Sparse Nonnegative Matrix Factorization for Multiple-Local-Community Detection

Dany Kamuhanda, Meng Wang, and Kun He, *Senior Member, IEEE*

***Abstract*— Local community detection consists of finding a group of nodes closely related to the seeds, a small set of nodes of interest. Such groups of nodes are densely connected or have a high probability of being connected internally than their connections to other clusters in the network. Existing local community detection methods focus on finding either one local community that all seeds are most likely to be in or finding a single community for each of the seeds. However, a seed member usually belongs to multiple local overlapping communities. In this work, we present a novel method of detecting multiple local communities to which a single-seed member belongs. The proposed method consists of three key steps: 1) local sampling with Personalized PageRank (PPR); 2) using the sparseness generated by a sparse nonnegative matrix factorization (SNMF) to estimate the number of communities in the sampled subgraph; and 3) using SNMF soft community membership vectors to assign nodes to communities. The proposed method shows favorable accuracy performance and a good conductance compared with state-of-the-art community detection methods by experiments using a combination of artificial and real-world networks.**

***Index Terms*— Clustering, local community detection, nonnegative matrix factorization (NMF), social networks, sparseness.**

I. Introduction

MANY complex data, such as user interaction in social networks, product purchases, scientific collaboration, and interaction among proteins of an organism, are represented by a graph (network) consisting of nodes connected by edges indicating the interactions [1]. There are three tasks that dominate the area of network analysis: 1) node classification that consists of predicting the label of a target node based on other labeled nodes [2], [3]; 2) link prediction that consists of predicting an edge between two unconnected nodes [4], [5]; and 3) community detection that finds a set of closely related nodes within a network [6], [7]. We are interested in community detection as this task can also be used to address the other two tasks. For node classification, a node can be assigned a label carried by its community members; and for link prediction, nodes of the same community are more likely to connect with each other compared with nodes from different communities.

A community is a group of nodes that are densely connected (cluster) or have a high probability of being connected internally than their connections to the nodes in other clusters of the network [8]. Community detection algorithms include: 1) global algorithms to detect all communities on the entire network [9]–[14]; 2) local algorithms that find a single local community to which a seed set belongs [15]–[18]; and 3) local algorithms that find multiple local communities around a seed set [19]–[21].

Community detection methods may use network structural information, metadata (node attribute), or a combination of the two [22]–[24]. Li *et al.* [22] optimize two objective functions: one custom function for node similarity and another (modularity) for network structure. As network metadata is not available in most cases, most community detection methods use network structural information only.

Community detection addresses a number of real-world problems. In social networks, one may be interested in finding all groups of family members or friends in a network for a specific reason, such as epidemic control. In communication networks, a suspect group linked to known criminals is more likely to be identified. Community detection can also be used for feature selection to speed up a machine learning algorithm [25] or find customer shopping patterns in recommendation systems.

As real-world networks (such as Facebook) are very large with millions or billions of nodes, detecting all communities globally becomes a computationally intensive task. Moreover, most of the time we are only interested in a subset of nodes of a local region. In such cases, the local community detection provides a solution [16], [26], [27]. For instance, a chef may know a few ingredients used for a particular recipe, but these seeding ingredients are not available on the market. Using a flavor network, local community detection methods can identify similar ingredients that could replace these ingredients without having to detect all communities of the flavor network [20], [28]. Likewise, recommender systems can use the similarity among consumers with similar shopping views or who often purchase the same products to recommend relevant products to a consumer [29]. Single-local-community detection assumes that all seeds belong to the same community, and the task is to find missing members of the community. Bian *et al.* [20] assumed that multiple seeds may belong to

Manuscript received January 19, 2020; revised April 27, 2020 and June 27, 2020; accepted July 9, 2020. Date of publication July 24, 2020; date of current version November 10, 2020. This work was supported in part by the National Natural Science Foundation under Grant U1836204, and in part by the Fundamental Research Funds for Central Universities under Grant 2019kfyXKJC021. The work of Dany Kamuhanda was supported in part by the Ministry of Commerce (MOFCOM) of the People's Republic of China, and in part by the University of Rwanda. *(Corresponding author: Kun He.)*

The authors are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: kamuhanda@hust.edu.cn; mengwang233@hust.edu.cn; brooklet60@hust.edu.cn).

Digital Object Identifier 10.1109/TCSS.2020.3008860

different communities and find a single community for each seed, which becomes a single-local-community detection task for each seed.

We address a more challenging problem that is rarely addressed in the literature. Given a single seed, the task is to find all possible communities that the seed belongs to. Compared with the multiple-local-community detection, the problem of finding one single community, which a seed is most likely to be in, is easier as this community is denser or stronger than other local communities this seed is in. The multiple-local-community detection problem is specifically difficult because we cannot use more than one seed to improve the accuracy as in other existing local community detection tasks. For this task, even when we know some nodes that belong to the same community as the seed, they are not helpful for improving the accuracy since they do not belong to all communities of the seed. In addition, as nodes belong to multiple communities, the overlapping portion becomes very dense and may sometimes look like a single community that is difficult to split into different communities. The problem of multiple community detection for a single seed was first introduced in [21], but the authors focused more on single-local-community detection. In our previous conference publication [19], we focused on the multiple-local-community detection for a single seed based on nonnegative matrix factorization (NMF). This work is an extended version with significant improvement in [19].

In this work, we propose a Sparseness-based Multiple-Local-Community (S-MLC) detection method for finding multiple local communities of a single seed. There are three key steps in S-MLC: local sampling to find relevant nodes for the seed; estimating the number of communities in the sampled subgraph; and detecting the local communities. We use sparse NMF (SNMF) to learn the structural information of a network as SNMF can find better representations than NMF [30], [31]. We then use soft community membership vectors generated by SNMF to assign nodes to their corresponding communities. Other network embedding methods, such as those based on graph neural networks (GNNs) [32], [33], Node2Vec [34], and DeepWalk [35], require the input node features that are not always available or generate embeddings that are difficult to interpret. The nonnegativity feature of SNMF allows each node's embedding vector to be interpreted as probabilities of belonging to different communities and makes it more suitable for this task.

S-MLC uses a similar framework of our conference version of MLC [19] in terms of local sampling and community estimation. MLC uses a BFS for local sampling, while S-MLC uses Personalized PageRank (PPR), which is often used for local sampling in existing single-local-community detection methods [27]. PPR is computationally intensive, but efficient approximation [15] can be used. To estimate the number of communities, MLC iterates NMF decompositions until the normalized H (values of each column sum up to 1) contains a row without a centroid node. S-MLC estimates communities by iterating SNMF on the sampled subgraph where the number of components that yield the maximum sparseness is used as the number of communities. This approach is based on the sparse coding that aims to find a few elements that can effectively represent the entire population. Regarding the final phase of community detection, both S-MLC and MLC use a threshold on community membership vectors generated by NMF/SNMF algorithm to assign nodes to communities.

Our contributions are summarized as follows. First, we propose a method of estimating the number of communities in social networks, which can be used in other community detection algorithms that require the number of communities as prerequisites. Second, we investigate and conduct extensive experiments on local sampling techniques to determine which techniques and parameters are suitable for sampling a subgraph containing almost all members of multiple local communities of a given seed. Especially, we demonstrate the sampling behavior of PPR and heat kernel (HK), which is useful in making a good choice between the two methods depending on the application scenario. Third, we address a challenging problem of finding multiple local communities for a single seed, which is rarely addressed in the literature, and propose a novel method, called S-MLC for solving this problem. The proposed approach outperforms state-of-the-art baselines via extensive experiments. S-MLC outperforms MLC [19] and M-LOSP [21] as evaluated on artificial and real-world networks. As there are few algorithms for multiple-local-community detection, we further run DEMON on the sampled subgraph to find approximate local communities, denoted as L-DEMON, and both S-MLC and MLC clearly outperform L-DEMON on most networks.

The rest of the article is organized as follows. Section II discusses the related work. Section III introduces the addressed problem, measures of evaluating the solution quality, local sampling with a focus on PPR, and how NMF is used for community detection. Section IV discusses the three steps of the proposed method for detecting communities of a single seed. Section V presents experimental results, followed by a conclusion.

II. Related Work

A. Local Community Detection

There are two types of problem formulation for the local community detection: 1) assuming all seeds belong to one community and 2) assuming the seeds belong to multiple communities simultaneously.

Clauset [36] and Chen *et al.* [37] iteratively expand a community C, which initially consists of a seed set by adding more members one-by-one from its boundary. A node is removed from the boundary and added to C; if it improves the local modularity [36] or the internal relation [37], then its neighbors join the boundary. The advantage of this approach is that we can query and get current community members at any time even when the algorithm is still processing. PPR [15], [38] and HK [17] based methods also assume that all seeds belong to the same community, then consider the seed set as an initial community, and grow the community by sorting their probability vector p in the decreasing order to obtain q followed by finding a set of nodes with a minimum conductance. The advantage of these approaches is that the

more that initial seeds we have, the higher the accuracy of the output. Spectral-based methods, such as LOSP [21], [27], find a local spectral basis supported by the seeds through a few steps of random walks; then, a sparse vector y is computed. This vector consists of probabilities that indicate the extent to which each node is likely to belong to the same community as the seed set. He *et al.* [27] systematically built a family of local spectral subspace-based methods through various diffusions from the seeds to form the local spectral subspace.

He *et al.* [21] discussed a method of extending their single-local-community detection algorithm called LOSP to find all the communities a single seed belongs to. It is the first work to address the multiple-local-community detection problem. They temporarily remove the seed from its ego network to get connected components; then, for each connected component, they add the seed back to build an initial seed set and use LOSP as a subroutine to find the local community for each initial seed set. This approach allows detecting multiple local communities, and we denote it as M-LOSP. Our previous work of MLC [19] uses a few steps of the breadth-first search for local sampling and then uses an NMF to learn the network structure encoded in the adjacency matrix so that nodes can be assigned to communities based on a threshold applied to community membership vectors generated by NMF. Hollocou *et al.* [39] used local scoring metrics (PPR [15], HK [17], and LEMON [26] score) to define embedding of the graph around the seed set and pick new seeds based on the embedding to uncover multiple communities, but the newly found communities are not for the original seeds but for the newly determined seeds.

B. Network Embedding

Recent community detection methods consist of encoding a network so that nodes belonging to the same community yield similar representation compared with nodes belonging to different communities. Such a representation is obtained by learning the network features using SVD [40], NMF [19], Autoencoder [41], DeepWalk [35], Node2Vec [34], or GNN [3] [33].

Rahimi *et al.* [42] encoded the input network using a set of locus-based adjacency representation. Each representation consists of n nodes, and each node is linked with a random neighbor. Connected components of each representation become communities, a solution generated by that representation. Solutions generated by each representation are compared by optimizing both the kernel K-means (KKM) and ratio cut (RC), and eligible solutions are retained. Among these retained solutions, the best is the one with a good normalized mutual information (NMI). This method has the advantage of not requiring the number of communities. However, it is applicable to global community detection and not for local community detection.

Mahmood and Small [43] generated a new representation of networks by computing geodesic distances of nodes and then use a sparse linear coding to decompose the obtained representation into a matrix of coefficients, which is clustered using spectral clustering to obtain communities.

NMF has the advantage of obtaining communities without requiring an additional clustering method, and its nonnegativity feature makes it easy to detect both overlapping and nonoverlapping communities [19]. Recent NMF-related works detect global communities by adding a regularization term that incorporates local and global network information for improving accuracy [44], [45]. The representations obtained using other mentioned methods are mainly used for nonoverlapping community detection or some other tasks, such as link-prediction or node classification. Especially, GNNs are suitable for networks with node labels in semisupervised tasks where available labels can be used in predicting unknown labels. Nonoverlapping communities can be detected using a clustering algorithm, such as k-means, with an input k as the number of communities and a representation of the network. The quality of the detected communities not only depends on the type of network representation but also on the choice of the community number.

C. Estimating the Number of Communities

Eigengap is a heuristic that is often used to estimate the number of clusters in a network. The estimated number of clusters k is based on identifying small gaps between consecutive eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ of a Laplacian matrix, followed by the largest gap between λ_k and λ_{k+1} [40], [46]. In some cases, such as in networks with overlapping communities, there is no clear gap, and eigengap heuristic does not work [46].

The modularity maximization model [47] is the most popular method for estimating the number of communities. It is based on a modularity matrix that encodes the eigenvectors of a network and is calculated as

$$B_{ij} = A_{ij} - \frac{d(v_i)d(v_j)}{2m}$$

where A represents the adjacency matrix, $d(v_i)$ is the degree of a node v of the ith index in A, and m is the total number of edges in the network. For a network of two disjoint communities with $h_i \in \{1, -1\}$, the community membership indicator for a node v_i, namely, the modularity, is computed as [47]

$$Q = \frac{1}{4m} h^T B h$$

where h is the column vector whose elements are h_i. The decomposition of a network into more than two communities can be done hierarchically by dividing the network into two communities, then each of them divided into two subcommunities, and so on up to nonincreasing modularity. As B is symmetric, there exists a decomposition such that $B \cong U \Lambda U^T$, where U is a matrix of eigenvectors and Λ is a matrix of eigenvalues on the diagonal. By extracting some eigenvectors from U, we have H that corresponds to the top k eigenvalues in Λ, sorted in nonincreasing order. The generalized modularity for more than two communities can be computed as [41]: $Q = \sum_{ii} H^T B H$. The final number of communities in a network is the one that maximizes modularity Q from various iterations of k.

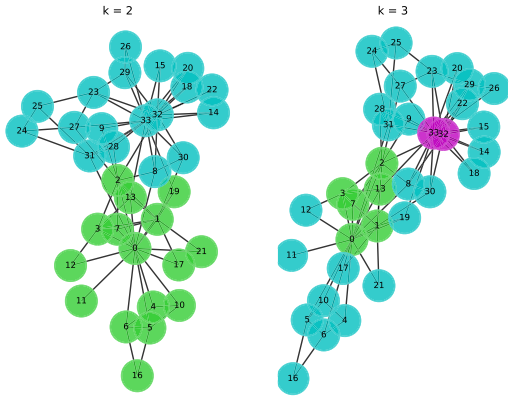


Fig. 1. Clusters found by k-means for different values of k on the Zachary Karate club network. The choice of the community number greatly affects the quality of community detection.

III. Preliminaries

A. Problem Formulation

We are given a network modeled as a graph $G = (V, E)$ and a seed $s \in V$, where G is an undirected, unweighted graph with n nodes $V = \{v_1, \ldots, v_n\}$, and m edges $E = \{e_1, \ldots, e_m\} \subseteq V \times V$. An adjacency matrix $A \in \mathbb{N}^{nxn}$ can be constructed to represent the network such that entry $a_{ij} = 1$ if a node v_i is connected to v_j; otherwise, $a_{ij} = 0$. Let $C^{(s)}$ be the set of k ($k \geq 1$) ground-truth communities containing s. The problem of concern is to detect communities $C^{'(s)}$ for the seed $s$such that

$$\forall C_i^{(s)} \in C^{(s)} \Longrightarrow \exists C_j^{'(s)} \in C^{'(s)} | C_i^{(s)} \equiv C_j^{'(s)}$$

where $1 \leq i \leq k$ and $|C^{'(s)}| = k' \cong k$. Here, $|\cdot|$ indicates the number of communities.

To reduce the computational complexity, we usually search the communities in a sampled subgraph $G_s = (V_s, E_s) \subseteq G$. In the process of addressing the main problem, we consider several subproblems: local sampling and estimating the number of communities in G_s. The number of communities is important as it affects the quality of follow-up community detection and clustering algorithms. For example, Fig. 1 shows how k-means clustering finds accurate communities in the Zachary karate club network [48] for $k = 2$ (its ground-truth community number) than the clustering for $k = 3$.

B. Evaluation Measures

Various measures exist to evaluate community detection algorithms [49], [50]. A number of measures have been selected to evaluate the quality of solutions to the addressed problem.

The conductance of a detected community $C_j^{'(s)}$ is a fraction of its edges that point outside the community [49], [51]

$$\emptyset\left(C_j^{'(s)}\right) = \frac{\Sigma\left(i \in C_j^{'(s)}, j \notin C_j^{'(s)}\right)^{A_{ij}}}{\min\left(\mathrm{Vol}\left(C_j^{'(s)}\right), \left(\mathrm{Vol}(V) - \mathrm{Vol}\left(C_j^{'(s)}\right)\right)\right)} \quad (1)$$

where Vol$(\cdot)$ denotes the total degree of a set of nodes. The lower the value, the denser the inside connections with sparser outside connections for the community.

The recall of a ground-truth community $C_i^{(s)}$ indicates how well it has been detected

$$\mathrm{Rec}\left(C_i^{(s)}\right) = \max_{j=1\cdots k'} \frac{\left|C_i^{(s)} \cap C_j^{'(s)}\right|}{\left|C_i^{(s)}\right|}. \quad (2)$$

The precision of a detected community $C_j^{'(s)}$ indicates its relevance compared with the ground-truth communities

$$\mathrm{Prec}\left(C_j^{'(s)}\right) = \max_{i=1,\ldots,k} \frac{\left|C_i^{(s)} \cap C_j^{'(s)}\right|}{\left|C_j^{'(s)}\right|}. \quad (3)$$

A combined measure $F_\sigma (\sigma = 1, 2)$ can be computed, where F_1 balances the precision and recall, and F_2 focusses more on the recall [52]

$$F_\sigma = (1 + \sigma^2) \times \frac{\mathrm{Prec} \times \mathrm{Rec}}{(\sigma^2 \times \mathrm{Prec}) + \mathrm{Rec}}. \quad (4)$$

C. Graph Diffusion

Graph diffusion consists of spreading a node mass step by step throughout the graph, and this approach is very popular in local community detection [15], [17], [20], [21]. For a seed s, an initial vector $p^{(0)} \in \mathbb{R}^{\mathrm{n}}$ consisting of ones for the seed and zeros for the remaining nodes indicates the distribution of a random walker at the initial stage. Then, the random walker spreads information across the graph starting from the neighbors of s. At the first iteration, neighbors of s have $1/d(s)$ probability of being visited and zero for the remaining nodes, where $d(s)$ is the degree of node s. For the entire network, this information can be summarized into a transition matrix T, where $T_{ij} = A_{ij}/d(i)$ or $T = D^{-1}A$, where D is the diagonal degree matrix. The random walk step k consists of computing new probability vectors [15]

$$p^{(1)} = Tp^{(0)}, \ldots, p^{(l)} = Tp^{(l-1)}.$$

PPR [15], initially proposed by Brin and Page [53] for ranking webpages, uses this idea to rank nodes based on the seed with α probability of following any edge and $(1 - \alpha)$ probability to restart

$$p^{(l)} = \alpha * Tp^{(l-1)} + (1 - \alpha) * p^{(0)} \quad (5)$$

where $\alpha \in (0, 1)$. Andersen *et al.* [15] proposed a fast approximation of p that finds a probability vector p' using push operations. It uses two vectors p' and r (nonnegative), and each push operation copies some probability from r to p', while the remaining probability gets spread in r. The adapted pseudocode is provided in Algorithm 1, while the original pseudocode can be found in [15].

Another graph diffusion is HK that computes a vector of probabilities h for all nodes of a graph based on how they relate to the seed node. Compared with the PPR, HK is a function of a nonnegative temperature t that determines the walk length (similar to PPR's α parameter) and the initial heat distribution vector $h^{(0)}$ that is similar to $p^{(0)}$ used by PPR [54]: $h = H^{(t)}h^{(0)}$, where $H^{(t)}$ is the heat

Algorithm 1 Approximate PPR

INPUT: G, s, α, ε
Initialize:
  $r[s] = 1$ #start with the highest probability on seed s
  $Q = [s]$ #queue to store any node x if $r[x] \geq \varepsilon d(x)$
  $p' = \{\}$ #start with an empty sample
while (Q is not empty) **do**:
  $u = Q$.dequeue #first in first out
  if u not in p' **do**: $p'[u] = 0$ **end if**
  $p'[u] += (1-\alpha) * r[u]$ #copy $1-\alpha$ probability to p'
  $remaining = \alpha * r[u]$ #remaining probability
  $r[u] = remaining/2$ #keep a half of it in r and
  spread the remaining to the neighbors
  for v in neighbors of u **do**:
    if v not in r **do**: $r[v] = 0$ **end if**
    if $r[v] < \varepsilon d(v)$ and $r[v] + \frac{remaining}{2*d(u)} \geq \varepsilon d(v)$
      do: Q.append(v) **end if**
    $r[v] = r[v] + \frac{remaining}{2*d(u)}$
  if $r[u] \geq \varepsilon d(u)$ **do**: Q.append(u) **end if**
RETURN p'

operator, $H^{(t)} = e^{-tT}$. As the exponent of any matrix $e^A = \sum_{k=0}^{\infty}(A^k/k!)$, the HK diffusion becomes [54]

$$h = e^{-t}\left(\sum_{k=0}^{\infty}\frac{t^k}{k!}T^k\right)h^{(0)}. \tag{6}$$

Kloster and Gleich [17] proposed a fast approximation of HK (HK-Relax), which, at each step j, copies a probability from a residual r for a node v to p' if $r(v, j) \geq (e^t \varepsilon d(v))/(2N\psi_j(t))$ [17]. Initially, for a seed s of the seed set S, the residual $r(s, 0) = 1/|S|$.

D. NMF

NMF is a matrix factorization technique that reduces an input nonnegative matrix $A \in \mathbb{R}^{m\times n}$ to two nonnegative matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ such that $A \cong WH$ [55]. The number of components k should be specified. NMF consists of solving one of the following optimization problems [56]

$$\min_{W,H\geq 0} J_F(W, H) = \|A - WH\|_F^2 \tag{7}$$

$$\min_{W,H\geq 0} J_{KL}(W, H) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{[WH]_{ij}} - A_{ij} + [WH]_{ij}\right) \tag{8}$$

where J_F and J_{KL} denote the Frobenius-norm and Kullback–Leibler divergence cost functions. The optimization is done using the multiplicative update rules in (9) and (10) or (11) and (12), respectively [56]

$$H_{kj} \leftarrow H_{kj}\frac{(W^TA)_{kj}}{(W^TWH)_{kj}} \tag{9}$$

$$W_{ik} \leftarrow W_{ik}\frac{(AH^T)_{ik}}{(WHH^T)_{ik}} \tag{10}$$

$$H_{kj} \leftarrow H_{kj}\frac{\sum_i W_{ik}A_{ij}/(WH)_{ij}}{\sum_i W_{ik}} \tag{11}$$

$$W_{ik} \leftarrow W_{ik}\frac{\sum_j H_{kj}A_{ij}/(WH)_{ij}}{\sum_j H_{kj}}. \tag{12}$$

The detection of communities with NMF is done by normalizing H using (13) to generate the community membership probabilities of the nodes

$$H_{ij} = \frac{h_{ij}}{\sum_{x=1}^{k} h_{xj}}. \tag{13}$$

Hard clustering of nodes can be done based on the highest membership value [12], and overlapping communities can be detected by assigning nodes to multiple communities if more than one of its community membership values exceed a particular threshold [14], [19]. We use the sparseness generated by an SNMF to estimate the number of communities.

IV. S-MLC Approach

The proposed S-MLC approach includes three steps, as shown in Fig. 2.

A. Local Sampling

The objective of local sampling is to maximize the recall but also balance it with precision. A high recall indicates that most of the nodes that we want have been sampled, and a high precision indicates that we have a few irrelevant nodes that need to be discarded during the community detection phase. We use PPR or HK approximate algorithms [15], [17]. For PPR, we fix $\alpha = 0.99$ and vary $\varepsilon \in \{10^{-3}, 10^{-4}\}$. The smaller ε is, the larger the sample as the algorithm stops when the probability of every node v in V is less than $\varepsilon d(v)$. More details can be found in [15]. For HK, we vary $\varepsilon \in \{10^{-2}, 10^{-3}\}$ and $t \in \{80, 40\}$, respectively, and then compute $N = 2t * \log(1/\varepsilon)$ and psis (ψ) automatically, as discussed in [17].

We first construct a subgraph subG_s consisting of nodes that are associated with positive probabilities in p'. Then, we find its biconnected components using a popular algorithm proposed by Hopcroft and Tarjan [57] and sort them based on their size. A top (the largest) biconnected component containing s (G_s) is used as a sample. This means that every node in G_s can be reached from any node in G_s, and the removal of any node does not disconnect G_s. The main reason of using such a biconnected component is based on the core-periphery structure [51], [58] of network communities in which overlapping nodes tend to form a dense core with whiskers (less connected) around. The biconnected component returns almost all members of the communities of s.

B. Estimating the Number of Communities

Local sampling usually includes nodes that are not part of the communities of the seed s. Even when local sampling returns nodes that match those in the ground-truth communities of s, we still need to assign each node to its community as s may belong to multiple communities. We tackle this problem by first estimating the number of communities in the sampled subgraph based on a sparse coding that aims to find a few elements that can represent the entire population [59].

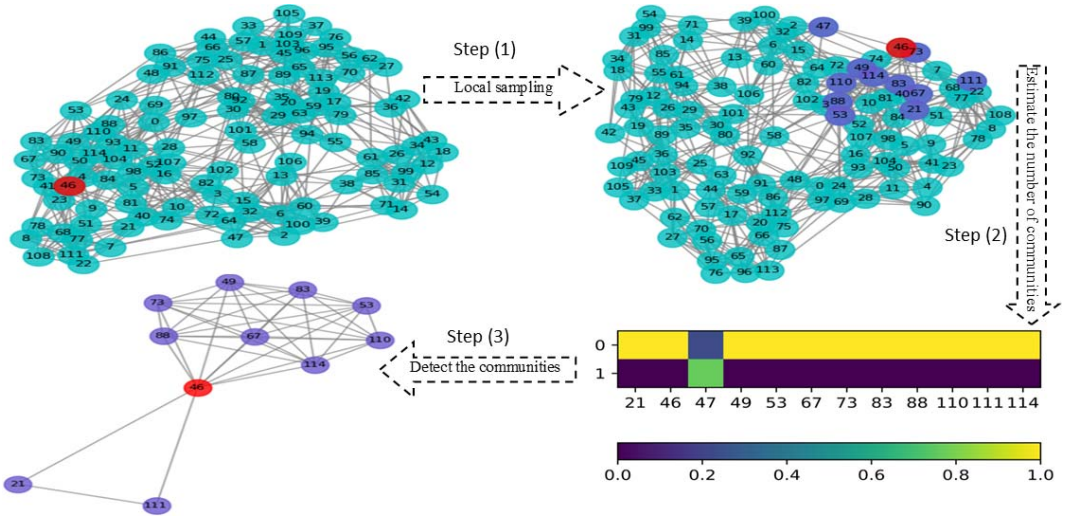


Fig. 2. Local community detection steps. We start with a seed s (node 46), sampling nodes around s, then estimate the number of communities in the sampled subgraph, and detect each of them based on the learned structure. The color bar indicates the community membership probabilities in H. Notice how node 47 is discarded as it forms its own community compared with other sampled nodes.

Given l input vectors $x_i\}_{i=1}^{l} \in \mathbb{R}^n$, sparse coding aims to find sparse codes $h_i\}_{i=1}^{l} \in \mathbb{R}^k$ and a dictionary $W \in \mathbb{R}^{nxk}$ (basis vectors) such that each x_i can be found using a linear combination of w^j: $x_i = \sum_{j=1}^{k} h_i^j w^j = Wh_i$ [59]. Most of the coefficients h_i^j are zeros or close to zero with few values (representative) far from zero, resulting in a sparse vector h_i. By using NMF, the sparsity can be enforced by adding penalties on either W, H, or both as follows:

$$J(W, H) = J_z(A, WH) + \omega\Phi(W) + \beta\psi(H) \tag{14}$$

where J_z is a cost function in (7) or (8); ω and β are sparsity parameters; and Φ and ψ are sparsity functions. L_0-norm would be the most appropriate, but it is difficult to optimize due to the nondifferentiability. L_1-norm is the most popular sparsity function and has been used in many applications with good results [30], [31].

The sparsity can be imposed on rows of H and/or columns of W. For community detection, the sparsity on rows of H would activate a few nodes in each community (the nodes belonging to the community), which is only useful when we know the number of communities.

In case we do not know the number of communities, the sparsity on columns of H would activate a few communities (components) for each node. In other words, each node belongs to a few communities as opposed to the sparsity on rows, which assumes that each community consists of a few nodes.

We use the SNMF on columns proposed by Kim and Park [31] to iteratively decompose the adjacency matrix of G_s and estimate the number of communities based on the sparseness of H

$$\min_{W,H\geq 0} \left\{ \|\text{A-WH}\|_F^2 + \beta \sum_{j=1}^{n} \|h_j\|_1^2 \right\}. \tag{15}$$

The optimization of (15) is done by alternating (16) and (17) [31]

$$\min_{W\geq 0} \|H^T W^T - A^T\|_F^2 \tag{16}$$

$$\min_{H\geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\beta}\mathbf{1} \end{pmatrix} - \begin{pmatrix} W \\ \mathbf{0} \end{pmatrix} \right\|_F^2 \tag{17}$$

where $\mathbf{1}$ is an all one-row vector of dimension $1 \times k$ and $\mathbf{0}$ is an all zero-row vector of dimension $1 \times n$.

SNMF generates partitions that overlap in which every node is associated with a probability value of belonging to each community, and this is useful in detecting overlapping communities of the seed. In addition, the sparsity property, when applied to columns of coefficients, activates a few components for each node and demonstrates the total number of communities in the network (the highest number of components that have been activated).

To control the degree of sparsity enforced, Hoyer [30] proposed a measure for the sparseness of a vector

$$\text{sparseness}(h) = \frac{\sqrt{n} - \left(\sum |h_i|\right)/\sqrt{\sum h_i^2}}{\sqrt{n} - 1} \tag{18}$$

where n is the dimensionality of vector h. The sparseness of a matrix can be computed by averaging the sparsity of its vectors. In the context of community detection, (18) can be interpreted as follows: if all community membership values of a node represented by a vector $h \in \mathbb{R}^{k'}$ (for k' communities) are equal, the value of the sparseness(h) becomes zero, which means that the node is equally distributed across all communities. If only one value is nonzero, the sparseness(h) becomes 1 (sparsest and hard clustering), and the node belongs to only one community; otherwise, the sparseness(h) takes a fuzzy value indicating the degree of sparsity for a node belonging to several communities.

Iterating from $k' = 2$ to $k' = k_{\max}$, where $k_{\max}$ is the maximum number of possible communities; we run SNMF

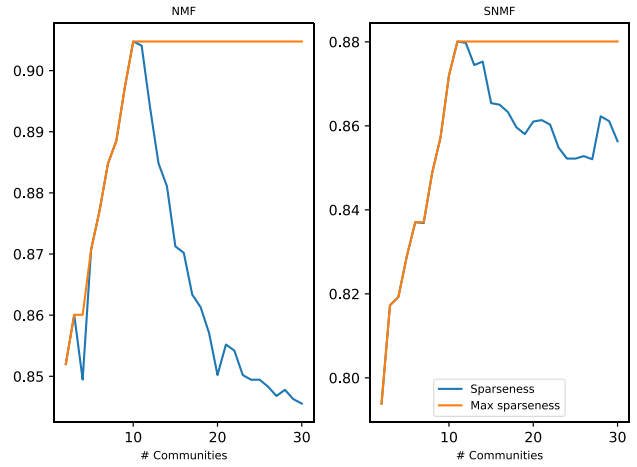


Fig. 3. Sparseness versus the number of communities using NMF in (7) and SNMF in (15) on the American college football network. The sparseness curve indicates the average sparseness of H for various NMF/SNMF iterations, while the max sparseness curve plots the maximum sparseness value from the initial iteration up to the current iteration. When the maximum sparseness gets close to 1, it becomes stable. At that point, NMF/SNMF iterations reach or get close to the representative number of components that we use as the number of communities. The SNMF sparseness is controlled, while NMF sparseness decreases dramatically when NMF iterations surpass the number of communities in the network.

in (15) and compute the average sparseness over all columns of H for each iteration. We keep track of the highest average sparseness using (18) and stop iterating when this average sparseness is nonincreasing for ten iterations (to make sure that there is no other number of components that yield a higher sparseness). This parameter is determined by experiments but can be changed (increase for more confidence to check if there might be a higher value of sparseness or reduce to improve the speed). This indicates that we can use the number of components corresponding to the maximum sparseness of H to effectively represent the entire network. For this reason, we use them as the estimated k', as shown in Fig. 3, for the American college football network [60].

For the entire network of the American college football, 11 communities are correctly estimated when our approach is used with the SNMF in (15) and ten communities when the same approach is used with the NMF in (7). This emphasizes the superiority of SNMF over NMF. In fact, NMF solutions are not unique [30], [61]. Any matrix X such that $WX \geq 0$ and $X^{-1}H \geq 0$ also provides a solution that causes unstable results that affect the community detection.

Our method of estimating the number of communities has never been used in the literature and is superior to our previous MLC approach. We keep the sparseness parameter very small as high values increase the approximation error and the corresponding components may not represent the network. Algorithm 3 would be slow when it is used on a large network due to SNMF decomposition. This explains why the initial sampling in Algorithm 2 is required to speed up Algorithm 3.

C. Detecting Multiple Local Communities

This is the final step of S-MLC for detecting multiple local communities for a specific seed. Given a community membership vector $h_i \in \mathbb{R}^{k'}$ from H for a node v_i, we want to assign v_i to its communities. Let $\theta = 1/|k'|$ be the average community membership probability for every node. A node that belongs to all k' communities would be represented by a value that is equivalent to $1/|k'|$ in each of the k' communities. We assign a node v_i to community j if $h_i^{(j)} \geq \theta$. This operation can be performed faster for all nodes using matrix operations, as shown in Algorithm 4. A node with a very high membership value (close to 1) in one community has less chance of being in any other community. A node whose community membership probabilities are equally distributed (equalsto θ) is assigned to all communities, and overlaps appear easily.

Note that a user can force θ to be a specific value depending on the type of community he (she) wants to detect. For example, if a user wants to detect communities where a node is allowed to be in no more than two communities, set $\theta = 0.5$. If one wants nodes to be in no more than three communities, then set $\theta = 0.33$; and if one wants communities whose

Algorithm 2 Local Sampling

1. **INPUT**: Graph $G = (V, E)$ and seed $s \in V$
2. **Set** PPR approximation parameters: $params$
3. $p' =$ **ApproximatePPR**$(G, s, params)$
4. $V_s = [v \text{for} v \text{in} V \text{if} p'[v] > 0]$
5. $subG_s =$ **subgraph**(G, V_s)
6. $G_s =$ biggest biconnected component in $subG_s$ containing s
7. **RETURN** G_s

Algorithm 3 Estimating the Number of Communities

1. **INPUT**: Graph$G_s = (V_s, E_s)$
2. $A =$ adjacency matrix of G_s in $n_s \times n_s$
3. **INITIALIZE:** $k_{max} = n_s/4$, $\beta = 1e-4$
   $x = 0.8$ #start with a good sparseness
   $counter = 1$ #counts of non-increasing sparseness
   $k' = 1$ #initial number of communities
   $finalH =$ ones$(1, n_s)$ #start with a single community
4. **for** $temp_{k'}$inrange(2,k_{max}) **do**:
5. $W, H = SNMF(A, temp_{k'})$ #using Eq. (15)
6. $avg_s = 0$; $x_h = 0$ #for computing sparseness of H
7. **for** $j$inrange(0,$n_s - 1$) **do:**#each node representation
8. $x_h +=$ sparseness$\left(h^{(j)}\right)$ #using Eq. (18)
9. **end for**
10. $avg_s = {}^{x_h}/_{temp_{k'}}$ #compute the sparseness of H
11. **if** $avg_s \leq x$ **do:** #sparseness not improving
12. $counter += 1$
13. **else:**#the sparseness has improved
14. $k' = temp_{k'}$; $finalH =$ Normalize(H) #Eq. (13)
15. $x = avg_s$; $counter = 1$ #reset the counter
16. **end if**
17. **if** $counter == 10$ **do:**#sparseness still not improving
18. break
19. **end if**
20. **end for**
21. **RETURN** k', $finalH$

Algorithm 4 Overall S-MLC Algorithm

1. **INPUT**: Graph $G = (V, E)$ and seed s
2. coms = [] #create an empty list of communities of s
3. G_s = Algorithm2(G, s)
4. k'', H = Algorithm3(G_s)
5. $\theta = 1/|k'|$
6. $H[H \geq \theta] = 1$
7. **for** i inrange(0,$k' - 1$) **do**
8. com = [$v$innodeswhoseindexin$H(i, :) = 1$]
9. **append** com to $coms$ if $s$is in com
10. **end for**
11. **RETURN** $coms$

members do not overlap with any other communities, then set $\theta = 1$ or close to 1.

D. Complexity

Approximate PageRank's time complexity depends on the graph size n and the parameters used: $O((\log n)/(\epsilon\alpha))$ [15]. Extracting a biconnected component is affected by the maximum value between the number of nodes n or the number of edges m of the graph: $\max(n, m)$ for both the time and space required [57]. Therefore, the complexity of local sampling is determined by the original graph size and the parameters used, ϵ and α. Especially, when α decreases, the more time it takes for sampling. For community number estimation, the complexity depends on the number of nodes n_s in the sampled subgraph as the worst case of decomposing its adjacency matrix is $n_s/4$ times. This happens when the subgraph consists of very small clusters ($\leq$4 nodes).

Normally, a network may not consist of more than $n_s/4$ clusters as most clusters of real-world networks have sizes greater than four nodes. The community detection phase depends on the number of clusters obtained.

The overall complexity of our algorithm depends on ϵ and α that determine the size of G_s, and the rest is $O(n_s)$.

V. Experiments

All the experiments are done on a laptop with a processor: i5 @ 1.70 GHz, RAM: 8 GB, and a 64-bit Windows operating system. All the algorithms are implemented in Python.

We first evaluate the local sampling, followed by community number estimation that we compare with the MLC approach and the modularity maximization, and then, we compare the proposed local community detection approaches with MLC [19] and M-LOSP [21]. As there are few algorithms for multiple-local-community detection, we also adapt DEMON [13], one of the popular global overlapping community detection algorithms, to run on the sampled subgraph to find approximate local communities, denoted as L-DEMON. For networks with ground-truth communities, we compare the detected communities with the ground-truth communities using F_1 and F_2 scores. A high F_1 score is important because it focuses on both good precision and good recall. This indicates the algorithm's ability to discard irrelevant nodes such as innocent people in criminal detection and find all relevant nodes (criminal people). F_2 score is high when the recall is relatively high as less attention is given to the precision. In criminal detection, if all criminals are found, F_2 would be high even if some innocents are included as criminals.

TABLE I
Real-World Benchmarks

Networks	n	m	k	average degree
Zachary karate club	34	78	2	4.59
Dolphin	62	159	4	5.13
American college football	115	613	10	10.66
Amazon	334,863	925,872	75,149	5.53
DBLP	317,080	1,049,866	13,477	6.62
ego-Facebook	4,039	88,234	-	43.69
gemsec Politician	5908	41,729	-	14.12
Gemsec TvShow	3,892	17,262	-	8.87
musae-Facebook	22,470	171,002	-	15.22

n: number of vertices; m: number of edges;
k: number of ground-truth communities;
-: no ground-truth communities.

NMI [8], [62] is another measure often used to evaluate the community detection quality, but it is suitable for evaluating global communities and not local communities. This is because any two community assignments (ground-truth and detected) should have the same number of nodes n in order to be compared using NMI (requires n). In local community detection, there is a detected portion and the rest of the network, and some nodes in the ground-truth communities may not appear in the detected communities, and vice versa. For networks without ground-truth communities, we compute the conductance to evaluate the quality of the detected communities.

A. Data Sets

1) Artificial Networks: We use the LFR benchmark networks [63] that simulate the characteristics of real-world networks as opposed to the GN benchmark [64], which assumes that all nodes have the same degree and all communities are of the same size. Eight unweighted, undirected LFR networks are generated by modifying the mixing parameter μ, minimum community size $C_{\min}$, maximum community size $C_{\max}$, number of overlapping nodes *on*, and number of memberships of the overlapping nodes *om*. Each network consists of 1000 nodes (n_s) with different number of edges (m_s). We choose the average degree $d = 5$ and the highest degree $d_{\max} = 15$.

2) Real-World Networks: We use a combination of small and large networks often used to evaluate community detection algorithms. The statistics of the networks are shown in Table I.

The Zachary karate club is a social network of friendships among 34 members of a karate club in a U.S. university in the 1970s [48]. A conflict between the club administrator and the instructor over the price of a karate lesson led the network to be split into two groups.

Dolphin is a network of 62 dolphins observed between 1994 and 2001 in Doubtful Sound, New Zealand. An edge connects two dolphins that were seen together "more often than the expected chance." The network consists of two main

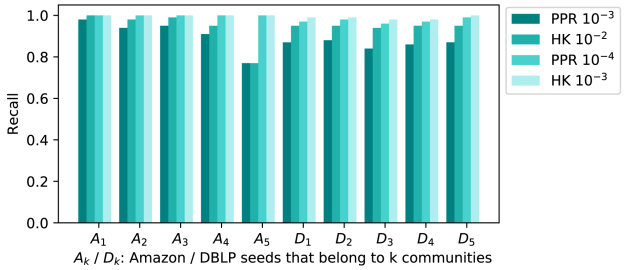


Fig. 4. PPR and HK sampling recall on Amazon and DBLP. HK returns all nodes in the ground-truth communities of the seed for both values of ε used. For PPR, the recall is slightly lower compared with HK. For both algorithms, as we decrease ε, we get a higher recall.

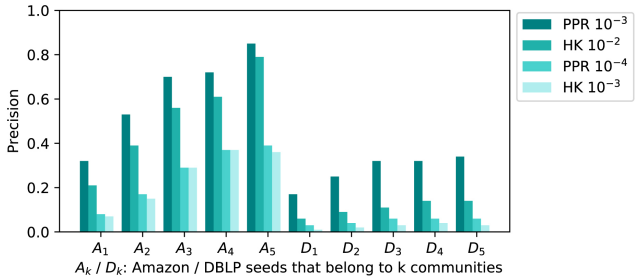


Fig. 5. PPR and HK sampling precision on Amazon and DBLP. PPR has higher precision than HK on both data sets for all 1000 seeds used in total. PPR returns less irrelevant nodes during sampling compared with HK. On Amazon, seeds that belong to five communities have a high precision; as most of their communities consist of a high overlap and sometimes, they may look like a single community.

communities although one community can be split into three subcommunities [65].

American college football is a network of games between Division I colleges in 2000 [64]. Colleges are grouped into conferences, and each conference is a ground-truth community. Colleges (teams) are nodes, and edges denote games between the teams.

Amazon is a network with ground-truth communities consisting of products that are often purchased together [1]. The DBLP computer science bibliography is a coauthorship network with ground-truth communities consisting of authors who published in the same journal or conference [1]. Both Amazon and DBLP have a minimum ground-truth community size of three nodes, and all their ground-truth communities are defined based on the metadata.

Ego-Facebook is a network of friendships on Facebook collected from 4039 users with a mobile app [1]. Gemsec-Politician, Gemsec-TvShow, and Musae-Facebook are networks representing verified Facebook pages of different categories. Nodes represent pages, while edges indicate mutual likes between pages [1]. These networks do not have ground-truth communities.

B. Local Sampling Results

We evaluate the quality of local sampling on large real-world networks only as community detection on small networks can be done without local sampling. Figs. 4 and 5 show the results obtained on Amazon and DBLP data sets using PPR approximation with $\alpha = 0.99$ and $\varepsilon \in \{10^{-3}, 10^{-4}\}$

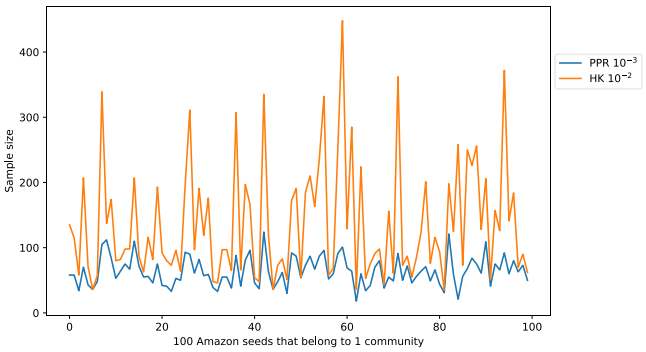


Fig. 6. Overview of sample sizes of HK 10^{-2} and PPR 10^{-3}.

and HK approximation with $\varepsilon \in \{10^{-3}, 10^{-4}\}$ and $t \in \{80, 40\}$, respectively. Each category (A_k or D_k) indicates the average results computed over 100 seeds.

Overall, the precision is higher on Amazon than DBLP, and PPR has higher precision than HK, while HK has higher recall than PPR. On both PPR and HK, the lower the ε, the lower the resultant precision, and the higher the resultant recall. The sampling results indicate that if we use HK sampling, most nodes we need will be sampled along with many irrelevant nodes that will need to be discarded during the community detection phase. If we use PPR, we may not sample all nodes we need, but we get fewer irrelevant nodes that need to be discarded during the community detection phase. As the precision gets lower, the community detection becomes much harder as a result of having many nodes to discard. On DBLP, we have a very high recall compared with the precision, which predicts harder community detection.

The sampling results differ based on the characteristics of the networks. For example, on Amazon, books can belong to the literature and fiction as one community. From these books, there are books for teens and young adults that can form a second community, while some books may be related to history (historical fiction) and form the third community. In this case, many books overlap in various communities, and this is the same for other products. As a result, the overlapping portion is denser compared with DBLP, and when you take an Amazon seed, most of its community members are easily sampled with an improved precision compared with DBLP. This explains the reason why the sampling precision is higher for seeds belonging to many communities than for seeds belonging to a single community, as shown in Fig. 5.

Figs. 6 and 7 show a comparison of PPR and HK. HK 10^{-2} samples more nodes compared with PPR 10^{-3}, and HK 10^{-3} sample a larger number of nodes compared with both PPR 10^{-3} and PPR 10^{-4}.

C. Results on Estimating the Number of Communities

We use small real-world networks (see Table II) whose number of communities is known and eight LFR networks summarized in Table III. Eight LFR networks are used to estimate their number of communities. As these networks are small, we set $G_s = G$ so that we can estimate the number of communities in the entire network.

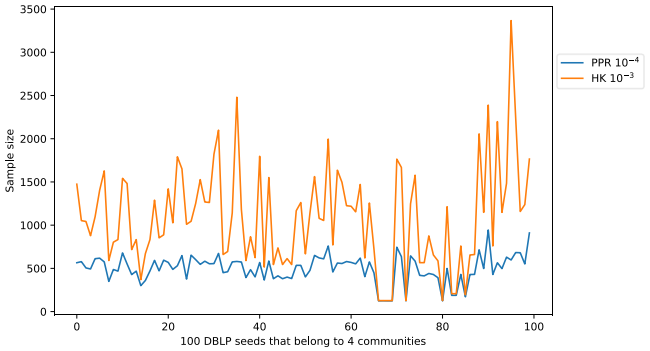


Fig. 7. Overview of sample sizes of HK 10^{-3} and PPR 10^{-4}.

TABLE II
NUMBER OF COMMUNITIES OF SMALL REAL-WORLD NETWORKS

Network	k	k'	k'_{MLC}	k_{mod}
Zachary karate Club	2	**2**	**2**	**2**
Dolphins	2	**2**	4	3
American college football	11	**11**	5	10

k: number of ground-truth communities;
k': number of communities estimated by the sparseness approach;
k'_{MLC}: number of communities detected by modularity maximization.
k_{mod}: number of communities detected by modularity maximization.

TABLE III
NUMBER OF COMMUNITIES ON ARTIFICIAL NETWORKS

Network	LFR parameters					n	m	k	k'	k'_{MLC}	k_{mod}
	μ	C_{min}	C_{max}	on	om						
G_1	0	10	30	0	0	1000	2778	57	61	41	53
G_2	0	100	200	0	0	1000	2813	6	**6**	**6**	5
G_3	0	3	20	20	2	1000	2667	112	115	40	**112**
G_4	0	20	200	20	2	1000	2769	9	**9**	18	8
G_5	0.1	20	200	20	2	1000	2745	13	**13**	**13**	12
G_6	0.1	20	200	20	3	1000	2721	10	**10**	12	**10**
G_7	0.1	20	200	20	4	1000	2756	11	**11**	14	10
G_8	0.1	20	200	20	5	1000	2837	18	17	13	16

μ: mixing parameter; C_{min}: minimum community size;
C_{max}:maximum community size; on: number of overlapping vertices;
om: number of memberships of the overlapping vertices.

Fig. 8 shows the correlation between the sparseness and the number of communities. The nonincreasing sparseness correlates with the ground-truth number of communities on graphs that consist of a few communities (G_2, G_4, G_5, G_6, and G_7) compared with graphs with many communities (G_1 and G_3). The sparseness-based approach accurately estimates the number of communities in G_2, G_4, G_5, G_6, and G_7. The modularity approach accurately estimates all the 112 communities in G_3 and G_6. In practice, a seed belongs to less than five communities, and our algorithm is more accurate than the modularity-based approach when estimating the number of communities in such cases.

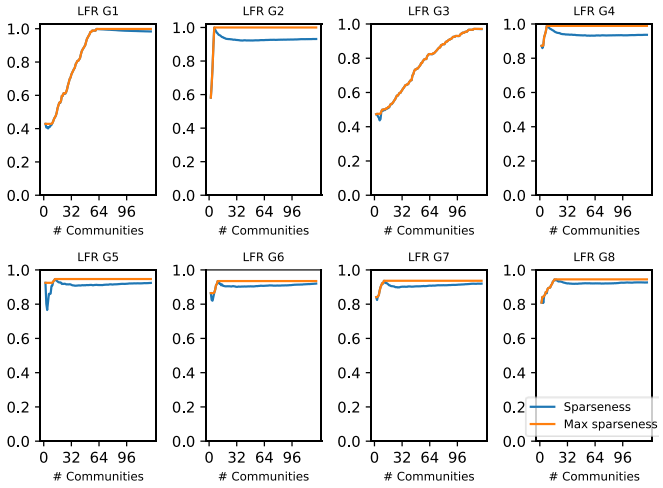


Fig. 8. Sparseness versus the number of communities in LFR networks. The nonincreasing sparseness correlates with the ground-truth number of communities for most networks. The sparseness becomes stable when greater than 0.8.

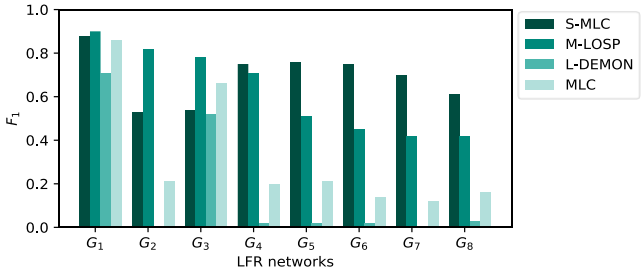


Fig. 9. F_1 results on LFR networks. S-MLC outperforms MLC, M-LOSP, and L-DEMON on most networks except on G_2 and G_3. G_3 consists of very small communities, while G_2 consists of large communities with a good community structure. DEMON's approach does not guarantee that each node will be assigned to a community. Thus, most detected communities do not include the seed, and the averages of F_1 scores are very low.

Overall, the sparseness approach clearly outperforms both the MLC and the modularity approaches as it accurately estimated five out of eight instances on artificial networks and all instances of small real-world networks.

D. Local Community Detection Results

We compare S-MLC with M-LOSP, MLC, and L-DEMON using F_1 and F_2 scores and execution times in seconds. S-MLC uses PPR with $\varepsilon = 10^{-3}$ for local sampling due to the improved precision compared with HK (see Fig. 5). Figs. 9 and 10 show the F_1 and F_2 results on artificial networks, while Fig. 11 shows the algorithms' execution time. Figs. 12 and 13 show the results on small real-world networks, while Figs. 14 and 15 show the results on large real-world networks. Fig. 16 provides insights into the quality of the communities detected by S-MLC based on the conductance measure (1). Fig. 17 compares the conductance results of different algorithms on data sets without ground-truth communities.

For the artificial networks summarized in Table III, we select 100 seeds that belong to one community for each of $\{G_1, G_2\}$. Then, for each of $\{G_3, G_4, G_5\}$, we select all

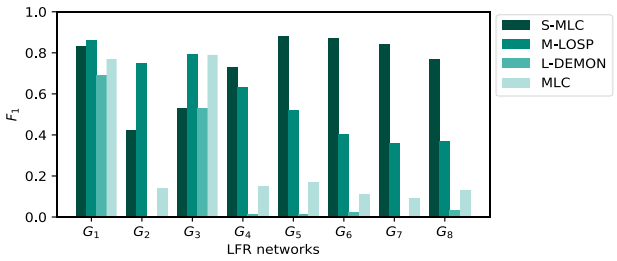


Fig. 10. F_2 results on LFR networks. S-MLC outperforms the other three algorithms of most networks except on G_2 and G_3.

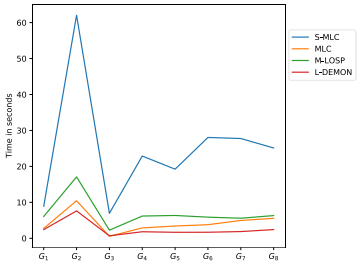


Fig. 11. Execution time on LFR networks. Most algorithms are fast on networks with small communities $\{G_1, G_3\}$ compared with networks that consist of large communities $\{G_2, G_4, G_5, G_6, G_7, G_8\}$. Overall, each algorithm detects communities of a given seed within one second as the results shown are averaged over 100 seeds for G_1 and G_2 and 20 seeds for the remaining networks as indicated by their number of overlapping nodes (on) in Table III.

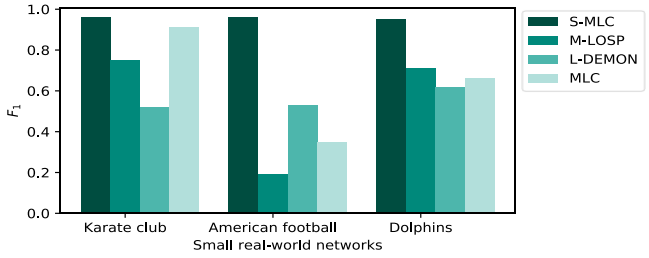


Fig. 12. F_1 results on small real-world networks. S-MLC outperforms the other three baselines.

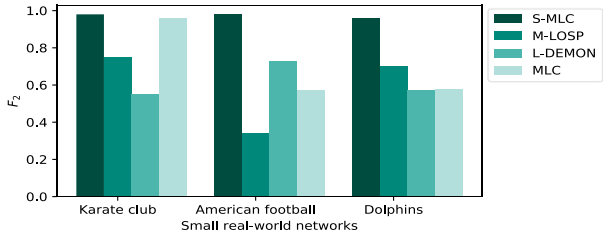


Fig. 13. F_2 results on small real-world networks. S-MLC outperforms the other three baselines.

seeds that belong to two communities, and for $\{G_6, G_7, G_8\}$, we select all seeds that belong to three, four, and five communities, respectively.

In Figs. 9 and 10, all the algorithms easily detect small single communities in G_1. M-LOSP and S-MLC outperform other algorithms in detecting large single communities in G_2. M-LOSP and MLC outperform the rest of the algorithms in finding small overlapping communities in G_3, while S-MLC dominates other algorithms in detecting large overlapping communities in $\{G_4, G_5, G_6, G_7, G_8\}$.

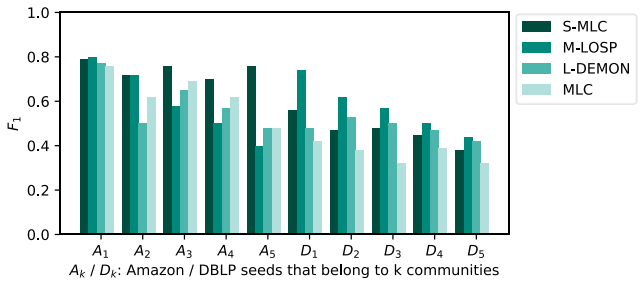


Fig. 14. F_1 results on large networks. In general, S-MLC outperforms M-LOSP, L-DEMON, and MLC on Amazon, especially for A_5 that contains more local communities. On DBLP, S-MLC outperforms MLC with a performance close to L-DEMON's. On DBLP, M-LOSP is the best, followed by S-MLC, L-DEMON, and MLC.

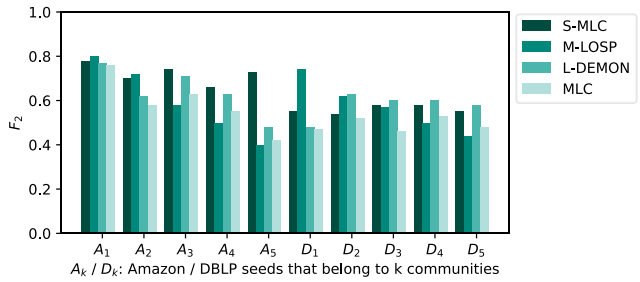


Fig. 15. F_2 results on large networks. Overall, F_2 is higher than F_1 results. In general, S-MLC outperforms M-LOSP, L-DEMON, and MLC on Amazon. On DBLP, L-DEMON has a higher F_2 score compared with the other three methods.

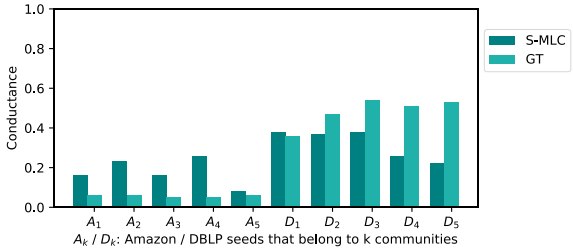


Fig. 16. Conductance of the detected communities compared with the conductance of the ground-truth communities. GT denotes the ground-truth communities.

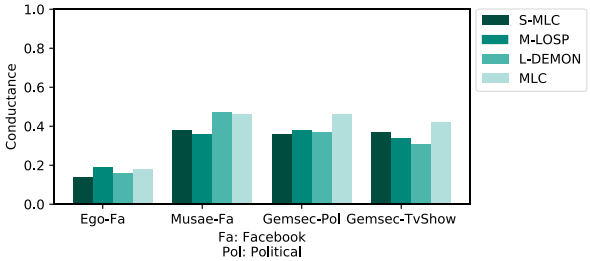


Fig. 17. Conductance of the detected communities for different algorithms on various Facebook data sets. The conductance is low on the ego-Facebook network (Ego-Fa) that represents a good community structure, while the conductance is high on data sets that represent mutual likes between pages.

For some seeds, L-DEMON gets very low scores. For such seeds, most returned communities do not include the seed, and in such cases, the corresponding F_1 and F_2 scores become zero. For each node v in the sampled subgraph, L-DEMON finds its ego network (direct neighbors of v) and then removes v from the ego network and applies the label propagation algorithm [13] on the remaining nodes to find communities. Label propagation assigns a node v to a community C if the maximum number of neighbors of v belong to C. When v is not densely connected, the label propagation operates on a very small subgraph and returns very small communities often consisting of two or three nodes each. Such communities are discarded by L-DEMON when they are no greater than a particular threshold (default threshold is 3). Larger communities are merged with or appended to previously detected communities. To sum up, although L-DEMON finds communities in an entire subgraph given as the input, there is no guarantee that all nodes will appear in the detected communities.

Regarding efficiency, L-DEMON is the fastest as it has been adapted for local sampling. The original DEMON would be much slower than the reported results of L-DEMON as DEMON would run on the entire network without local sampling. MLC is faster than S-MLC as enforcing the sparsity on S-MLC adds some computation but yields improved accuracy, as shown in Fig. 9 especially on $G_4, G_5, G_6, G_7,$ and G_8.

On small real-world networks, S-MLC also outperforms the rest of the algorithms (see Figs. 12 and 13). As we did not use sampling on such networks (for all algorithms), the results indicate the superiority of the community estimation approach and community detection approach of S-MLC over the ones used in MLC as indicated by F_1 and F_2 scores.

On large real-world networks, S-MLC performs well on Amazon, and M-LOSP performs well on DBLP, as shown in Figs. 14 and 15. Overall, S-MLC performs well when a good sample is available. For instance, a very good performance on seeds for five communities (A_5) is based on a high sampling precision. S-MLC uses PPR-10^{-3} for local sampling. This is the case of small networks and Amazon. On Amazon, in general, S-MLC outperforms M-LOSP, followed by L-DEMON and MLC. On DBLP, in general, M-LOSP is the best, and S-MLC and L-DEMON are competitive, followed by MLC. We see that when there is a sample with many irrelevant nodes that need to be discarded, such as DBLP (see Fig. 5), S-MLC is still able to discard most of the irrelevant nodes and uncover more than 50% of the relevant nodes, as shown by F_1 and F_2 scores.

In summary, S-MLC shows favorable accuracy compared with the other three baselines. We further see that all communities detected by S-MLC have a good conductance of less than 0.5 on average (see Fig. 16). A small conductance indicates a good community structure, and the worst community structure has a conductance of 1. On Amazon, the quality of the detected communities is lower than the quality of the ground-truth communities based on their conductance. Most of the time, this is caused by detected communities larger than the ground-truth communities where additional nodes reduce the quality. On DBLP, it is the opposite. The ground-truth communities are very large with a low community quality (high conductance), while the detected communities are smaller and more compact as indicated by lower conductance compared with the ground-truth communities. This kind of interpretation is specifically useful when we do not know the ground-truth communities, as shown in Fig. 17. This means that users in the ego-Facebook network tend to form compact groups of friends where a user may be a friend of almost all members in the same community. For Facebook pages, one page may like many pages that do not like each other, and their community structure is not well structured as the one formed by Facebook users.

VI. Conclusion

As opposed to the existing local community detection methods that focus on detecting a single community supervised by some exemplary seeds, we assume that a seed may belong to multiple communities and propose S-MLC, a new method of detecting multiple local communities. The method is based on three steps that can be used independently by other community detection algorithms. For instance, our approach of estimating the number of communities in a network can be used in other algorithms that require the number of communities as prerequisites.

We evaluated S-MLC using real-world and artificial networks, and experiments showed favorable accuracy on S-MLC, which, in general, outperforms the three state-of-the-art baselines on artificial networks, small real networks, and large real network of Amazon, only having exceptions on a large real network of DBLP. The evaluation also showed that community detection results depend highly on the sampling quality. A good sampling generates all nodes of the target communities with no or few irrelevant nodes to be discarded during community detection. The sampling also has an impact on the complexity of our algorithm as subsequent community detection tasks depend on the size (number of nodes) of the subgraph G_s. PPR was selected over HK for sampling as PPR generates samples with higher precision compared with HK.

For networks without ground-truth communities, the smaller the conductance of the detected communities, the better the obtained communities and S-MLC generated communities with an average conductance of less than 0.5.

References

[1] J. Leskovec and A. Krevl. (2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. Accessed: Jan. 6, 2020. [Online]. Available: http://snap.stanford.edu/data/

[2] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 1067–1077.

[3] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1025–1035.

[4] Y. Xiao, X. Li, H. Wang, M. Xu, and Y. Liu, "3-HBP: A three-level hidden Bayesian link prediction model in social networks," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 430–443, Jun. 2018.

[5] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 4, pp. 1–9, Oct. 2009.

[6] K. Taha, "Disjoint community detection in networks based on the relative association of members," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 493–507, Jun. 2018.

[7] X. Zhang, C. Wang, Y. Su, L. Pan, and H.-F. Zhang, "A fast overlapping community detection algorithm based on weak cliques for large-scale networks," *IEEE Trans. Comput. Social Syst.*, vol. 4, no. 4, pp. 218–230, Dec. 2017.

[8] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–43, Nov. 2016.

[9] 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.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[11] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, Apr. 2011, Art. no. e18961.

[12] N. Binesh and M. Rezghi, "Fuzzy clustering in community detection based on nonnegative matrix factorization with two novel evaluation criteria," *Appl. Soft Comput.*, vol. 69, pp. 689–703, Aug. 2018.

[13] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "DEMON: A local-first discovery method for overlapping communities," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 615–623.

[14] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2013, pp. 587–596.

[15] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using PageRank vectors," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2006, pp. 475–486.

[16] I. M. Kloumann and J. M. Kleinberg, "Community membership identification from small seed sets," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 1366–1375.

[17] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *Proc. KDD*, 2014, pp. 1386–1395.

[18] P. Shi, K. He, D. Bindel, and J. E. Hopcroft, "Local Lanczos spectral approximation for community detection," in *Proc. ECML PKDD*, 2017, pp. 651–667.

[19] D. Kamuhanda and K. He, "A nonnegative matrix factorization approach for multiple local community detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, no. 1, Aug. 2018, pp. 642–649.

[20] Y. Bian, Y. Yan, W. Cheng, W. Wang, D. Luo, and X. Zhang, "On multi-query local community detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 9–18.

[21] K. He, Y. Sun, D. Bindel, J. Hopcroft, and Y. Li, "Detecting overlapping communities from local spectral subspaces," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 769–774.

[22] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 1963–1976, Jul. 2018.

[23] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. IEEE 13th Int. Conf. Data Mining (ICDM)*, Dec. 2013, pp. 1151–1156.

[24] X. Teng, J. Liu, and M. Li, "Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, early access, Aug. 27, 2020, doi: 10.1109/TCYB.2019.2931983.

[25] P. Moradi and M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection," *Knowl.-Based Syst.*, vol. 84, pp. 144–161, Aug. 2015.

[26] Y. Li, K. He, D. Bindel, and J. E. Hopcroft, "Uncovering the small community structure in large networks: A local spectral approach," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 658–668.

[27] K. He, P. Shi, D. Bindel, and J. E. Hopcroft, "Krylov subspace approximation for local community detection in large networks," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 5, Oct. 2019, Art. no. 52.

[28] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Sci. Rep.*, vol. 1, no. 1, p. 196, Dec. 2011.

[29] F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, "TCARS: Time-and community-aware recommendation system," *Future Gener. Comput. Syst.*, vol. 78, pp. 419–429, Jan. 2018.

[30] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Nov. 2004.

[31] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, Jun. 2007.

[32] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bulletin.*, vol. 40, no. 3, pp. 52–74, Sep. 2017.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[34] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2016, pp. 855–864.

[35] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 701–710.

[36] A. Clauset, "Finding local community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, Aug. 2005, Art. no. 026132.

[37] J. Chen, O. Zaïane, and R. Goebel, "Local community identification in social networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Jul. 2009, pp. 237–242.

[38] D. F. Gleich and R. A. Rossi, "A dynamical system for PageRank with time-dependent teleportation," *Internet Math.*, vol. 10, nos. 1–2, pp. 188–217, Apr. 2014.

[39] A. Hollocou, T. Bonald, and M. Lelarge, "Multiple local community detection," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 3, pp. 76–83, Mar. 2018.

[40] S. Sarkar and A. Dong, "Community detection in graphs using singular value decomposition," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 4, pp. 1–16, Apr. 2011.

[41] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proc. 25th Int. J. Conf. Artif. Intell.*, 2016, pp. 2252–2258.

[42] S. Rahimi, A. Abdollahpouri, and P. Moradi, "A multi-objective particle swarm optimization algorithm for community detection in complex networks," *Swarm Evol. Comput.*, vol. 39, pp. 297–309, Apr. 2018.

[43] A. Mahmood and M. Small, "Subspace based network community detection using sparse linear coding," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 801–812, Mar. 2016.

[44] 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.

[45] M. Mohammadi, P. Moradi, and M. Jalili, "AN NMF-based community detection method regularized with local and global information," in *Proc. Electr. Eng. (ICEE), Iranian Conf.*, May 2018, pp. 1687–1692.

[46] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[47] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[48] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[49] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proc. IEEE 12th Int. Conf. Data Mining (ICDM)*, Dec. 2012, pp. 745–754.

[50] X. Wang and J. Liu, "A comparative study of the measures for evaluating community structure in bipartite networks," *Inf. Sci.*, vols. 448–449, pp. 249–262, Jun. 2018.

[51] 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.

[52] C. D. Manning, P. Raghavan, and H. Schutze, *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[53] S. Brin and L. Page, "The PageRank citation ranking: Bringing order to the Web," *BMC Syst. Biol.*, vol. 4, no. 66, p. S13, Nov. 1999.

[54] F. Chung, "The heat kernel as the PageRank of a graph," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 50, pp. 19735–19740, 2007.

[55] 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.

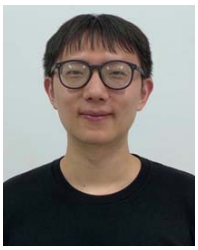
[56] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, no. 1, 2001, pp. 556–562.

[57] J. Hopcroft and R. Tarjan, "Algorithm 447: Efficient algorithms for graph manipulation," *Commun. ACM*, vol. 16, no. 6, pp. 372–378, Jun. 1973.

[58] J. Yang and J. Leskovec, "Overlapping communities explain core–periphery organization of networks," *Proc. IEEE*, vol. 102, no. 12, pp. 1892–1902, Dec. 2014.
[59] D. J. Field, "What is the goal of sensory coding?" *Neural Comput.*, vol. 6, no. 4, pp. 559–601, Jul. 1994.
[60] T. S. Evans, "Clique graphs and overlapping communities," *J. Stat. Mech., Theory Exp.*, vol. 2010, no. 12, Dec. 2010, Art. no. P12037.
[61] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen, "Theorems on positive data: On the uniqueness of NMF," *Comput. Intell. Neurosci.*, vol. 2008, pp. 1–9, Mar. 2008.
[62] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech. Theory Express*, vol. 26, no. 9, pp. 219–228, Sep. 2005.
[63] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.
[64] 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.
[65] 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.

Dany Kamuhanda received the bachelor's degree in computer science from the Kigali Institute of Education, Kigali, Rwanda, in 2010, and the master's degree in computer science and information systems from Nelson Mandela Metropolitan University, Port Elizabeth, South Africa, in 2015. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China.

Since 2015, he has been an Assistant Lecturer of computer science with the University of Rwanda, Kigali. His research interests include machine learning and community detection in social networks.



Meng Wang received the B.S. degree in applied mathematics from the Wuhan University of Technology, Wuhan, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan.

His research interests include community detection in social networks.

Kun He (Senior Member, IEEE) received the B.S. degree in physics from Wuhan University, Wuhan, China, in 1993, the M.S. degree in computer science from Huazhong Normal University, Wuhan, in 2002, and the Ph.D. degree in system engineering from the Huazhong University of Science and Technology (HUST), Wuhan, in 2006.

From 2016 to 2017, she was a Mary Shepard B. Upson Visiting Professor of engineering with Cornell University, Ithaca, NY, USA. She is currently a Professor with the School of Computer Science and Technology, HUST. Her research interests include machine learning, deep learning, social networks, and algorithm design and analysis.