortunately, its built-in estimation of the number of communities can mislead the factorization and return an incorrect solution. Wang et al. [23] investigated how to apply NMF to the community detection for directed, undirected and hybrid networks and developed the optimization algorithm by iteration. Besides, Chen et al. [16] proposed NFOSCM method that can detect network structure with unknown community number by feature matrix preprocessing and ranking optimization. Zhang and Yeung [24] proposed a constrained NMF triple decomposition model, named BNMTF. However, BNMTF requires the number of communities to be set in advance. Zhang et al. [17] applied NMF model for overlapping community detection. This method requires an estimation of the maximum number of communities in advance, same as [22]. However, it is very difficult to set a reasonable value for networks without ground-truth. Recently, Li et al. [15] proposed a method based on semi-supervised matrix factorization and random walk to execute community partition. Wu et al. [10] introduced hypergraph in NMF and utilized the higher-order relationship among the points to promote the clustering performance. However, both of these two methods have the problem that the number of communities is used as a prior information.

Nevertheless, determining the number of communities is still an open issue. Traditional NMF based community detection algorithm can get the number of communities by optimizing the embedded target function. However, these methods are susceptible to various factors, such as the initialized matrix and optimization target function, so it is difficult to accurately determine the number of communities.

To address the aforementioned issues, we use the improved density peak clustering to obtain the number of centers as an NMF input parameter.

B. DENSITY PEAKS CLUSTERING AND ITS RELATED IMPROVED METHODS

Rodriguez and Laio [20] proposed the Density Peaks Method (DPC) which is published in the Journal of Science. The key idea of DPC is that the cluster centers are characterized by higher density than their neighbors and by relatively larger distance from points with higher densities. Here, \( \rho_i \) and \( d_i \) denote the local density of node \( i \) and the distance from \( i \) to points of higher density, respectively.

There are two ways to calculate the local density \( \rho_i \), the first one is defined as:

\[
\rho_i = \sum_j \chi(d(i, j) - d_c),
\]

with

\[
\chi(x) = \begin{cases} 
1, & x \leq 0, \\
0, & \text{otherwise},
\end{cases}
\]

where \( d(i, j) \) is the Euclidean distance between nodes \( i \) and \( j \), \( d_c \) is a cutoff distance, and \( \rho_i \) is the local density of node \( i \).

The second way is defined as Gaussian Kernel function, which is shown as follows:

\[
\rho_i = \sum_j \exp\left(-\frac{d(i, j)^2}{d_c^2}\right).
\]
The minimum distance between node $i$ and any other nodes with higher density, denoted by $\delta_i$, is defined as:

$$\delta_i = \left\{ \begin{array}{ll} \max_j(d(i,j)), & \text{otherwise}, \\ \min_{j: \rho_j > \rho_i}(d(i,j)) & \text{if } \exists j \text{ s.t. } \rho_j > \rho_i. \end{array} \right. \tag{4}$$

Only those nodes with relatively high $\rho_i$ and $\delta_i$ are denoted as cluster centers. After cluster centers are chosen, each remaining node will be assigned to the same cluster as its nearest neighbor with a higher density.

DPC algorithm has several desirable advantages; arbitrary shape of the clusters can be detected, the number of clusters needn’t to be set in advance and the selecting process of centers can be visualized via a decision graph. However, DPC still has deficiencies. First, cut-off distance $d_c$ has a greater impact on the clustering results. Besides, DPC requires human intervention to select appropriate cluster centers.

To tackle the above issues, researchers have made lots of effort [25]–[30], [38]. Xie et al. [31] presented an improved DPC by adopting fuzzy K-nearest neighbors to assign the remaining points to the most probable cluster. Chen et al. [32] proposed the CLUB algorithm by identifying the density backbones, which can automatically select cluster centers. Liang and Chen [29], motivated by a divide and conquer strategy and DPC algorithm, proposed the 3DC algorithm which can automatically detect the correct number of clusters as well. Mehmood et al. [28] proposed the CFSFDP-HD non-parametric algorithm by adopting a heat equation to calculate the density results and reduce the sensitivity of $d_c$. Moreover, Bai et al. [33] proposed the OCDDP algorithm, which is the first method of applying the idea of the DPC to detect community.

As far as we know, there is no existing work for community detection that combines the DPC with NMF-based method and automatically selects cluster centers.

III. METHODS

We’ll introduce the DPCNMF and DPCSNMF algorithms in this section. Start with a brief explanation of the improved DPC method, we will then describe the clustering process of DPCNMF and DPCSNMF in detail. Finally, the time complexity of our proposed algorithms will be analyzed.

The core idea of DPC algorithm is the premise that community centers, which reside far apart from each other, are characterized by a higher density node surrounded by neighbors with lower density nodes. The DPC algorithm includes two steps: first, detecting the density peaks, namely, the cluster centers; second, assigning the remaining nodes to their corresponding clusters. Though the DPC is theoretically efficient, the datasets of complex networks usually have high-dimensional and sparse characteristics, which makes the similarity between nodes meaningless. Moreover, two main indexes $\rho_i$ and $\delta_i$ of DPC mainly depend on the distances between nodes, while the distances between data points are more uniform or even identical. Hence DPC method cannot be employed directly for community detection. To solve this problem, we adopted the improved DPC method to obtain the number of communities as the NMF input parameter.

A. IMPROVED DPC

In this section, we’ll expatiate our DPCSNMF model. Let $G = (V, E)$ be an undirected and unweighted network. The vertex set $V$ contains $N$ nodes, and the edge set $E$ contains $M$ edges, the adjacent matrix of the graph $G$ is $A$, where $a_{ij} = 1$ if there is a link between vertices $i$ and $j$, or 0 if there isn’t.

The local density of a node $i$, denoted by $\rho_i$, is defined as:

$$\rho_i = \text{degree}(V_i). \tag{5}$$

The benefit of employing degree of nodes as local density is that it can avoid the disadvantages caused by cut-off parameter $d_c$ settings and high time complexities.

The calculation of $\delta_i$ is quite simple: the minimum distance between node $i$ and higher density nodes. It is defined as following:

$$\delta_i = \left\{ \begin{array}{ll} \max_j(d_{ij}), & \text{otherwise}, \\ \min_{j: \rho_j > \rho_i}(d_{ij}), & \text{if } \exists j \text{ s.t. } \rho_j > \rho_i. \end{array} \right. \tag{6}$$

where $d_{ij}$ refers to the shortest path distances between node $i$ and $j$, namely hop counts.

The community centers are considered as nodes with relatively high $\rho_i$ and high $\delta_i$. Hence, for each point $i$, we calculate $\gamma_i$ using Eq.(7)

$$\gamma_i = \rho_i \delta_i. \tag{7}$$

Afterward, we sort the elements in $\gamma$ in descending order and denote it as $\gamma^*$. We choose those elements with satisfied $\gamma(hubs(i)) > mean(\gamma) + std(\gamma)$ into the core list, where, $hubs(i)$ presents the potential community centers, $mean(\gamma)$ and $std(\gamma)$ is the mean and the standard deviation of all $\gamma$, respectively. Finally, we compute $\Delta \gamma^*$ using Eq.(8) as following

$$\Delta \gamma^*_j = \gamma^*_i - \gamma^*_j, \tag{8}$$

where $i \in [1, [20\%N]], j = i + 1$, and $[\cdot]$ is the ceiling function. As a rule of thumb, we only consider 20% nodes that have the chance to be selected as cluster centers. Next, we sort $\Delta \gamma^*$ in descending order and use this list to decide the density peaks, viz, the number of communities. To help readers better understand how to choose the number of centers, we present a visualization of selecting peaks in the next section.

The pseudo code of DPCNMF and DPCSNMF algorithms is presented in Algorithm 1. There are two key steps in this algorithm. The first step is the choice of core nodes based on the improved density peak clustering, and the second step is to adopt NMF and SNMF to achieve community partition. There are three sub-steps in the first key step: 1) Calculate $\rho$ by Eq.(5). 2) Calculate $\delta$ by Eq.(6). 3) Select the community core nodes. Where algorithm output: NMI is an abbreviation for normalized mutual information, is defined in Eq.(14). We execute 10 independent experiments and
Hence, in this paper we also adopt SNMF model to achieve community detection task. The objective function of SNMF can be expressed as:

$$\min \|A - HH^T\|^2_F \quad s.t. \quad H \geq 0 \quad \text{where} \quad A = A^T.$$

(12)

Similar to the logics of the MUR for solving NMF, an alternating updating rule can be used to solve the SNMF and the updating rule is as follows:

$$h_{ij} \leftarrow \frac{1}{2} \left[ h_{ij} \left( 1 + \frac{(A^T H)_{ij}}{(HH^T H)_{ij}} \right) \right].$$

(13)

C. COMPLEX ANALYSIS

Our algorithm consists of two key steps. The first step is to adopt the improved DPC to get the number of communities and the second step is to carry out the NMF and the SNMF. The complexity of the algorithm is as follows.

For the first key step, its complexity consists following substeps: computation of $\rho$ with the time complexity of $O(N)$; computation of distance matrix with the time complexity of $O(N^3)$, where $N$ is the number of nodes; calculation of $\delta$ with the time complexity of $O(N^2)$; calculation of $\gamma$ and sorting $\gamma$ with the time complexity of $O(N)$ and $O(N\log N)$, respectively; determining centers require $O(N)$ in the worst case. For the second key step, its time complexity is $O(N^2)$. Thus, the total time complexity of our method is $O(N + N^3 + N^2 + N\log N + N^2)$.

IV. EXPERIMENTS AND RESULTS

To assess our method, we use real and synthetic data that are used in related work or publicly available sources, taken from http://www-personal.umich.edu/~mejn/Netdata/ and http://snap.stanford.edu/data/. The compared methods are as follows: the Newman’s fast algorithm [36], denoted as FN, the louvain method [5], denoted as BGLL, BNMF method [22] by Prorakis et al. and SBMF [17] by Zhang et al. The hardware environment for the experiment is as follows: Inter(R) Core(TM) i3-4160M CPU, 3.60GHz, and 4 GB of memory. The DPCNMF and DPCSNMF algorithms are implemented in MATLAB R2014a.

A. EVALUATION METRICS

In this paper, we apply normalized mutual information (NMI) to evaluate the performance of different algorithms. NMI [37] is a widely used similarity measure metric, which is originated from information theory and proved to be reliable. NMI characterizes the similarity between the true community partition and the partition obtained by the algorithm. Let $C$ be the confusion matrix whose element $C_{ij}$ denotes the number of shared vertices between real community $A$ and detected community $B$. NMI($A, B$) is defined as:

$$\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{CA} \sum_{j=1}^{CB} C_{ij} \log(C_{ij} \pi_C / \pi_A \pi_B)}{\sum_{i=1}^{CA} \sum_{j=1}^{CB} C_{ij} \log(C_{ij} / \pi_C)}.$$

(14)

where $CA$ ($CB$) is the number of real (detected) communities. $C_{ij}$ is the sum of the elements of $C$ in row $i$ (column $j$), and $N$ is the number of nodes. If the partitions are identical.

Algorithm 1 DPCNMF and DPCSNMF Algorithms

Require: The adjacent matrix $A$

Ensure: $label, NMI, NMI_{max}, NMI_{mean}, NMI_{std}$

Step 1: $\rho_i \leftarrow \text{degree}(V_i)$

Step 2: $d_{ij} \leftarrow \text{shortest path distances}$

Step 3: Calculate $\delta_i$ via Eq. (6)

Step 4: Calculate $\gamma_i$ via Eq. (7)

Step 5: $[y_i^*, hubs] \leftarrow \text{sort } y_i \text{ in descending order}$

Step 6: $\Delta y^* \leftarrow \text{diff}(y^*)$

Step 7: $[y, Ind] \leftarrow \text{sort } \Delta y^*$ in descending order

Step 8: cores $\leftarrow$ selection of the community cores

Step 9: Carry out NMF

Step 10: Carry out SNMF

evaluated the results by NMI, and we recorded the maximum, mean and the corresponding standard deviation of NMI as $NMI_{max} = \max(NMI(1), NMI(2), \ldots, NMI(10))$, $NMI_{mean} = \frac{\sum_{i=1}^{10} NMI(i)}{10}$, $NMI_{std} = \sqrt{\frac{\sum_{i=1}^{10} (NMI(i) - NMI_{mean})^2}{n-1}}$, where $i = [1, 2, \ldots, 10]$, $n = 10$.

B. NMF AND SNMF COMMUNITY DETECTION

Given a set of nonnegative data matrix $A \in \mathbb{R}^{M \times N}$, the basic idea of NMF [34] is to find two latent variables $F \in \mathbb{R}_+^{M \times K}$ and $G \in \mathbb{R}_+^{N \times K}$, where $K$ is the dimension of representation. The matrix $A$ can be approximated by the product of these two matrices, namely,

$$A \approx FG^T.$$ 

(9)

In order to find an approximate decomposition process, an objective function must be defined to guarantee the effect of the approximation. Usually, Frobenius norm is used and the following objective function is minimized by NMF:

$$\min \|A - FG^T\|^2_F \quad s.t. \quad F \geq 0, G \geq 0,$$

(10)

where $\| \cdot \|$ is the matrix Frobenius norm. $F \geq 0$ and $G \geq 0$ indicate that the elements of $F$ and $G$ are nonnegative and, usually, $K \ll \min(M, N)$. $F$ can be explained as a basis vector matrix and $G$ is the coefficient matrix. Hence, we can interpret the result as a hard clustering by assigning the $i$th data point to the index that corresponds to the largest element in the $i$th row of $G$.

The multiplicative updating rules (MUR) updating rules for the Euclidean distance objective function are as follows:

$$f_{ij} \leftarrow f_{ij} \frac{(AG)^{ij}}{(FG^T G)^{ij}},$$

$$g_{ij} \leftarrow g_{ij} \frac{(A^T F)^{ij}}{(GF^T F)^{ij}}.$$ 

(11)

SNMF is a variant of NMF and is used for symmetric matrices, which is necessary for clustering non-linearly separable data [35]. Whereas, we don’t know exactly the data is linear or nonlinear before carrying out the experiment. Hence, in this paper we also adopt SNMF model to achieve
(A = B), then $NMI(A, B) = 1$. If A and B are completely different, then $NMI(A, B) = 0$.

In order to evaluate the performance of networks without ground-truth, we use the widely accepted modularity function, namely $Q$ [36] as the metric in the experiment. The modularity is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \theta(c_i, c_j), \tag{15}$$

where $A_{ij}$ represents the adjacency matrix with row and column as node $i$ and $j$ with $k_i, k_j$ as the degree of node $i$ and $j$, $m$ is the total number of edges in the graph and $c_i$ is the community to which vertex $i$ is assigned. $\theta(c_i, c_j)$ is defined as follow:

$$\theta(c_i, c_j) = \begin{cases} 1, & (i, j) \in c, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

### B. PARAMETER SETTING

For fair comparison, the parameters in compared methods were set according to the original papers. For BGLL, FN, DPCNMF, DPCSNMF methods which are parameter-free, community detection is implemented based on the topological information. For BNMF and SBMF methods, the maximum number of communities is required as an input parameter, so the max $K$ is predefined according to the result of BGLL and FN.

### C. SYNTHETIC NETWORKS AND PERFORMANCE COMPARISON

Firstly, we execute our approach on two types of synthetic networks—the GN benchmark networks [1] and H134 benchmark network. Each GN benchmark network consists of four equally-sized non-overlapping communities with 32 nodes. Each node has $Z_{in}$ edges connecting with other nodes in the same community and $Z_{out}$ edges connecting with the other three communities. On average, the total degree of each node is $Z_{in} + Z_{out} = 16$. As expected, the community structure becomes less clear and the community detection task becomes more challenging as $Z_{out}$ increases.

To demonstrate our methods’ efficiency further, we adopt H134 benchmark graph. H134 has a hierarchical community structure, 256 nodes and two hierarchical levels. At a higher level, there are 4 communities, each with 64 nodes. Each high-level community has embedded four small communities, each small community with 16 nodes. Fig. 1 is the visualization of the sparsity pattern of GN8 network and H134 network.

In Fig. 2, elements of $\gamma$ are plotted. It can be seen from Fig. 2, four centers can be detected from GN8 and H134 network; $\text{GN8 hubs} = [51, 8, 112, 83, 63, 111, 118]$ and $\text{H134 hubs} = [14, 82, 68, 213, 131, 200, 5, 7, 76, 123]$. The recommended number of communities are $\text{Ind} = [4, 2, 3, 7]$ and $\text{Ind} = [6, 2, 4, 11]$, respectively. Owing to $\gamma(63) = 22 < \text{mean}(\gamma) + \text{std}(\gamma) = 28.3387$, the cores list of GN8 is [51, 8, 112, 83]. Similarly, for H134 network...
\( \gamma(5) = 25 < \text{mean}(\gamma) + \text{std}(\gamma) = 26.2578 \), so the cores list of H134 is [14, 82, 68, 213, 131, 200]. Now, we can utilize the amendatory recommended number of communities as the input parameter of DPCNMF and DPCSNNMF.

Fig. 3 shows two evaluation metrics to compare six algorithms’ performance on GN8 and H134 networks. For probabilistic algorithms like BNMF, SBMF, DPCNMF and DPCSNNMF, we execute 10 independent experiments and record the mean and the corresponding standard deviation of NMI and \( Q \). For deterministic algorithms like BGLL and FN, only one run is executed. As can be seen from Fig. 3, the performance of DPCSNNMF and BGLL is nip and tuck on GN8 network but better than the other four algorithms; \( \text{NMI}_{BGLL} = 0.5728 > \text{NMI}_{DPCSNNMF} = 0.5605 \), and \( Q_{BGLL} = 0.2471 < Q_{DPCSNNMF} = 0.2828 \). Likewise, for H134 network, performance of DPCSNNMF, FN and BGLL ranks first, the value of \( \text{NMI} = 1 \) and \( Q = 0.6702 \) superior to the other three algorithms. Note that although the average performance score of DPCNMF is not high, its maximum value on \( \text{NMI} \) reaches 1 and \( Q \) value is the best among all 6 algorithms.

### D. REAL WORLD NETWORKS PERFORMANCE COMPARISON

To further evaluate our algorithm, we choose some representative networks with different sizes, tabulated in Table 1.

#### TABLE 1. Attribute description of real-world networks.

| Networks | N  | M   | \( \bar{\kappa} \) | \( K \) | Descriptions                                      |
|----------|----|-----|-------------------|-------|--------------------------------------------------|
| Karate   | 34 | 78  | 4.59              | 2     | Zachary’s karate club                             |
| Dolphin  | 62 | 159 | 5.13              | 2     | Dolphin social network                            |
| Risk     | 42 | 83  | 7.91              | 6     | A map of the popular board game Risk               |
| Football | 115| 616 | 10.66             | 12    | American College football                         |
| Lesmis   | 77 | 254 | 6.60              | -     | Victor Hugo novel Les Miserables                  |
| Book     | 105| 441 | 8.40              | 3     | Books about US politics                           |
| School6  | 69 | 220 | 6.38              | 6     | High school friendship network                    |
| School7  | 69 | 220 | 6.38              | 7     | High school friendship network                    |
| SFI      | 118| 200 | 3.39              | -     | Collaboration network scientists at the Santa Fe  |
| Jazz     | 198| 2742| 27.70             | -     | Network of Jazz musicians                         |
| Adjnoun  | 112| 425 | 7.59              | 2     | Charles Dickens novel David Copperfield           |
| Polblogs | 1224| 16717| 27.32           | 2     | Blogs about politics                              |
| Egonet   | 4039| 88234| 43.69            | -     | Social circles from Facebook                      |

Here, \( n \) denotes the number of nodes, \( m \) denotes the number of edges, \( \bar{k} \) denotes the average of node degree, \( K \) denotes the number of communities, and ‘\(-\)’ implies that the number of communities is unknown.

Fig. 4 is the decision graphs of 12 real-world networks. Decision graph of Egonet network is omitted due to limited space. We sort \( \gamma \) values in descending order and gain hubs and \( \text{Ind} \), namely, density peaks and the recommended number of communities. Due to space constraints, only three networks’ automatic selection for centers are discussed in detail. \( \text{Karate\_hubs} = [34, 1, 33, 3, 2, 4] \), \( \text{Karate\_Ind} = [2, 5, 3, 4] \) and \( \gamma_{\text{Karate}}(33) = 12 < (\text{mean}(\gamma_{\text{Karate}}) + \text{std}(\gamma_{\text{Karate}}) = 12.9369) \). Therefore, Karate cores list only includes [34,1] and the amendatory recommended number of communities is 2. \( \text{Dolphins\_hubs} = [15, 18, 21, 38, 46, 34, 52] \), \( \text{Dolphins\_Ind} = [2, 3, 5, 7] \) and \( \gamma_{\text{Dolphins}}(38) = 11 < (\text{mean}(\gamma_{\text{Dolphins}}) + \text{std}(\gamma_{\text{Dolphins}}) = 13.6775) \). Therefore, Dolphins cores list only includes [15,18,21] and the amendatory recommended number of communities is 2 and 3. \( \text{Risk\_hubs} = [5, 12, 23, 16, 26, 33, 36] \), \( \text{Risk\_Ind} = [2, 3, 7, 10] \), \( \text{mean}(\gamma_{\text{Risk}}) < (\gamma_{\text{Risk}}(16, 26, 33, 36) = 6 < (\text{mean}(\gamma_{\text{Risk}}) + \text{std}(\gamma_{\text{Risk}}) = 9.5586), \text{and } \rho_{\text{Risk}}(16, 26, 33, 36) = \rho_{\text{max}} > (\text{mean}(\rho_{\text{Risk}}) + \text{std}(\rho_{\text{Risk}}) = 5.1201). \text{So, Risk cores list only includes [5,12,23,16,26,33,36] and the amendatory recommended number of communities is 2,3 and 7. The superiority of this method lies in that cluster centers can be automatically
extracted, which eliminates the shortcoming of cluster centers determination by manual selection.

Table 2 and Table 3 show the performance of six algorithms on real-world network in terms of the $Q$ and $NMI$ values, respectively. In the two tables, the number in the bracket of each cell represents the number of communities detected by a certain algorithm. The entries corresponding to the ground-truth’s best detection result are emboldened and second-best are underlined. For BNMF, SBMF, DPCNMF and DPCSNMF which are probabilistic algorithms, we execute 10 independent experiments and record the maximum $Q$ and $NMI$. Owing to space constraints, we omit the mean and the corresponding standard deviation of $Q$ and $NMI$. For BGLL and FN which are deterministic algorithms, only one run is executed. Since BNMF and SBMF have to scan $K$ in a large range, we predefined maximum $K$ for BNMF and SBMF according to the result of BGLL or FN for the sake of computation efficiency.

As can be seen from Table 2, although $Q$ value of our methods is not the highest among six algorithms for some datasets, the result of DPCNMF and DPCSNMF is consistent with reality. For example, on the karate network, DPCNMF and DPCSNMF achieve $NMI = 1$ but only achieve $Q = 0.3715$. However, we know that $Q = 0.3715$ is indeed the true partition result of the karate network according to Table 3. Based on the result of the $NMI$ and modularity $Q$ in Table 3 and Table 2, we find that the higher modularity value does not always correspond to the better partition of the network. By observing Table 2, we also note that the $Q$ value of DPCNMF and DPCSNMF is slightly inferior to the compared algorithms on School7 network, Jazz network, and Egonet network. However, this does not suggest that our methods perform poorly because it tends to detect large community structure. School6 network and School7 network are the same network with different labels so the detection result of our two methods are same, and the number of community
TABLE 2. Comparison in terms of $Q$ on real-world networks (The entries corresponding to the ground-truth’s best detection result are emboldened and second-best are underlined, for networks without ground-truth we don’t make the mark).

| Networks | BGLL | FN | BNMF | SBMF | DPCNMF | DPCSNMF |
|----------|------|----|------|------|--------|---------|
| Karate   | 0.4188(4) | 0.3807(3) | 0.3991(3) | 0.4026(4) | 0.3715(2) | 0.3715(2) |
| Dolphins | 0.5188(5) | 0.5049(4) | 0.5037(4) | 0.4918(3) | 0.3899(2) | 0.3848(2) |
| Risk     | 0.6337(7) | 0.6248(6) | 0.6337(7) | 0.6012(7) | 0.5504(7) | 0.6337(7) |
| Football | 0.6046(10) | 0.5497(6) | 0.6029(9) | 0.6005(12) | 0.5652(13) | 0.5932(13) |
| Lesmis   | 0.5556(6) | 0.5006(5) | 0.5123(6) | 0.5382(6) | 0.5177(4) | 0.5502(7) |
| Book     | 0.4986(3) | 0.5020(4) | 0.5241(4) | 0.5062(4) | 0.4978(3) | 0.5124(3) |
| School6  | 0.5927(7) | 0.5872(5) | 0.5976(6) | 0.5831(6) | 0.5775(6) | 0.5871(6) |
| School7  | 0.5927(7) | 0.5872(5) | 0.5976(6) | 0.5831(6) | 0.5775(6) | 0.5871(6) |
| SFI      | 0.7497(8) | 0.7497(8) | 0.7303(7) | 0.6951(8) | 0.6355(3) | 0.7326(7) |
| Jazz     | 0.4431(4) | 0.5430(4) | 0.3892(4) | 0.3544(3) | 0.4062(3) | 0.1845(7) |
| Adjoinn  | 0.2906(7) | 0.2947(7) | -0.0856(6) | 0.2021(2) | -0.2718(2) | 0.2012(2) |
| Polblogs | 0.4270(10) | 0.4270(11) | 0.4195(7) | 0.3613(7) | 0.4248(2) | 0.4252(2) |
| Egonet   | 0.8287(17) | 0.7774(13) | 0.8264(16) | - | 0.8074(8) | 0.8121(8) |

TABLE 3. Comparison in terms of $NMI$ on real-world networks (The entries corresponding to the best detection result are emboldened and second-best are underlined).

| Networks | BGLL | FN | BNMF | SBMF | DPCNMF | DPCSNMF |
|----------|------|----|------|------|--------|---------|
| Karate   | 0.5866(4) | 0.6925(3) | 0.8255(3) | 0.6759(4) | 1(2) | 1(2) |
| Dolphins | 0.5162(5) | 0.5727(4) | 0.6056(5) | 0.4896(5) | 0.7532(2) | 0.8141(2) |
| Risk     | 0.9453(7) | 0.8939(6) | 0.9453(7) | 0.8919(7) | 0.8020(7) | 0.9453(7) |
| Football | 0.8930(10) | 0.6977(6) | 0.8930(9) | 0.9232(12) | 0.9119(13) | 0.9336(13) |
| Book     | 0.5745(3) | 0.5308(4) | 0.5506(4) | 0.5094(4) | 0.5979(2) | 0.5979(2) |
| School6  | 0.8518(7) | 0.7271(5) | 0.9521(6) | 0.8286(6) | 0.8255(6) | 0.7984(6) |
| School7  | 0.8784(7) | 0.7623(5) | 0.9194(6) | 0.8413(6) | 0.8541(6) | 0.8287(6) |
| Adjoinn  | 0.0037(7) | 0.0052(7) | 0.3489(7) | 0.0186(2) | 0.3742(2) | 0.3406(2) |
| Polblogs | 0.6486(10) | 0.6501(11) | 0.5268(7) | 0.3315(7) | 0.7276(2) | 0.7197(2) |

is six. For Jazz network, the $Q$ value of DPCNMF better than that of DPCSNMF, and the detected number of community is less than or equal to compare methods. The reason may lie that distribution of Jazz network is linear independent. For Egonet network the number of community detected by both of DPCNMF and DPCSNMF is less than that of compared methods.

In addition, from the Table 3, we notice that the maximum value of $NMI$ obtained by DPCNMF method on Karate network, Dolphins network, Football network, Polbooks network and Risk network is 1, 0.8141, 0.9336, 0.5979, 0.9453, respectively. Moreover, DPCSNMF method on Polblogs network is the second-best with $NMI$ of 0.7197. So, DPCSNMF achieves five best and one second-best $NMI$ out of nine real-world networks with ground-truth among six algorithms. This result clearly indicates that DPCNMF and DPCSNMF outperform the compared algorithms and can obtain superior community structure in real-world networks. Note that we assign ground-truth of 12 to BMNF and SBMF because the number of community obtained by BGLL and FN is 10, 6, respectively. Consequently, adopting ground-truth as an input parameter, SBMF not only achieves the best $Q$, but also gains the second-best $NMI$ at Football network.

TABLE 4. Comparison in terms of execution time (in seconds) on real-world networks (The entries corresponding to the fastest time are emboldened. For Egonet network SBMF method does not finish within 24 hours, so we use '-' indicate).

| Networks | BGLL | FN | BNMF | SBMF | DPCNMF | DPCSNMF |
|----------|------|----|------|------|--------|---------|
| Karate   | 0.0312 | 0.0151 | 0.1551 | 2.8622 | 0.0757 | 0.0596 |
| Dolphins | 0.0469 | 0.0606 | 0.3504 | 5.6841 | 0.1554 | 0.0689 |
| Risk     | 0.0273 | 0.0080 | 0.2001 | 6.8409 | 0.1234 | 0.0737 |
| Football | 0.0849 | 0.0170 | 0.7773 | 36.6788 | 0.4617 | 0.2436 |
| Lesmis   | 0.0701 | 0.0178 | 0.5153 | 8.8752 | 0.1453 | 0.1075 |
| Book     | 0.0520 | 0.0203 | 0.5531 | 6.7051 | 0.1655 | 0.0994 |
| School6  | 0.0560 | 0.0190 | 0.4197 | 10.6340 | 0.1913 | 0.1345 |
| School7  | 0.0476 | 0.0200 | 0.4259 | 11.1882 | 0.1765 | 0.1077 |
| SFI      | 0.0592 | 0.0285 | 0.6923 | 19.9560 | 0.1475 | 0.1296 |
| Jazz     | 0.3028 | 0.0569 | 7.6359 | 49.9679 | 0.2535 | 0.1656 |
| Adjoinn  | 0.0805 | 0.0394 | 0.8174 | 15.3645 | 0.2072 | 0.0849 |
| Polblogs | 2.6301 | 12.0642 | 102.6899 | 4995.6652 | 30.9498 | 19.1658 |
| Egonet   | 18.4471 | 571.6403 | 1215.2826 | - | 911.6894 | 656.3098 |
Furthermore, by observing Table 3, we also find that for School6 network, School7 network, Adjnoun network and Polblogs network, DPCNMF’s maximum value of NMI is better than DPCSNMF’s. This is probably because the aforementioned four networks contain linear independent data. Hence, asymmetric DPCNMF should outperform symmetric DPCSNMF. In a nutshell, each clustering method has its suitable scenario. Kuang et al. in [35] discovers that asymmetric NMF performs well when different clusters correspond to linear independent data, while symmetric NMF can capture nonlinear independent subspace. In many cases, the distribution of data is hard to be known in advance, which is the exact reason that we propose DPCNMF and DPCSNMF to deal with different networks.

The time costs are exhibited in Table 4. The fastest algorithms are bolded. From Table 4, we have the following observations. The BNMF, SBMF, DPNMF and DPSNMF methods share a much higher time cost than BGLL and FN. In addition, both BNMF and SBMF consume more running time owing to searching for the best community structure among all candidates. Although our methods suffer from higher computational cost than BGLL and FN, they can generally achieve the best detection performance among all compared methods. It should be noted that the runtime of DPCSNMF is slightly inferior to BGLL for small networks. While for larger networks such as Polblogs network and Egonet network, the running time of BGLL is nearly 7 times and 35 times faster than DPCSNMF, respectively. The proposed methods are not suitable for applying to large-scale networks due to the fact that we use the Floyds algorithm to calculate the shortest distances and its complexity is $O(N^3)$. This bottleneck motivates us to improve our methods to address larger-scale networks in the future.

For the sake of clarity, we choose Karate network and Dolphins network as the case study to visualize the detected communities. The nodes are drawn in different colors corresponding to different communities. The first sub-figure of Fig. 5 is the visualization results of our methods on Karate network, the second sub-figure of the first line is Dolphins network and two sub-figures of the second line are the visualization results of Dolphins network with 2 and 3 communities.
Karate network ($Q = 0.3715$, $NMI = 1$). Both DPCNMF and DPCSNMF can correctly divide Karate network into two communities. The second sub-figure of Fig. 5 is the visualization results of original Dolphins network. The third and the fourth sub-figures of Fig. 5 are the visualization results of our methods on Dolphins network with 2 and 3 communities. As can be seen from the third sub-figure, only node 40 is misclassified by our method.

From Fig. 4, Fig. 5 and above experimental analysis, we can conclude that our methods are capable of detecting communities in different resolutions. If we want to get small groups of communities, we can adopt bigger $K$, vice versa. Adopting different $K$ values is like viewing a network from a coarse-grained view (lower $K$) to a fine-grained view (higher $K$). Therefore, our methods can provide insight about the interior structures of communities in the networks.

V. CONCLUSION

In this paper, we put forward DPCNMF and DPCSNMF algorithms for community detection in complex networks, which combine NMF algorithm with improved DPC algorithm. Compared with traditional NMF-based methods, our methods mainly have three advantages. Firstly, our methods are parameter-free algorithms. Secondly, the proposed methods can automatically detect the correct number of communities rather than manual selection. Thirdly, the improved DPC method ensures the centers selection. Meanwhile, according to the recommended centers nodes, we can detect different scales of community. Finally, our methods have been tested on a series of benchmark graphs and real-world networks. The experimental results show that the DPCNMF and DPCSNMF methods have better performance in uncovering indistinct community structure compared with other state-of-the-art methods.

However, there is still room for improvements in terms of performance. The potential avenue for future work includes how to extend our methods to handle extremely larger-scale networks and considering how to integrate constraint information into our model and we’ll mainly focus on the semi-supervised community detection algorithm.

ACKNOWLEDGMENT

The authors would like to thank Zhu Shen for giving very valuable advice on their manuscript and providing the polish of the manuscript. They would also like to thank anonymous reviewers for their help in improving the manuscript.

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] H. Lu, X. Zhu, H. Liu, G. Skogerboe, J. Zhang, Y. Zhang, L. Cai, Y. Zhao, S. Sun, and J. Xu, “The interactome as a tree:—An attempt to visualize the protein-protein interaction network in yeast,” Nucleic Acids Res., vol. 32, no. 16, pp. 4804–4811, Sep. 2004.
[3] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, “Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters,” Internet Math., vol. 6, no. 1, pp. 29–123, Jan. 2009.
[4] M. Newman, Networks: An Introduction. London, U.K.: Oxford Univ. Press, 2010.
[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” J. Stat. Mech., vol. 2008, no. 10, Oct. 2008, Art. no. P10008.
[6] J. Huang, H. Sun, Q. Song, H. Deng, and J. Han, “Revealing density-based clustering structure from the core-connected tree of a network,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 8, pp. 1876–1889, Aug. 2013.
[7] 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.
[8] Z.-Y. Zhang, “Community structure detection in complex networks with partial background information,” Europhys. Lett., vol. 101, no. 4, Feb. 2013, Art. no. 48005.
[9] L. Yang, D. Jin, X. Wang, and X. Cao, “Active link selection for efficient semi-supervised community detection,” Sci. Rep., vol. 5, Mar. 2015, Art. no. 9039.
[10] W. Wu, S. Kwong, Y. Zhou, Y. Jia, and W. Gao, “Nonnegative matrix factorization with mixed hypergraph regularization for community detection,” Inf. Sci., vol. 435, pp. 263–281, Apr. 2018.
[11] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, “A unified semi-supervised community detection framework using latent space graph regularization,” IEEE Trans. Cybern., vol. 45, no. 11, pp. 2585–2598, Nov. 2015.
[12] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” Phys. Rep., vol. 659, pp. 1–44, Nov. 2016.
[13] S. Fortunato, “Community detection in graphs,” Phys. Rep., vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.
[14] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” ACM Comput. Surv., vol. 45, no. 4, pp. 1–35, Aug. 2013.
[15] W. Li, J. Xie, M. Xin, and J. Ma, “An overlapping network community partition algorithm based on semi-supervised matrix factorization and random walk,” Expert Syst. Appl., vol. 91, pp. 277–285, Jan. 2018.
[16] N. Chen, Y. Liu, and H.-C. Chao, “Overlapping community detection using non–negative matrix factorization with orthogonal and sparseness constraints,” IEEE Access, vol. 6, pp. 21266–21274, 2018.
[17] Z. Zhang, Y. Wang, and Y. Ahn, “Overlapping community detection in complex networks using symmetric binary matrix factorization,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 87, no. 6, 2013, Art. no. 062803.
[18] X. Ma, L. Gao, X. Yong, and L. Fu, “Semi-supervised clustering algorithm for community structure detection in complex networks,” Phys. A, Stat. Mech. Appl., vol. 389, no. 1, pp. 187–197, Jan. 2010.
[19] X. Shi, H. Lu, Y. He, and S. He, “Community detection in social network with pairwisely constrained symmetric non-negative matrix factorization,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), 2015, pp. 541–546.
[20] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[21] Y. Chen, P. Zhao, P. Li, K. Zhang, and J. Zhang, “Finding communities by their centers,” Scientific reports, vol. 6, p. 24017, 2016.
[22] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, “Overlapping community detection using Bayesian non-negative matrix factorization,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 83, no. 6, 2011, Art. no. 066114.
[23] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, “Community discovery using nonnegative matrix factorization,” Data Mining Knowl. Discovery, vol. 22, no. 3, pp. 495–521, May 2011.
[24] Y. Zhang and D.-Y. Yeung, “Overlapping community detection via bounded nonnegative matrix tri-factorization,” in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2012, pp. 606–614.
[25] M. Du, S. Ding, and H. Jia, “Study on density peaks clustering based on k-nearest neighbors and principal component analysis,” Knowl.-Based Syst., vol. 99, pp. 135–145, May 2016.
[26] M. Xu, Y. Li, R. Li, F. Zou, and X. Gu, “EADP: An extended adaptive density peaks clustering for overlapping community detection in social networks,” Neurocomputing, vol. 337, pp. 287–302, Apr. 2019.
[27] Z. Li and Y. Tang, “Comparative density peaks clustering,” Expert Syst. Appl., vol. 95, pp. 236–247, Apr. 2018.
[28] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, “Clustering by fast search and find of density peaks via heat diffusion,” Neurocomputing, vol. 208, pp. 210–217, Oct. 2016.
[29] Z. Liang and P. Chen, “Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering,” Pattern Recognit. Lett., vol. 73, pp. 52–59, Apr. 2016.

[30] L. Yaohui, M. Zhengming, and Y. Fang, “Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy,” Knowl.-Based Syst., vol. 133, pp. 208–220, Oct. 2017.

[31] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, “Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors,” Inf. Sci., vol. 354, pp. 19–40, Aug. 2016.

[32] M. Chen, L. Li, B. Wang, J. Cheng, L. Pan, and X. Chen, “Effectively clustering by finding density backbone based on kNN,” Pattern Recognit., vol. 60, pp. 486–498, Dec. 2016.

[33] X. Bai, P. Yang, and X. Shi, “An overlapping community detection algorithm based on density peaks,” Neurocomputing, vol. 226, pp. 7–15, Feb. 2017.

[34] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[35] D. Kuang, C. Ding, and H. Park, “Symmetric nonnegative matrix factorization for graph clustering,” in Proc. SIAM Int. Conf. Data Mining, Apr. 2012, pp. 106–117.

[36] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 69, no. 6, 2004, Art. no. 066133.

[37] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification,” J. Stat. Mech., Theory Exp., vol. 2005, no. 9, Sep. 2005, Art. no. P09008.

[38] D. Yu, G. Liu, M. Guo, X. Liu, and S. Yao, “Density peaks clustering based on weighted local density sequence and nearest neighbor assignment,” IEEE Access, vol. 7, pp. 34301–34317, 2019.

[39] Y. Lei, Y. Zhou, and J. Shi, “Overlapping communities detection of social network based on hybrid C-means clustering algorithm,” Sustain. Cities Soc., vol. 47, May 2019, Art. no. 101436.

HONG LU received the M.S. degree in computer application from the Taiyuan University of Science and Technology, Taiyuan, China, in 2012. She is currently pursuing the Ph.D. degree in pattern recognition and intelligence system with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.

Her current research interests include community detection in complex networks and data mining.

XIAOSHUANG SANG received the B.S. degree from Huaibei Normal University, Huaibei, China, in 2014, and the M.S. degree from the School of Mathematical Sciences, Anhui University, Hefei, China, in 2017. She is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.

Her current research interests include pattern recognition, optimization algorithm, and metric learning.

QINGHUA ZHAO received the B.S. degree from Xingtai University, Xingtai, China, in 2006, the M.S. degree from Northwest Minzu University, Lanzhou, China, in 2011, and the Ph.D. degree in pattern recognition and intelligent systems from the Nanjing University of Science and Technology, Nanjing, China, in 2018.

From 2013 to 2015, he was a Visiting Scholar with the University of Georgia, Athens, USA. He is currently an Assistant Professor with the College of Information Engineering, Nanjing University of Finance and Economics. His current research interests include pattern recognition, medical image analysis, and optimization algorithm.

HONG LU received the M.S. degree in computer application from the Taiyuan University of Science and Technology, Taiyuan, China, in 2012. She is currently pursuing the Ph.D. degree in pattern recognition and intelligence system with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.

Her current research interests include community detection in complex networks and data mining.

JIANFENG LU received the B.S. degree in computer software and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from the Nanjing University of Science and Technology, Nanjing, China.

He is currently a Professor with the Nanjing University of Science and Technology. His research interests include image processing, pattern recognition, and data mining.

VOLUME 8, 2020 5759