IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 20, 2018, accepted May 23, 2018, date of publication June 7, 2018, date of current version June 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2843726

Learning Latent Factors for Community Identification and Summarization

TIANTIAN HE[1], LUN HU[2], KEITH C. C. CHAN[1], AND PENGWEI HU[1]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Pengwei Hu (csphu@comp.polyu.edu.hk)

This work was supported by the National Natural Science Foundation of China under Grant 61602352.

ABSTRACT Network communities, which are also known as network clusters, are typical latent structures in network data. Vertices in each of these communities tend to interact more and share similar features with each other. Community identification and feature summarization are significant tasks of network analytics. To perform either of the two tasks, there have been several approaches proposed, taking into the consideration of different categories of information carried by the network, e.g., edge structure, node attributes, or both aforementioned. But few of them are able to discover communities and summarize their features simultaneously. To address this challenge, we propose a novel latent factor model for community identification and summarization (LFCIS). To perform the task, the LFCIS first formulates an objective function that evaluating the overall clustering quality taking into the consideration of both edge topology and node features in the network. In the objective function, the LFCIS also adopts an effective component that ensures those vertices sharing with both similar local structures and features to be located into the same clusters. To identify the optimal cluster membership for each vertex, a convergent algorithm for updating the variables in the objective function is derived and used by LFCIS. The LFCIS has been tested with six sets of network data, including synthetic and real networks, and compared with several state-of-the-art approaches. The experimental results show that the LFCIS outperforms most of the prevalent approaches to community discovery in social networks, and the LFCIS is able to identify the latent features that may characterize those discovered communities.

INDEX TERMS Network analysis, social network, complex network, graph clustering, community detection, Community summarization, latent factor analysis.

I. INTRODUCTION

Anetwork can be modeled as a graph containing a set of vertices and edges, which represent data entities, and the inter-relationship between them, respectively. Different from random graphs, there are particular latent structures in the real-world graphs that are worthy looking into. Among these latent structures, communities, which are also known as clusters are the most typical ones. How to identify such communities and the features that may characterize them has drawn much attention in recent years [1], [2]. The discovery of such communities in the network is directly related to a number of significant real-world applications, such as social community detection in social networks [3], [14], [21], [38], functional module detection in biological networks [14], [39], and document segmentation [23]–[27], [29], etc.

The identification of communities is concerned by most algorithms related to network analytics. To identify the communities in the network data, there have been a number of so-called graph clustering algorithms proposed. Most of them are able to detect communities based on, not surprisingly some pre-defined measures on edge structure. One of the most widely used measures is modularity [3], which is defined as a function of the differences in density within communities and a null-graph in which vertices are randomly connected. Based on this measure, three approaches [4]–[6] are proposed to detect communities making use of modularity maximization.

Besides these algorithms, there are also other approaches proposed to detect network communities, utilizing other topological measures. For example, a clique-percolation based method is proposed in [7]. In [8], a clustering method called affinity propagation (AP) is proposed to detect clusters by maximizing the similarities of edge structure between candidates of cluster centers and other vertices. By making use

of a concept of link-graph facilitating the link density, link-Com [9] is proposed to detect communities in such link graphs. In [10], spectral clustering (SC) is proposed to detect communities in graphs by making use of normalized cut [11] which may reveal similar edge structures of vertices within the same communities. In [12], another spectral clustering based method, LM is proposed. LM is able to discover network communities by analyzing the spectral features of the transition matrix which is constructed based on the local connectivity of the network. In [13], CoDa is proposed to detect communities in complex networks by modeling community membership of each vertex as posterior probabilities. Such posterior probabilities can be optimized by assigning vertices with similar edge structures with the same membership. In [14], the Mixed Membership Stochastic Models (MMSB) are proposed. MMSB models are able to discover network communities by maximizing the posterior probability that each pair of vertices are connected in the same blocks.

To reveal more meaningful communities in the network, there are some approaches proposed taking into the consideration both edge structure and attributes that may characterize the vertices. In [15] and [16], SA-Cluster and inc-Cluster are proposed to discover network community by making use a neighborhood random walk model in which the transition probability between each pair of vertices is evaluated by the similarity of their local edge structure and attribute similarity. In [17], EDCAR is proposed to detect sub-spaces as clusters, taking into the consideration edge density and attribute similarity between pairwise vertices.

Besides the above algorithms, some model-based approaches are also proposed to identify network communities making use of both edge structure and node attributes. For example, a Bayesian generative model (GBAGC) [18] for clustering network data is proposed. The cluster membership for each vertex in GBAGC can be revealed by estimating a posterior probability measuring the similarity of edge structure and vertex attributes in the cluster. In [19], an algorithm called CESNA is proposed to make use of a generative process to model edge structure and attribute similarity between pairwise nodes. In [20], Circles is proposed to model the communities in social networks as social circles. The community membership for each vertex can be revealed by estimating the posterior probability that measures the similarity between the node attributes and that are commonly observed on other ones in the same circle. In [21], a deep-learning based method (DMNF) for detecting network communities is proposed. DMNF is able to detect communities in the network by performing spectral clustering in a fused network that is learned by deep learning model.

Inspired by probabilistic topic models [22], there are several topic-model based algorithms, including Link-PLSA-LDA [23], Relational Topic Model [24], iTopicModel [25], PL-DC [26] and Block-LDA [27], proposed to discover communities in relational data. The community membership is modeled as a posterior probability that vertices in the same cluster are labeled with similar topics. As such topic model based methods always require for a high computational effort, they are not efficient algorithms for discovering communities and summarizing their features in the network data [19].

There are some other approaches to discovering communities effectively. Different from those model-based ones, such algorithms make use of different objective functions to measure the overall quality of communities and the community membership of each vertex is obtained by optimizing the objective function. For examples, MISAGA [28], and FSPGA [30] are two approaches to clustering in attributed graphs, which are able to perform the task by maximizing the objective function measuring the overall edge density and attribute similarity in all the clusters. In [31], an evolutionary algorithm for community detection in social networks (ECDA) is proposed. ECDA is able to discover communities in network data by maximizing the intra-degrees of attribute similarity between connecting vertices within the clusters. The objective function used by ECDA can be optimized by evolutionary computation.

Though effective in discovering communities in network data, most of the above approaches cannot identify the features that are able to characterize the discovered communities. To address this challenge, several algorithms for community summarization are proposed. Through maximizing the similarity of node features within the same clusters, both community features and community membership for each vertex in the network can be obtained. For example, there are some attempts making use of k-means algorithm [32] to discover communities in the network in which vertices are share higher similarity of attributes. In [33], an algorithm called MAC, which is based on a probabilistic generative model is proposed to discover graph clusters in which vertices are labeled with Boolean attribute values. In [34], a graph summarization algorithm called k-SNAP is proposed to detect graph clusters by grouping vertices into the same cluster according to a similarity measure of attribute values. Though such algorithms may reveal the features that may characterize the communities, they are not effective in discovering meaningful community structures as these methods ignore the edge structure of the network data.

Given the prevalent algorithms, we have the following findings that may motivate us to develop a novel approach. First, most algorithms are proposed to either discover communities, or summarize community features. There are almost no effective algorithms that are able to complete both two tasks simultaneously. For examples, algorithms like MISAGA and FSPGA are very effective in clustering the attributed network, but they cannot summarize community features as they only consider node similarity when performing the task. Second, some approaches are able to simultaneously detect community and summarize their features, e.g., those topic-models based ones, their high computational requirement leads them to be infeasible for the analytics in large network data. To address the mentioned challenges, we propose a novel Latent Factor Model for Community Identification and Summarization (LFCIS). By modeling the

strength of affiliation between vertices and communities w.r.t. edge structure, attribute similarity, and common latent features as low-dimensional latent spaces, LFCIS formulates the task of the identification of communities and their features as an optimization problem which is related to the learning of optimal factors in the mentioned latent spaces. To ensure those vertices sharing similar edge structures and similar attributes to be located into the same cluster, LFCIS adopts an effective method to regulate the structure of latent spaces w.r.t. edge structure and attribute similarity when performing the task. The corresponding latent factors learned by LFCIS may reveal the community membership for each vertex, taking into the consideration edge structure, attribute similarity between vertices, and common features of the discovered communities.

For performance testing, LFCIS is used with both synthetic and real-world networks and compared with several prevalent approaches to community identification or summarization. Having evaluated the discovered communities using ground-truth data, we find that LFCIS outperforms most of state-of-the-art approaches. The communities discovered by LFCIS match with the ground truth better than those done by other baselines. It is said that LFCIS is a very promising approach for community identification and community summarization.

To introduce the details of LFCIS and the corresponding experiments for performance testing, the rest of this paper is organized as the following. In Section II, the mathematical preliminaries and notations used in this paper are introduced. In Section III, how LFCIS models community identification and summarization as an optimization problem, and how LFCIS solves the formulated problem are introduced in detail. In Section IV, we present the experiments that are used to test the performance of LFCIS and other compared baselines. In Section V, the contributions of this paper and the proposals of the future works are summarized.

II. MATHEMATICAL PRELIMINARIES AND NOTATIONS

Given a set of network data containing n vertices, m node attributes, and |E| edges, it can be represented as a graph $G = \{V, E, \Lambda\}$, where V, E, and Λ represent the vertex, edge, and attribute set in the network, respectively. For the vertex set, it is defined as $V = \{v_i | 1 \leq i \leq n\}$. The edge set, is defined as $E = \{e_{ij}=1 |\ v_i$ and v_j are connected$\}$. And the attribute set is defined as $\Lambda = \{\Lambda_i\ | 1 \leq i \leq m\}$. LFCIS makes use of two matrices, $\mathbf{M}$ and $\mathbf{F}$, to represent the edge structure and node attributes in G. $\mathbf{M}$ is an n-by-n adjacency matrix each element of which, say $\mathbf{M}_{ij}$, equals to 1 if v_i and v_j are connected in G, and 0 if they are disconnected. $\mathbf{F}$ is an m-by-n matrix each element of which say $\mathbf{F}_{ij}$, equals to 1 if vertex v_j is associated with attribute Λ_i, and vice versa.

For notations, we use a subscript, e.g., $\mathbf{M}_i$, to represent the ith column of a given matrix, say $\mathbf{M}$. We use $\mathbf{M}_{ij}$, to represent the entry of $\mathbf{M}$, in ith row, jth column. $tr(\cdot)$ represents the matrix trace. $|\cdot|_F$ and $|\cdot|_1$ represent the matrix Frobenius norm, and $l1$ norm, respectively. All these mentioned mathematical preliminaries and notations are used by LFCIS to model the problem of community identification and summarization.

III. LFCIS IN DETAIL

In this section, how LFCIS models the community identification and summarization as an optimization problem, making use of different latent spaces, and how the factors in these latent spaces are fitted, are introduced in detail.

A. MODELING COMMUNITY IDENTIFICATION AND SUMMARIZATION

As mentioned above, there are two sub-tasks, i.e., identifying latent communities, and summarizing their features, that LFCIS has to complete. For the identification of network communities, LFCIS attempts to assign those vertices sharing similar edge structure and node attributes into the same communities. To project each vertex in G from a high dimension into a lower one, LFCIS makes use of a k-by-n latent matrix, $\mathbf{S}$ to represent the latent edge structure for each vertex. Each column of $\mathbf{S}$, say $\mathbf{S}_i$, represents the inter-relationship w.r.t. edge structure between a vertex, say v_i, and k latent structural components. Obviously, a larger value of an element in $\mathbf{S}$, say $\mathbf{S}_{ij}$, means v_j has a stronger relationship with ith latent component. Using another k-by-n matrix, $\mathbf{C}$ to represent the community membership that each vertex belongs to each of the k communities, LFCIS makes use of the difference between the original adjacency matrix of a graph, G and the one that is jointly constructed by $\mathbf{S}$ and $\mathbf{C}$, to measure the structural loss after using $\mathbf{S}$ to project the edge structures of n vertices into the k-dimensional latent space. It is apparent that a minimum of such loss leads to a better projection. And this structural loss function is defined as

$$\begin{aligned} &\text{minimize} \\ &O_1 = |\mathbf{M} - \mathbf{S}^T\mathbf{C}|_F^2 \end{aligned} \tag{1}$$

Besides considering the edge structure of the vertices within the same community, LFCIS also takes into the consideration attribute similarity between each pair of vertices. As the feature vectors for a pair of vertices v_i and v_j, $\mathbf{F}_i$ and $\mathbf{F}_j$ are always with high dimensionality and are always different, LFCIS makes use of the following kernel function to measure the attribute similarity between a pair of vertices, v_i and v_j in G

$$\mathbf{X}_{ij} = exp(-\frac{|\mathbf{F}_i - \mathbf{F}_j|^2}{2\sigma}) \tag{2}$$

(2) is a Gaussian kernel which can be used to measure the overall similarity w.r.t. attributes associated with any pair of vertices in G. A higher value of that means there are more attributes commonly associated with both v_i and v_j, which in other words, v_i and v_j are more similar w.r.t. node attributes. After obtaining the attribute similarity between each pair of vertices, LFCIS uses an n-by-n matrix, $\mathbf{X}$ to represent the attribute similarity between each pair of vertices in G. Similarly, LFCIS uses a k-by-n latent matrix, $\mathbf{B}$ to represent the latent attribute similarity between each vertex and k latent

attribute components. For an element in $\mathbf{B}$, say $\mathbf{B}_{ij}$, its value means the strength of attribute similarity between v_j and ith latent component. Similar to (1), LFCIS makes use of the difference between $\mathbf{X}$ and the one jointly constructed by $\mathbf{B}$, and $\mathbf{C}$, to measure the loss of attribute similarity. It is defined as

$$\begin{aligned} &\text{minimize} \\ &O_2 = |\mathbf{X} - \mathbf{B}^T\mathbf{C}|_F^2 \end{aligned} \tag{3}$$

As LFCIS aims to find k communities in each of which vertices are connecting more and sharing higher attribute similarity with each other, it makes use of the following function to regulate the latent spaces of $\mathbf{S}$ and $\mathbf{B}$

$$\begin{aligned} &\text{minimize} \\ &O_3 = |\mathbf{S} - \mathbf{B}|_F^2 \end{aligned} \tag{4}$$

By making use of (4), the latent spaces, $\mathbf{B}$ and $\mathbf{S}$ are regulated to be similar so that LFCIS is forced to assign those vertices sharing higher similarity of both edge structure and attributes into the same communities, when fitting the model. The adoption of (4) distinguishes LFCIS from most of the previous approaches related to community discovery in the network, as they do not consider to model the interrelationship between latent spaces of edge structure and node attributes.

To summarize the features that are able to characterize the communities, LFCIS assumes that, the community features are hidden in those m attributes in G, and the features for one community are always different from those for others'. Based on this assumption, LFCIS utilizes a k-by-m latent matrix, $\mathbf{A}$ to represent the inter-relationship between each of m attributes and k communities. It is apparent that a higher value of an entry in $\mathbf{A}$, say $\mathbf{A}_{ij}$, means Λ_j is more possible to become a feature characterizing community i. By making use of $\mathbf{C}$ as the latent matrix representing the community membership, LFCIS utilizing the following objective function to measure the overall difference between $\mathbf{F}$ and the one constructed by $\mathbf{A}$ and $\mathbf{C}$

$$\begin{aligned} &\text{minimize} \\ &O_4 = |\mathbf{F} - \mathbf{A}^T\mathbf{C}|_F^2 \end{aligned} \tag{5}$$

It is apparent that when (5) is minimized, the corresponding latent spaces represented by $\mathbf{A}$ may best interpret the features characterizing the k found communities.

Having introduced the objective functions that LFCIS uses to complete the sub-tasks, we may know that minimizing the following function means LFCIS performs the identification of communities and the summarization of community features simultaneously

$$\begin{aligned} &\text{minimize} \\ &O = O_1 + O_2 + O_3 + O_4 \end{aligned} \tag{6}$$

Here we assume that the objectives O_1, O_2, and O_4 share the same latent space representing the community membership. Based on (6), we know that those vertices sharing a higher similarity of edge structure and attributes can be grouped together, so that the community features can be summarized based on the community membership, when (6) is minimized. By ignoring the terms which are independent to the model optimization, minimizing (6) is equivalent to

$$\begin{aligned} &\text{maximize} \\ &O = tr(\mathbf{M}^T\mathbf{S}^T\mathbf{C} + \mathbf{X}^T\mathbf{B}^T\mathbf{C}) + tr(\mathbf{F}^T\mathbf{A}^T\mathbf{C}) + tr(\mathbf{S}^T\mathbf{B}) \\ &\quad - \frac{1}{2}(|\mathbf{S}^T\mathbf{C}|_F^2 + |\mathbf{B}^T\mathbf{C}|_F^2 + |\mathbf{A}^T\mathbf{C}|_F^2 + |\mathbf{B}|_F^2 + |\mathbf{S}|_F^2) \end{aligned} \tag{7}$$

As all the entries in $\mathbf{M}$, $\mathbf{X}$, and $\mathbf{F}$ are non-negative, we propose LFCIS uses the following objective function to perform the tasks of community detection and summarization

$$\begin{aligned} \text{maximize } O &= tr(\mathbf{M}^T\mathbf{S}^T\mathbf{C} + \mathbf{X}^T\mathbf{B}^T\mathbf{C}) \\ &\quad + tr(\mathbf{F}^T\mathbf{A}^T\mathbf{C}) + tr(\mathbf{S}^T\mathbf{B}) \\ &\quad - \frac{1}{2}[|\mathbf{S}^T\mathbf{C}|_F^2 + |\mathbf{B}^T\mathbf{C}|_F^2 + |\mathbf{A}^T\mathbf{C}|_F^2 \\ &\quad + \Omega(\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{C})] \times \Omega(\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{C}) \\ &= |\mathbf{A}|_F^2 + \alpha|\mathbf{A}|_1 + |\mathbf{B}|_F^2 + |\mathbf{S}|_F^2 + |\mathbf{C}|_F^2 \\ \text{subject to } &\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{C} \geq 0 \end{aligned} \tag{8}$$

where Ω contains the regularization terms preventing the latent factors in the latent spaces from overfitting. If (8) can be optimized, LFCIS is able to assign n vertices into k sub-networks in each of which vertices are densely connected, share relatively high attribute similarity with each other, and the community features can be obtained from the m attributes in G. Such k sub-networks possessing the mentioned features are the ones that LFCIS considers as best communities.

B. MODEL OPTIMIZATION

To identify the optimal latent spaces that are used to represent the community structure and community features, LFCIS has to optimize (8). Given the characteristics of (8), we find that it is convex for variables in $\mathbf{C}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$ respectively, when fixing all variables in other matrices. Given this feature, we may derive a series of iterative rules for inferring the optimal latent factors in $\mathbf{C}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$.

1) INFERENCE OF C

Let β_{ij} be the Lagrange multiplier for $\mathbf{C}_{ij} \geq 0$, and the Lagrange function for variables in $\mathbf{C}_{ij}$ is shown as the following

$$L(\mathbf{C}, \beta) = O - tr(\beta^T\mathbf{C}) \tag{9}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\begin{aligned} \frac{\partial L(\mathbf{C}, \beta)}{\partial \mathbf{C}_{ij}} &= [\mathbf{SM} + \mathbf{BX} + \mathbf{AF} - \mathbf{SS}^T\mathbf{C} - \mathbf{BB}^T\mathbf{C} \\ &\quad - \mathbf{AA}^T\mathbf{C} - \mathbf{C} - \beta]_{ij} \\ \beta_{ij} \cdot \mathbf{C}_{ij} &= 0 \\ \beta_{ij} &\geq 0 \end{aligned} \tag{10}$$

Solving the equation system in (10), we may derive the element-wise updating rule for inferring the latent factors in $\mathbf{C}$

$$\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij} \cdot \frac{[\mathbf{SM} + \mathbf{BX} + \mathbf{AF}]_{ij}}{[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij}} \tag{11}$$

2) INFERENCE OF S

Let γ_{ij} be the Lagrange multiplier for $\mathbf{S}_{ij} \geq 0$, and the Lagrange function for variables in $\mathbf{S}$ is shown as the following

$$L(\mathbf{S}, \gamma) = O - tr(\gamma^T \mathbf{S}) \tag{12}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\begin{aligned} \frac{\partial L(\mathbf{S}, \gamma)}{\partial \mathbf{S}_{ij}} &= [\mathbf{CM} + \mathbf{B} - \mathbf{CC}^T\mathbf{S} - \mathbf{S} - \gamma]_{ij} \\ \gamma_{ij} \cdot \mathbf{S}_{ij} &= 0 \\ \gamma_{ij} &\geq 0 \end{aligned} \tag{13}$$

Solving the equation system in (13), we may derive the element-wise updating rule for inferring the latent factors in $\mathbf{S}_{ij}$

$$\mathbf{S}_{ij} \leftarrow \mathbf{S}_{ij} \cdot \frac{[\mathbf{CM} + \mathbf{B}]_{ij}}{[\mathbf{CC}^T\mathbf{S} + \mathbf{S}]_{ij}} \tag{14}$$

3) INFERENCE OF B

Let η_{ij} be the Lagrange multiplier for $\mathbf{B}_{ij} \geq 0$, and the Lagrange function for variables in $\mathbf{B}$ is shown as the following

$$L(\mathbf{B}, \eta) = O - tr(\eta^T \mathbf{B}) \tag{15}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\begin{aligned} \frac{\partial L(\mathbf{B}, \eta)}{\partial \mathbf{B}_{ij}} &= [\mathbf{CX} + \mathbf{S} - \mathbf{CC}^T\mathbf{B} - \mathbf{B} - \eta]_{ij} \\ \eta_{ij} \cdot \mathbf{B}_{ij} &= 0 \\ \eta_{ij} &\geq 0 \end{aligned} \tag{16}$$

Solving the equation system in (16), we may derive the element-wise updating rule for inferring the latent factors in $\mathbf{B}$

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \cdot \frac{[\mathbf{CX} + \mathbf{S}]_{ij}}{[\mathbf{CC}^T\mathbf{B} + \mathbf{B}]_{ij}} \tag{17}$$

4) INFERENCE OF A

Let μ_{ij} be the Lagrange multiplier for $\mathbf{A}_{ij} \geq 0$, and the Lagrange function for variables in $\mathbf{A}$ is shown as the following

$$L(\mathbf{A}, \mu) = O - tr(\mu^T \mathbf{A}) \tag{18}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\begin{aligned} \frac{\partial L(\mathbf{A}, \mu)}{\partial \mathbf{A}_{ij}} &= [\mathbf{CF}^T - \mathbf{CC}^T\mathbf{A} - \mathbf{A} - \mu]_{ij} - \alpha \\ \mu_{ij} \cdot \mathbf{A}_{ij} &= 0 \\ \mu_{ij} &\geq 0 \end{aligned} \tag{19}$$

Solving the equation system in (19), we may derive the element-wise updating rule for inferring the latent factors in $\mathbf{A}$

$$\mathbf{A}_{ij} \leftarrow \mathbf{A}_{ij} \cdot \frac{[\mathbf{CF}^T]_{ij}}{[\mathbf{CC}^T\mathbf{A} + \mathbf{A}]_{ij} + \alpha} \tag{20}$$

By iteratively updating latent factors in $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$, and $\mathbf{A}$, respectively, while fixing the others, LFCIS is able to find the optimal latent factors that maximize (8).

C. CONVERGENCE ANALYSIS

To prove the convergence of the algorithm, we may make use of one property of an auxiliary function that is also used in the proof of the Expectation-Maximization algorithm [35]. The property of the auxiliary function is described as the following. If there exists an auxiliary function satisfying the conditions that $Q(x,x') \leq F(x)$ and $Q(x,x) = F(x)$, then F is non-decreasing under the updating rule that

$$x^{t+1} = \underset{x}{argmax}\, Q(x, x^t) \tag{21}$$

The equality $F(x^{t+1}) = F(x^t)$ holds only when x is a local maximum of $Q(x,x^t)$. By iteratively updating x according to (21), F will converge to the local maximum $x^{max} = argmax_x F(x)$. By defining an appropriate auxiliary function for O, we may show the convergence of (8).

First, we may prove the convergence of the updating rule (11) for the inference of $\mathbf{C}$. Let $\mathbf{C}_{ij}$ be any element in $\mathbf{C}$, $O_{\mathbf{C}_{ij}}$ be the partial of (8) that is related to $\mathbf{C}_{ij}$, $O_{\mathbf{C}_{ij}}(\mathbf{C}'_{ij})$ be the partial objective value of (8) that is related to $\mathbf{C}_{ij}$ when $\mathbf{C}_{ij}$ is equal to some value, say $\mathbf{C}'_{ij}$. Since the updating rule for $\mathbf{C}$ is element wise, it is sufficient to show $O_{\mathbf{C}_{ij}}$ is non-decreasing according to the updating rule (11). To prove this, we define the following auxiliary function for $O_{\mathbf{C}_{ij}}$:

$$\begin{aligned} Q(c, \mathbf{C}^t_{ij}) = O_{\mathbf{C}_{ij}}(\mathbf{C}^t_{ij}) + O'_{\mathbf{C}_{ij}}(c - \mathbf{C}^t_{ij}) \\ - \frac{[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij}}{2\mathbf{C}^t_{ij}}(c - \mathbf{C}^t_{ij})^2 \end{aligned} \tag{22}$$

where $O'_{\mathbf{C}_{ij}}$ is the first order partial derivative of (8) relevant to $\mathbf{C}_{ij}$. Although the auxiliary function is defined in (22), we need to prove it satisfies the aforementioned conditions. Apparently, $Q(c, c) = O_{\mathbf{C}_{ij}}(c)$. Hence, the left we need to prove is $Q(c,\mathbf{C}^t_{ij}) \leq O_{\mathbf{C}_{ij}}(c)$. To prove this, we compared $Q(c, \mathbf{C}^t_{ij})$ shown in (22) with the Taylor expansion of $O_{\mathbf{C}_{ij}}$ near to $\mathbf{C}^t_{ij}$

$$O_{\mathbf{C}_{ij}}(c) = O_{\mathbf{C}_{ij}}(\mathbf{C}^t_{ij}) + O'_{\mathbf{C}_{ij}}(c - \mathbf{C}^t_{ij}) + \frac{1}{2}O''_{\mathbf{C}_{ij}}(c - \mathbf{C}^t_{ij})^2 \tag{23}$$

where $O'_{\mathbf{C}_{ij}}$ and $O''_{\mathbf{C}_{ij}}$ are the first and second order partial derivatives relevant to $\mathbf{C}_{ij}$. Note that

$$\begin{aligned} O'_{\mathbf{C}_{ij}} &= \frac{\partial O}{\partial \mathbf{C}_{ij}} \\ &= [\mathbf{SM} + \mathbf{BX} + \mathbf{AF} - \mathbf{SS}^T\mathbf{C} - \mathbf{BB}^T\mathbf{C} - \mathbf{AA}^T\mathbf{C} - \mathbf{C}]_{ij} \\ O''_{\mathbf{C}_{ij}} &= \frac{\partial^2 O}{\partial (\mathbf{C}_{ij})^2} = -[\mathbf{SS}^T + \mathbf{BB}^T + \mathbf{AA}^T]_{ii} - 1 \end{aligned} \tag{24}$$

Using (24) to replace the relevant terms in (23), we can see that if $Q(c,\mathbf{C}^t_{ij}) \leq O_{\mathbf{C}_{ij}}(c)$, the following inequality must hold

$$\begin{aligned} -\frac{[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij}}{2\mathbf{C}^t_{ij}} &\leq \frac{1}{2} O''_{\mathbf{C}_{ij}} \\ &= -\frac{1}{2}[\mathbf{SS}^T + \mathbf{BB}^T + \mathbf{AA}^T]_{ii} - \frac{1}{2} \end{aligned} \tag{25}$$

Therefore, to show $Q(c,\mathbf{C}^t_{ij}) \leq O_{\mathbf{C}_{ij}}(c)$, it is equivalent to show

$$\begin{aligned} &[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij} \\ &\qquad \geq \mathbf{C}^t_{ij}[\mathbf{SS}^T + \mathbf{BB}^T + \mathbf{AA}^T]_{ii} + \mathbf{C}^t_{ij} \end{aligned} \tag{26}$$

Since the elements in $\mathbf{C}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{S}$ are non-negative, we have

$$\begin{aligned} &[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij} \\ &= \sum_l [\mathbf{SS}^T_{il}\mathbf{C}^t_{lj} + \mathbf{BB}^T_{il}\mathbf{C}^t_{lj} + \mathbf{AA}^T_{il}\mathbf{C}^t_{lj}] + \mathbf{C}^t_{ij} \\ &\geq \mathbf{SS}^T_{ii}\mathbf{C}^t_{ij} + \mathbf{BB}^T_{ii}\mathbf{C}^t_{ij} + \mathbf{AA}^T_{ii}\mathbf{C}^t_{ij} + \mathbf{C}^t_{ij} \end{aligned} \tag{27}$$

Up to here, $Q(c,\mathbf{C}^t_{ij}) \leq O_{\mathbf{C}_{ij}}(c)$ has been proved thus (22) is an auxiliary function for $O_{\mathbf{C}_{ij}}$.

Next, we will define the auxiliary functions regarding to the updating rules for the inference of $\mathbf{S}$, $\mathbf{B}$, and $\mathbf{A}$, which are shown in (14), (17), and (20). Similarly, let $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$ be the partial of (8) relevant to $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$, $O_{\mathbf{S}_{ij}}(\mathbf{S}'_{ij})$, $O_{\mathbf{B}_{ij}}(\mathbf{B}'_{ij})$, and $O_{\mathbf{A}_{ij}}(\mathbf{A}'_{ij})$ be the partial objective values when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ equal to $\mathbf{S}'_{ij}$, $\mathbf{B}'_{ij}$ and $\mathbf{A}'_{ij}$, respectively. Since the updating rules for the inferring $\mathbf{S}$, $\mathbf{B}$, and $\mathbf{A}$ are also element wise, it is sufficient to show that $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$ are non-decreasing according to the updating rules (14), (17), and (20). Let the following be the auxiliary functions regarding to $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$

$$\begin{aligned} Q(s, \mathbf{S}^t_{ij}) &= O_{\mathbf{S}_{ij}}(\mathbf{S}^t_{ij}) + O'_{\mathbf{S}_{ij}}(s - \mathbf{S}^t_{ij}) \\ &\quad - \frac{[\mathbf{CC}^T\mathbf{S} + \mathbf{S}]_{ij}}{2\mathbf{S}^t_{ij}}(s - \mathbf{S}^t_{ij})^2 \\ Q(b, \mathbf{B}^t_{ij}) &= O_{\mathbf{B}_{ij}}(\mathbf{B}^t_{ij}) + O'_{\mathbf{B}_{ij}}(b - \mathbf{B}^t_{ij}) \\ &\quad - \frac{[\mathbf{CC}^T\mathbf{B} + \mathbf{B}]_{ij}}{2\mathbf{B}^t_{ij}}(b - \mathbf{B}^t_{ij})^2 \\ Q(a, \mathbf{A}^t_{ij}) &= O_{\mathbf{A}_{ij}}(\mathbf{A}^t_{ij}) + O'_{\mathbf{A}_{ij}}(a - \mathbf{A}^t_{ij}) \\ &\quad - \frac{[\mathbf{CC}^T\mathbf{A} + \mathbf{A}]_{ij} + \alpha}{2\mathbf{A}^t_{ij}}(a - \mathbf{A}^t_{ij})^2 \end{aligned} \tag{28}$$

Since the proof for the above functions to be auxiliary functions for $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$ is similar to that for $O_{\mathbf{C}_{ij}}$, we don't show the proof in detail due to the space limitation.

Having obtained the auxiliary functions for $O_{\mathbf{C}_{ij}}$, $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$, now we can show the convergence of (8) using the updating rules (11), (14), (17) and (20). Since (22) is an auxiliary for $O_{\mathbf{C}_{ij}}$, according to (21), we have

$$\begin{aligned} \mathbf{C}^{t+1}_{ij} &= \underset{c}{argmax}\, Q(c, \mathbf{C}^t_{ij}) \\ &= \mathbf{C}^t_{ij} \cdot \frac{[\mathbf{SM} + \mathbf{BX} + \mathbf{AF}]_{ij}}{[\mathbf{SS}^T\mathbf{C} + \mathbf{BB}^T\mathbf{C} + \mathbf{AA}^T\mathbf{C} + \mathbf{C}]_{ij}} \end{aligned} \tag{29}$$

The above result is the same to the updating rule (11). Since (22) is an auxiliary function, $O_{\mathbf{C}_{ij}}$ is non-decreasing when $\mathbf{C}_{ij}$ is updated according to (29) or (11). This is equivalent to say that O is non-decreasing when $\mathbf{C}_{ij}$ is updated according to (11) as $\mathbf{C}_{ij}$ is any element of $\mathbf{C}$.

Since (28) are auxiliary functions for $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$, according to (21), we have

$$\begin{aligned} \mathbf{S}^{t+1}_{ij} &= \underset{s}{argmax}\, Q(s, \mathbf{S}^t_{ij}) = \mathbf{S}^t_{ij} \cdot \frac{[\mathbf{CM} + \mathbf{B}]_{ij}}{[\mathbf{CC}^T\mathbf{S} + \mathbf{S}]_{ij}} \\ \mathbf{B}^{t+1}_{ij} &= \underset{b}{argmax}\, Q(b, \mathbf{B}^t_{ij}) = \mathbf{B}^t_{ij} \cdot \frac{[\mathbf{CX} + \mathbf{S}]_{ij}}{[\mathbf{CC}^T\mathbf{B} + \mathbf{B}]_{ij}} \\ \mathbf{A}^{t+1}_{ij} &= \underset{a}{argmax}\, Q(a, \mathbf{A}^t_{ij}) = \mathbf{S}^t_{ij} \cdot \frac{[\mathbf{CF}^T]_{ij}}{[\mathbf{CC}^T\mathbf{A} + \mathbf{A}]_{ij} + \alpha} \end{aligned} \tag{30}$$

The above results are the same to the updating rules (14), (17), and (20). Since (28) are auxiliary functions, $O_{\mathbf{S}_{ij}}$, $O_{\mathbf{B}_{ij}}$, and $O_{\mathbf{A}_{ij}}$ are non-decreasing when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ are updated according to (14), (17), and (20). This is equivalent to say that O is non-decreasing when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ are updated according to (14), (17), and (20), respectively. The above proof shows that O is non-decreasing when $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$ are iteratively updated according to (11), (14), (17) and (20). Thus, we have

$$\begin{aligned} &O(\mathbf{C}^0, \mathbf{S}^0, \mathbf{B}^0, \mathbf{A}^0) \leq O(\mathbf{C}^1, \mathbf{S}^0, \mathbf{B}^0, \mathbf{A}^0) \\ &\quad \leq O(\mathbf{C}^1, \mathbf{S}^1, \mathbf{B}^0, \mathbf{A}^0) \leq \cdots \leq O(\mathbf{C}^{opt}, \mathbf{S}^{opt}, \mathbf{B}^{opt}, \mathbf{A}^{opt}) \end{aligned} \tag{31}$$

where O shows a non-decreasing trend in each iteration of updating and it may finally achieve to the local optima.

D. THE TERMINATION OF OPTIMIZATION

As $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$ are iteratively updated, the objective value converges to the local optima asymptotically. Simultaneously, the variation of the four matrices, becomes less evident as the elements in each matrix are approximate to the magnitudes which lead the objective value to the local optima. Thus, we may use the following stopping criterion to determine whether the optimization process should be terminated and LFCIS may obtain optimal latent factors in matrices $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$ that lead O to converge

$$|\mathbf{C}^t - \mathbf{C}^{t-1}|_F < \tau \tag{32}$$

where $\mathbf{C}^t$ stands for the latent space representing the community membership after the tth iteration of updating, τ represents the predefined tolerance which the Frobenius norm of the difference of $\mathbf{C}$ between two iterations

should satisfy. When τ is set to be a relatively small value, LFCIS may obtain a latent matrix $\mathbf{C}$ which is very approximate to the optimal.

E. SUMMARY REMARKS

Having obtained the updating rules for $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$ and the stopping criterion for the optimization process, now we may describe the details of LFCIS. Based on the aforementioned description, the proposed algorithm for learning the latent factors in LFCIS can be summarized as the pseudo codes shown in Algorithm 1. As it is seen, there are not many parameters that need to be input. After the parameters of maximum number of iteration *maxiter*, tolerance for improvement τ, penalty factor α and the dimensionality of latent spaces, k are determined, LFCIS will iteratively update the matrices $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$, in which are the latent factors representing the community membership and their features, till the variation of $\mathbf{C}$ between two iterations is less than τ or the objective function converges to the local maxima. After the optimization process is terminated, LFCIS obtains the matrices for community membership and features, $\mathbf{C}$ and $\mathbf{A}$ which contain the optimal or approximately optimal membership between each vertex and k communities and the community features generated based on the m attributes in G. Given $\mathbf{C}$, LFCIS can directly identify the best community membership for each vertex in the network.

For the model comlexity, we mainly analyze the computational cost when LFCIS is updating those variables in $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$ and $\mathbf{A}$ in each iteration. Based on (11), updating all the factors in $\mathbf{C}$ follows the order of $O(k^2(2n^2 + 3n + mn) + k(2n^2+mn+n))$. Based on (14), updating all the latent factors in $\mathbf{S}$ follows the order of $O(k^2(n^2 + n) + k(n^2 + 2n))$. Based on (17), updating all the variables in $\mathbf{B}$ follows the order as what updating $\mathbf{S}$ does. Based on (20), updating all the latent factors in $\mathbf{A}$ follows the order of $O((k^2 + k)(nm + m))$. It is seen that the computational complexity of LFCIS is about the order of $O(n^2)$. As $\mathbf{M}$, $\mathbf{X}$, and $\mathbf{F}$ are always very sparse, the complexity of LFCIS should be much lower than the analytical.

Algorithm 1 Inference of Latent Factors in LFCIS

Input: $\mathbf{M}$, $\mathbf{X}$, $\mathbf{F}$, α, *maxiter*, τ, k
Output: $\mathbf{C}$, $\mathbf{S}$, $\mathbf{B}$, $\mathbf{A}$

```
Randomly initialize C, S, B, A;
for count = 1 : maxiter do
    Fixing S, B, A
    updating C according to (11);
    Fixing C
    updating A according to (20);
    Fixing B
    updating S according to (14);
    Fixing S
    updating B according to (17);
    if |C^i − C^{i−1}|_F < τ then
        compute objective value according to (8);
        break;
    end if
end for
return C, S, B, A;
```

IV. EXPERIMENTS AND ANALYSIS

To evaluate the effectiveness of LFCIS, we performed a number of experiments using both synthetic and real-world datasets. In this section, we present the details of the datasets we used, the criteria we used to evaluate the performance, and how we performed the experiments.

A. EXPERIMENTAL SET-UP AND PERFORMANCE METRICS

1) DATASETS DESCRIPTIONS

We used both synthetic and real datasets with known ground truth for performance evaluations. We used synthetic data to test the effectiveness, efficiency of LFCIS and other compared baselines, and parameter sensitivity of LFCIS. We used real-world datasets to test the robustness of different algorithms. The details of datasets we used are described below.

There are five real-world datasets used in our experiments, including *Caltech* [36], *Twitter* [20], *Ego-facebook* [19], *Googleplus-1* [20], and *Googleplus-2* [20], all of which are collected from online-social networking sites and are widely used as testing datasets for community identification.

Caltech is a set of network data which is constructed based on the friendship of social network users from California Institute of Technology. The social network users at Caltech can be segmented into 10 large classes according to the college's dorm system [36]. There are 769 vertices representing 769 social network users, and 16656 edges representing social ties between these users. A total of 53 attributes represent the user profiles.

Twitter dataset is constructed based on a number of social circles extracted from twitter.com. For this dataset, there are 2511 vertices representing twitter users, 37154 edges representing the friendship between them, and 9067 attributes representing social topics they concern, and the locations where the users post twits. There are 132 social circles that have been verified as ground truth communities.

Ego-facebook is a set of social network data that are constructed based on a number of sub-networks extracted from facebook.com. In this dataset, there are 4039 vertices that represent facebook users, 88234 edges representing online social ties between these users, and 1283 attributes that represent the user profiles. 191 social communities have been verified as ground truth communities so that they can be used for benchmarking the identified ones.

Googleplus-1 is a set of online social network data which are collected from plus.google.com. There are 5630 vertices, 463537 edges, and 4229 attributes in the dataset. In this dataset, vertices, edges, and attributes represent googleplus users, friendships, and user profiles, respectively. There are

58 social clusters that have been verified as ground truth communities in *Googleplus-1*.

Googleplus-2 is another set of social network data which are constructed based on the sub-networks from plus.google.com. There are 7856 vertices, 321268 edges, and 2024 attributes in the dataset. In this dataset, there are 91 social communities of ground truth that are able to be used for benchmarking the identified ones.

Syn1k is a set of synthetic data which is generated based on the rule that the probability of intra-community edges is higher than that of inter-community edges and that vertices in the same cluster are more related to each other than those that are not. In *Syn1k*, there are 1000 vertices that are divided into 4 disjoint ground truth communities, 9900 edges and 50 attributes that are possibly associated with each vertex.

The above data sets are used to test the effectiveness of LFCIS and other algorithms. In addition, to test the scalability of LFCIS, we have generated several additional synthetic datasets ranging in the size from 5,000 to 100,000 for our experiments.

2) EVALUATION METRICS

For performance evaluation, we are considering different evaluation measures which are widely used for evaluating network clustering algorithms. For measures used for validating graph clusters, we used the Normalized Mutual Information (*NMI*), and the Average Accuracy (*Acc*) [37].

The *NMI* measures the overall accuracy of the matches between detected communities and those that are considered as ''ground truth''. It is defined as

$$\begin{aligned} NMI &= \frac{\sum_{C_i, C_j^*} Pr(C_i, C_j^*) log \frac{Pr(C_i, C_j^*)}{Pr(C_i) Pr(C_j^*)}}{\max(H(C), H(C^*))} \\ H(C) &= -\sum_i Pr(C_i) log Pr(C_i) \\ H(C^*) &= -\sum_j Pr(C_j^*) log Pr(C_j^*) \end{aligned} \tag{33}$$

where $\Pr(C_i, C_j^*)$ denotes the probability that vertices are in both the detected community i and the true community j, and $\Pr(C_i)$ denotes the probability that a vertex is found to exist in community i. Based on this definition, if the *NMI* measure is high, it means that the communities detected match well with the ground-truth ones.

Contrary to the *NMI*, the *Acc* measure evaluates the detected community individually. It is defined as

$$Acc = \sum_i \frac{|C_i|}{|C|} f(C_i, C^*) \tag{34}$$

where $|C|$ means the size of the detected communities, and $f(\cdot)$ stands for a mapping function between cluster i and the ground truth. For our purpose, we define $f(\cdot)$ to be the maximum overlap between detected community i and a ground-truth community. Thus, *Acc* evaluates the best matching of each cluster. A higher value of *Acc*, therefore means that each detected community has a better match with the ground truth. The higher the *Acc* of all communities detected by an algorithm, therefore means that the algorithm is more effective.

3) BASELINES FOR COMPARISON

To test the effectiveness of LFCIS, we selected a number of approaches as compared baselines. These algorithms include CNM [5] Affinity Propagation clustering (AP) [8], Spectral clustering (SC) [10], k-means clustering [32], Relational topic model (RTM) [24], CESNA [20], ECDA [19], and MISAGA [28]. Selecting these algorithms as baselines is because they are either the latest algorithms or classical ones and have been used effectively to detect network communities in various networks. Specifically, CNM is an effective algorithm for community detection which is based on modularity optimization. AP and SC may detect clusters that take different topological properties of network data. For our experiments, we used the SC that makes use of the normalized cut [11] in graph clustering. k-means is able to detect graph communities through grouping together those vertices with similar attributes. Therefore, we used the information in Λ as the input that is used to compute the similarity between pairwise vertices for k-means. Algorithms like RTM, CESNA, ECDA, and MISAGA are ones taking into consideration both graph topologies and attributes. RTM has been shown to be a very effective topic-model based approach to segment relational data. CESNA is able to discover network communities by maximizing the logarithmic posterior probability of structural and attribute similarity between pairwise vertices. ECDA performs its tasks using an evolutionary graph clustering algorithm. MISAGA is a very effective algorithm which is proposed recently. It can perform the task of community detection in graphs taking into the consideration edge structure and attribute similarity between pairwise vertices.

For performance benchmarking, we used the source code or executables made available by the authors. All the experiments were conducted under the same environment which is included into a workstation with 4-core 3.4GHz CPU and 16GB RAM.

4) EXPERIMENTAL SET-UP

To ensure that the algorithms we used in the experiment may obtain a robust performance, we tested them using the parameters in such a way that either the default settings as recommended by the authors are used or that they are tuned by trials to find the best settings.

Specifically, we let CNM, AP, CESNA, and ECDA detect network communities using the default settings as all of them do not require input parameters. For algorithms, including SC, k-means, MISAGA, and RTM, which require parameters to manually input into the system, we tried as many different settings as we can, to obtain the best results for performance benchmarking. For example, SC requires that the parameter of σ to be set by the users before it can run. To find a better set of parameters, we tried SC using

different σ from 1 to 10. The settings that give the best performance of SC are recorded and presented in our performance analysis below. As for the number of clusters, k, we set it for those algorithms that need k as a predefined parameter, including, SC, k-means, MISAGA, and RTM, to be equal to the number of ground truth communities in each dataset.

For LFCIS, we set α to 0.5 to control the sparsity of $\mathbf{A}$. As for the other parameters, we set the maximum number of iterations to 500, and τ to 1e-6. As for k, it is set to be the same as the other algorithms, which is equal to the number of ground-truth communities in each of the datasets. All the algorithms, including LFCIS, were executed 10 times to obtain statistical averages for the performance measures.

B. EXPERIMENTAL RESULTS USING SYNTHETIC DATA

1) THE PERFORMANCE OF COMMUNITY DETECTION

For performance evaluation, we used a set of synthetic network data containing 1000 vertices to test the effectiveness of all algorithms. There are four disjoint ground truth clusters in the synthetic dataset. As mentioned above, the synthetic data are generated by assuming that the probability of vertices within the same community to be connected with other vertices to be higher than that of the probability between communities. For our experiment, the data set *Syn1k* was generated by setting the probability of intra-community connections to be 0.05 and the probability of inter-community connections to be 0.01.

The performance of LFCIS and other algorithms in *Syn1k* with respect to *NMI*, and *Acc* is given in Table 1. As the table shows, LFCIS performs better than other algorithms. No matter which of *NMI*, or *Acc* is considered, LFCIS may outperform all the compared baselines in dataset *Syn1k*. These experimental results show that LFCIS can be very effective in the discovering of communities in the synthetic network data.

TABLE 1. ***NMI*** **and** ***Acc*** **in** ***Syn1k*****.**

Approach	*NMI*	*Acc*
CNM	0.813	0.939
AP	0.152	0.747
SC	0.232	0.528
k-means	0.691	0.835
RTM	0.766	0.797
CESNA	0.657	0.844
ECDA	0.272	0.466
MISAGA	0.981	0.996
LFCIS	**0.995**	**0.999**

2) SENSITIVITY TEST OF α

As mentioned in Section III, there is only one parameter, α, which is used to control the sparsity of $\mathbf{A}$ in LFCIS, and it might take effect on the performance of the model. To investigate how the parameter α may take effect on the performance of LFCIS, we performed the sensitivity test using the dataset *Syn1k*. In our experiment, α was set to different values from 0.1 to 2, with an increment of 0.1, and LFCIS was used under these settings to fit the model for discovering communities. The performance was measured with *NMI*, and *Acc* and the results are shown in Fig. 1.

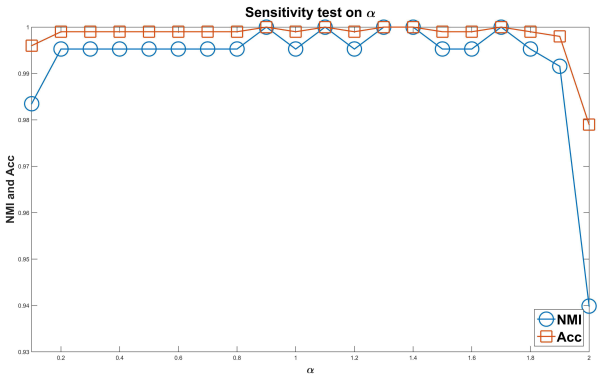


FIGURE 1. **Sensitivity test of α.**

As it is shown in the figure, LFCIS may obtain a worse performance when α is set to be either near to 0, or near to 2. LFCIS may perform steadily when α is set to a value between 0.2 and 1.5. In our experiments, we set α to 0.5, when LFCIS performs the tasks of community identification and summarization in all the datasets. Using this setting may guide LFCIS to exclude those attributes with relatively lower possibility of being ones that may characterize the identified communities, while preserve those that are more possible to be community features.

3) SCALABILITY TEST

In order to find how LFCIS may scale up when the size of the graph data increases, we used a series of synthetic datasets ranging from 5,000 to 100,000, that are generated using the same probabilities of 0.05 and 0.01 for intra- and inter-community connections as is with *Syn1k*, to test the scalability of LFCIS and compare it with MISAGA, SC, and RTM. As all of these algorithms are based on iterative optimization, the comparison is made based on the average execution time per iteration. The experimental results of scalability comparison are shown in Fig. 2.

As it is shown in the figure, LFCIS scales up well when compared with MISAGA, RTM, and SC. Even when the data size increases up to 100,000 vertices, LFCIS is able to complete each iteration of parameter updating around 1 second. This is slightly faster than MISAGA. When MISAGA and LFCIS use the same setting of maximum number of iterations for optimization, LFCIS is able to identify the community membership and summarize the community features in a shorter time. Given this fact, LFCIS is more efficient.

As for RTM and SC, the computational time used by them was much more demanding than LFCIS. When the data size is increased to 10,000, RTM and SC are already not able to cope. The computational time required by RTM and SC is intolerable.

TABLE 2. Experimental results in real-world datasets.

Datasets	*Caltech*		*Twitter*		*Ego-facebook*		*Googleplus-1*		*Googleplus-2*	
Approach	*NMI*	*Acc*	*NMI*	*Acc*	*NMI*	*Acc*	*NMI*	*Acc*	*NMI*	*Acc*
CNM	0.423	0.309	0.378	0.283	0.483	0.380	0.151	0.290	0.336	0.329
AP	0.381	0.458	0.609	0.479	0.571	0.416	0.267	0.525	0.330	0.273
SC	0.411	0.335	0.479	0.305	0.536	0.447	0.103	0.321	0.170	0.296
k-means	0.211	0.149	0.286	0.239	0.405	0.291	0.146	0.217	0.255	0.195
RTM	0.242	0.146	0.196	0.099	0.254	0.167	0.077	0.309	0.280	0.151
CESNA	0.393	0.384	0.577	0.473	0.491	0.384	0.238	0.314	0.410	0.248
ECDA	0.260	0.202	0.527	0.385	0.333	0.234	0.325	0.468	0.265	0.255
MISAGA	0.298	0.256	0.658	**0.503**	0.565	0.452	0.524	0.735	0.452	0.363
LFCIS	**0.521**	**0.475**	**0.688**	0.493	**0.624**	**0.512**	**0.555**	**0.741**	**0.630**	**0.473**
Improvement(%)	**23.17**	**3.71**	**4.56**	-1.98	**9.28**	**13.27**	**5.91**	**0.82**	**39.38**	**30.3**

Improvement(%): The percentage of improvement when LFCIS is compared with the second-best approach in different datasets

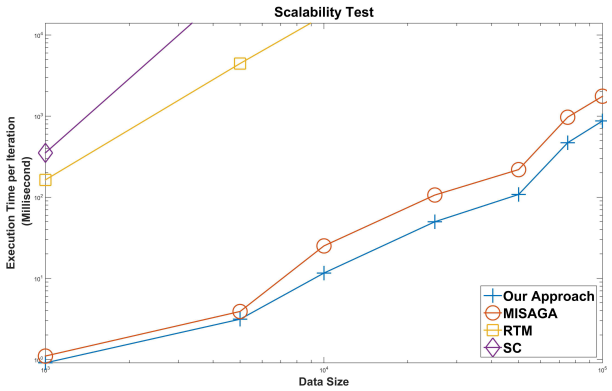


FIGURE 2. Scalability comparison between LFCIS and other approaches.

C. EXPERIMENTAL RESULTS USING REAL-WORLD DATA

Community detection is one of the most significant tasks of network analysis. To test the effectiveness of LFCIS and other compared baselines, we use them to perform the task of community detection in five sets of real-world social network data, including *Caltech*, *Twitter*, *Ego-facebook*, *Googleplus-1*, and *Googleplus-2*. These five sets of real-world data are different from vertex size and the dimensionality of attributes that are used to characterize the vertices. All these datasets have known ground truth communities which have been verified in the previous works. For this reason, the performance of LFCIS and other baselines can be more objectively compared.

The experimental results of *NMI* and *Acc* obtained with these datasets are summarized in Table 2. As the tables show, LFCIS performs more robustly, compared with other baselines. When *NMI* is considered, LFCIS is better than any other baselines in all the five datasets. LFCIS outperforms the second-best methods by 23.17%, 4.56%, 9.28%, 5.91%, and 39.38% in *Caltech*, *Twitter*, *Ego-facebook*, *Googleplus-1*, and *Googleplus-2*, respectively. When *Acc* is considered, LFCIS is better than any other baselines, except the case in *Twitter* dataset. In *Caltech*, *Ego-facebook*, *Googleplus-1*, and *Googleplus-2*, the improvement related to *Acc*, is 3.71%, 13.27%, 0.82%, and 30.3%, respectively, when LFCIS is compared with the second-best algorithms. Given the robust performance obtained by LFCIS in these real-world datasets, it is said that LFCIS is a very effective model for identifying latent communities in social network data, while ensuring the community features also to be identified.

D. CASE STUDY-THE COMMUNITY FEATURES AND MEMBER ATTRIBUTES

To investigate whether LFCIS is able to effectively summarize the features which may be used to characterize a particular community, we compared the community features identified by LFCIS with those attributes shared by the vertices in the same community. Here we make a detailed analysis on a community identified by LFCIS to show the effectiveness on the community summarization of LFCIS. In *Twitter* dataset, one social community was identified by LFCIS and its structure almost match one in the ground truth database. The structure of this ground truth community is shown in Fig. 3. In fact, only one vertex, 537550306 was not successfully identified by LFCIS in our experiments. Making use of $\mathbf{A}$ fitted by LFCIS, we also find that there are 14 attributes with higher probabilities that may characterize this community. These attributes are shown Table 3. Given them, we may conclude that this social community is about the campaigns related to some sports, e.g., wrestling. And attributes of the members in this community, should be related to the community features, to some extent. In Table 3, we also list the attributes shared by two members in this community, 533426485 and 41284429. As it is shown in the table, topics like "SmackDown!," "SuperShow?," and "SurvivorSeries" are all related to an American professional wrestling event, i.e., WWE. And these topics have been identified by LFCIS as features of this community. Similarly, such overlap can also be found between the attributes of other community members and the community features. Given this fact, it is said that, LFCIS is able to summarize the community features, while ensuring its robust performance on community detection in network data.

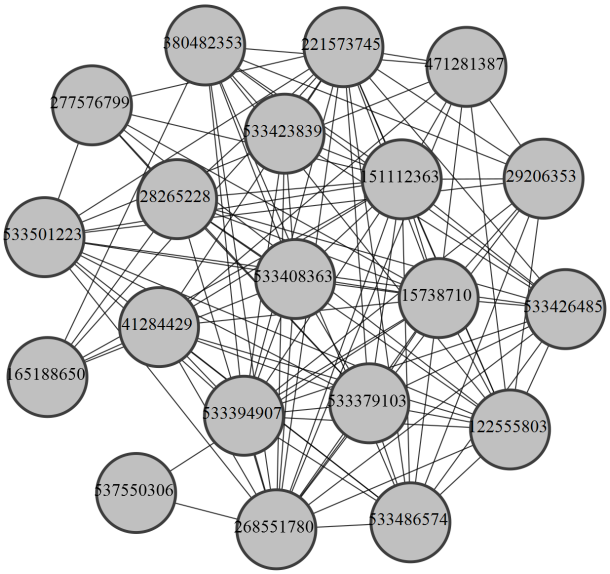


FIGURE 3. **The structure of a ground truth community in Twitter dataset. LFCIS identified all the vertices except of 537550306 .**

TABLE 3. **Community features and attributes shared by community members.**

Community Features identified by LFCIS
#39;t, #Getchapopcornready, **#SmackDown!***, **#SuperFriends**, **#SuperShow?**, **#SurvivorSeries**, #ThankYouEdge, #UFC, #WRESTLEMANIA, #WrestleMania28, #f, @AllyKeyoni, @DaveSeperson, @EveMarieTorres.
#SmackDown!, #SuperFriends, #SuperShow?, #SurvivorSeries
Member 1: 533426485 Member 2: 41284429
Shared Attributes between two community members

* The features belonging to both the community and members are in bold fonts.

V. CONCLUSION

In this paper, a novel latent factor model for community identification and summarization, LFCIS is proposed. Different from most prevalent approaches that focus on either community identification or community summarization, LFCIS is able to complete the two tasks simultaneously. Taking into the consideration edge structure and vertex attributes in the network, LFCIS formulates the identification of community and summarization of community features as an optimization problem. And the optimal community membership for each vertex can be learned by LFCIS through a series of rules for updating the model parameters. What distinguishes LFCIS from other approaches is that LFCIS is considering to model the interrelationship between the latent spaces w.r.t. structure and attribute similarity so that those vertices sharing similar latent structures w.r.t. topology and attributes are more probably assigned with similar community memberships. LFCIS has been tested with both synthetic and real-world network data and has been compared with several prevalent algorithms for community discovery or community summarization. It outperforms most state-of-the-art approaches in the experiments related to the test of effectiveness and efficiency. It is concluded that LFCIS is a very promising approach to identifying communities and summarizing their features in the network data. In future, we will test LFCIS using more comprehensive metrics to reveal its effectiveness. We will attempt to improve the efficiency of LFCIS and develop the parallel version of LFCIS. And we will also try to investigate the feasibility of making use of LFCIS to detect overlapping communities in network data.

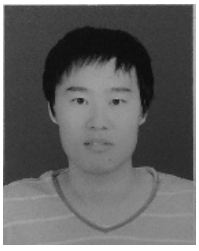
ACKNOWLEDGMENT

(Tiantian He and Lun Hu contributed equally to this work.)

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.
[2] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proc. WWW*, 2010, pp. 631–640.
[3] 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, Apr. 2002.
[4] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.
[5] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, p. 066111, 2004.
[6] 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, p. P10008, Oct. 2008.
[7] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, pp. 814–818, Jun. 2005.
[8] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, Feb. 2007.
[9] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.
[10] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–426, Dec. 2007.
[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[12] B. Yang, J. Liu, and J. Feng, "On the spectral characterization and scalable mining of network communities," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 2, pp. 326–337, Feb. 2012.
[13] J. Yang, J. McAuley, and J. Leskovec, "Detecting cohesive and 2-mode communities indirected and undirected networks," in *Proc. WSDM*, 2014, pp. 323–332.
[14] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Sep. 2008.
[15] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endowment*, vol. 21, no. 1, pp. 718–729, 2009.
[16] Y. Zhou, H. Cheng, and J. X. Yu, "Clustering large attributed graphs: An efficient incremental approach," in *Proc. ICDM*, Dec. 2010, pp. 689–698.
[17] S. Gunnermann, B. Boden, I. Farber, and T. Seidl, "Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors," in *Proc. PAKDD*, 2013, pp. 261–275.
[18] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "GBAGC: A general Bayesian framework for attributed graph clustering," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 1, 2014, Art. no. 5.
[19] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. ICDM*, Dec. 2013, pp. 1151–1156.
[20] J. McAuley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 1, 2014, Art. no. 4.
[21] P. Hu, K. C. C. Chan, T. He, and H. Leung, "Deep fusion of multiple social networks for learning latent social communities," in *Proc. 29th Int. Conf. Tools Artif. Intell.*, Boston, MA, USA, Nov. 2017.

[22] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.
[23] R. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proc. SIGKDD*, 2008, pp. 542–550.
[24] J. Chang and D. Blei, "Relational topic models for document networks," in *Proc. AISTATS*, 2009, pp. 81–88.
[25] Y. Sun, J. Han, J. Gao, and Y. Yu, "iTopicModel: Information network-integrated topic modeling," in *Proc. ICDM*, Dec. 2009, pp. 493–502.
[26] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: A discriminative approach," in *Proc. SIGKDD*, 2009, pp. 927–936.
[27] R. Balasubramanyan and W. W. Cohen, "Block-LDA: jointly modeling entity-annotated text and entity-entity links," in *Proc. SDM*, 2011, pp. 450–461.
[28] T. He and K. C. C. Chan, "MISAGA: An algorithm for mining interesting subgraphs in attributed graphs," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1369–1382, May 2018.
[29] L. Hu and K. C. C. Chan, "Fuzzy clustering in a complex network based on content relevance and link structures," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 2, pp. 456–470, Apr. 2016.
[30] T. He and K. C. C. Chan, "Discovering fuzzy structural patterns for graph analytics," *IEEE Trans. Fuzzy Syst.*, to be published, doi: 10.1109/TFUZZ.2018.2791951.
[31] T. He and K. C. C. Chan, "Evolutionary community detection in social networks," in *Proc. CEC*, Jul. 2014, pp. 1496–1503.
[32] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge, U.K.: Cambridge Univ. Press, 2003.
[33] M. Frank, A. P. Streich, D. Basin, and J. M. Buchmann, "Multi-assignment clustering for Boolean data," *J. Mach. Learn. Res.*, vol. 13, no. 3, pp. 459–489, 2012.
[34] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization," in *Proc. SIGMOD*, 2008, pp. 567–580.
[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.
[36] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter, "Comparing community structure to characteristics in online collegiate social networks," *SIAM Rev.*, vol. 53, no. 3, pp. 526–543, Aug. 2011.
[37] G.-J. Qi, C. C. Aggarwal, and T. Huang, "Community detection with edge content in social media networks," in *Proc. ICDE*, Apr. 2012, pp. 534–545.
[38] 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, 2017, doi: 10.1109/ACCESS.2017.2783542.
[39] T. He and K. C. C. Chan, "Evolutionary graph clustering for protein complex identification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 3, pp. 892–904, May/Jun. 2018.



TIANTIAN HE received the B.Eng. degree in computer science and technology from the North China University of Technology, Beijing, China, in 2008, and the M.Sc. degree in information systems and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2012 and 2017, respectively.

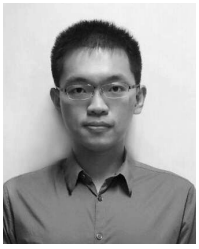
He is currently a Post-Doctoral Research Assistant with the Department of Computing, The Hong Kong Polytechnic University. His research interests include machine learning, data mining, and bioinformatics.

LUN HU received the B.Eng. degree from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2006, and the M.Sc. and Ph.D. degrees from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2008 and 2015, respectively.

He is currently an Assistant Professor with the School of Computer Science and Technology, Wuhan University of Technology, Wuhan. His research interests include data mining algorithms and applications to graph clustering and bioinformatics.

KEITH C. C. CHAN received the B.Math. degree (Hons.) in computer science and statistics, and the M.A.Sc. and Ph.D. degrees in systems design engineering from the University of Waterloo, ON, Canada, in 1984, 1985, and 1989, respectively.

He was a Software Analyst for the development of multimedia and software engineering tools at the IBM Canada Laboratory, Toronto, Canada. He joined The Hong Kong Polytechnic University in 1994, where he is currently a Professor with the Department of Computing. His research is supported by both government research funding agencies and the industry. His research interests include bioinformatics, data mining, and software engineering. He has over 200 publications in these areas. He serves on the editorial board of five journals and has been serving on the program committees of numerous conferences.



PENGWEI HU received the B.Eng. degree from the Foreign Trade and Business College, Chongqing Normal University, China, in 2011.

He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. His current research interests mainly focus on data mining algorithms and applications to social network and bioinformatics.