Multityped Community Discovery in Time-Evolving Heterogeneous Information Networks Based on Tensor Decomposition

Wu, Jibing; Yu, Lianfei; Zhang, Qun; Shi, Peiteng; Liu, Lihua; Deng, Su; Huang, Hongbin

DOI
10.1155/2018/9653404

Publication date
2018

Document Version
Final published version

Published in
Complexity

Hindawi
Complexity
Volume 2018, Article ID 9653404, 16 pages
https://doi.org/10.1155/2018/9653404

Research Article

Multityped Community Discovery in Time-Evolving Heterogeneous Information Networks Based on Tensor Decomposition

Jibing Wu,[1] **Lianfei Yu**,[1,2] **Qun Zhang**,[1] **Peiteng Shi**,[3] **Lihua Liu**,[1] **Su Deng**,[1] **and Hongbin Huang**[1]

[1]*Science and Technology on Information System Engineering Laboratory, National University of Defense Technology, Changsha, China*
[2]*Army Academy of Border and Coastal Defense, Urumqi, China*
[3]*Department of Intelligent Systems, Delft University of Technology, Delft, Netherlands*

Correspondence should be addressed to Hongbin Huang; hbhuang@nudt.edu.cn

Received 29 August 2017; Revised 15 January 2018; Accepted 31 January 2018; Published 6 March 2018

Academic Editor: Manlio De Domenico

The heterogeneous information networks are omnipresent in real-world applications, which consist of multiple types of objects with various rich semantic meaningful links among them. Community discovery is an effective method to extract the hidden structures in networks. Usually, heterogeneous information networks are time-evolving, whose objects and links are dynamic and varying gradually. In such time-evolving heterogeneous information networks, community discovery is a challenging topic and quite more difficult than that in traditional static homogeneous information networks. In contrast to communities in traditional approaches, which only contain one type of objects and links, communities in heterogeneous information networks contain multiple types of dynamic objects and links. Recently, some studies focus on dynamic heterogeneous information networks and achieve some satisfactory results. However, they assume that heterogeneous information networks usually follow some simple schemas, such as bityped network and star network schema. In this paper, we propose a multityped community discovery method for time-evolving heterogeneous information networks with general network schemas. A tensor decomposition framework, which integrates tensor CP factorization with a temporal evolution regularization term, is designed to model the multityped communities and address their evolution. Experimental results on both synthetic and real-world datasets demonstrate the efficiency of our framework.

1. Introduction

Most artificial online systems, such as World Wide Web, social networks, and collaboration networks, can be represented as information networks, which describe the interactions and relationships between numerous objects, for example, hyperlinks between web pages, friendships between users, and coauthorships between researchers. The information network analysis is attracting an increasing number of researchers from a variety of fields, such as social science [1, 2], machine learning [3–5], and recommendation systems [6, 7]. Community discovery is one of the most significant focuses in information network analysis, which aims to discover interpretable hidden structures, patterns of interactions among objects, and their evolution along with time in such network. Although community detection in networks has been studied for many years, most existing approaches are designed to analyze static information network [1, 8, 9] and homogeneous information network [10–12]. That is, there is only one type of objects and links contained in the network, and the objects and links are not time-varying.

However, in real-world scenarios, information networks are typically heterogeneous and time-evolving. In contrast to communities in traditional approaches, which only contain one type of static objects and links, communities in time-evolving heterogeneous information networks contain multiple types of dynamic objects and links. For example, the DBLP network, an open resource including most bibliographic information on computer science, is a typical time-evolving heterogeneous information network. DBLP

network contains four types of objects: author (A), paper (P), venue (i.e., conference or journal) (V), and term (T). The links between different object types represent different semantic relationships, such as "an author wrote a paper" and "a paper published in a conference." The most intriguing communities in DBLP are research areas, which contain the authors with similar research interests, the papers they wrote, the conferences they attended, and the terms they used. With the addition of new authors and new hot topics, the structures of communities are dynamic and varying gradually.

Although the traditional community discovery methods can be applied to time-evolving heterogeneous information network by converting such network into a set of homogeneous information networks and aggregating the time-evolving objects and links along with all timestamps into one snapshot, the rich semantic relationships among different object types and the dynamic property of the communities are lost. In recent years, community discovery in time-evolving heterogeneous information networks has emerged as an outstanding challenge and attracted the attention of many researchers. For instance, Sun et al. used net-clusters [13] to describe the communities and proposed a Dirichlet Process Mixture Model based algorithm named Evo-NetClus [14, 15] to detect the communities in heterogeneous information networks with star network schema. In the star network schema, the links only appear between target objects and attribute objects.

In this paper, we focus on community discovery in time-evolving heterogeneous information networks with general network schemas, which presents several challenges as follows:

(i) Heterogeneity: obviously, the communities in heterogeneous information networks are also heterogeneous, which contain multityped objects and links.

(ii) Time-varying: the communities are constantly changing, with new objects coming and old objects vanishing. We assume that the evolution of communities at two adjacent snapshots should be smooth.

(iii) Being suitable for general network schema: the network schema of a heterogeneous information network is often more complex than star network schema. The community discovery method should be able to handle the general network schema.

(iv) Online mode: although some offline frameworks can produce a global view of community evolution along time by capturing all historical information, online framework is more realistic.

To overcome the aforementioned challenges, we propose a tensor decomposition framework for modeling the multityped communities and address their evolution in time-evolving heterogeneous information networks with general network schemas. Essentially, a time-evolving heterogeneous information network consists of a sequence of network snapshots. We model the time-evolving heterogeneous information network as a sequence of multiway arrays, that is, tensors. Tensor is a highly effective and veracious approach for modeling high-mode data, which can naturally express the complex structures and interactions in heterogeneous information networks. By integrating the tensor CP factorization with a temporal evolution regularization term, the multityped communities and their evolution along time can be formalized as a tensor decomposition problem. A second-order stochastic gradient descent algorithm is presented to solve the problem, and the experimental results on both synthetic and real-world datasets demonstrate the efficiency of our framework.

The rest of this paper is organized as follows. In Section 2, we discuss the related work on community discovery in time-evolving heterogeneous information networks. Section 3 formalizes the problem as tensor decomposition, which integrates tensor CP factorization with a temporal evolution regularization term. A second-order stochastic gradient descent algorithm is presented in Section 4. Section 5 discusses some implementation issues, including dead and new objects, online deployment, and time complexity analysis. The experimental results on both synthetic and real-world datasets are presented in Section 6. Finally, the conclusions are drawn in Section 7.

2. Related Work

Community discovery is a fundamental technique of information network analysis. Many creative methods for discovering communities in static and homogeneous network have been deployed in the past decades. Stochastic block model [16, 17] and mixed membership model [18] are powerful probabilistic community discovery models for analyzing static networks. These two models, however, lack capability of time-evolving networks and cannot be directly used for heterogeneous information networks.

Tracking the evolution of communities [11, 19] takes the dynamic properties in time-evolving networks into consideration. A commonly used framework [20–22] is to apply the static community detection algorithms for each snapshot of the time-evolving networks and then generate the evolution of communities by computing the match between two adjacent snapshots. Another attempt to track community evolution in time-evolving networks is multiobjective optimization model [23–25], which integrates the measurement of community quality and temporal smoothness into a multiobjective cost function. Nevertheless, these methods are designed for homogeneous networks.

Recently, the community discovery in heterogeneous information networks has become a hot topic. Tang et al. introduced the community evolution in multimode network and proposed a framework which partitioned the multimode network into a set of bityped networks [26, 27]. Sun et al. used net-clusters [13] to describe the communities and proposed Evo-NetClus [14, 15] to detect the communities automatically. However, the net-clusters and Evo-NetClus are only suitable for star network schema, where the links only appear between target objects and attribute objects.

To analyze the heterogeneous information networks with general network schemas, tensor factorization offers a promising way for extracting hidden communities in such networks. Tensor is an effective expression of complicated

and interpretable structures among different dimensions in heterogeneous information network. For instance, Lin et al. proposed MetaGraph Factorization [28, 29] to detect the communities from dynamic social networks. In addition, a tensor factorization based mixed membership framework [30] simulates the generation of communities as Dirichlet distribution, which can identify the communities automatically. However, this method needs to partition the heterogeneous network into four parts artificially and organize them as a 3-star network. Meanwhile, the 3-star count tensor must be converted to an orthogonal symmetric tensor. Thus the capability of this method to deal with time-evolving heterogeneous information networks could be degraded.

Our prior works in [31–33] have also focused on clustering heterogeneous information networks based on tensor decomposition, which can cluster multityped objects simultaneously in heterogeneous information networks. However, these methods treat the heterogeneous information networks as static networks and integrate the time-evolving networks into one snapshot, which lose the dynamic properties among multityped objects and links.

Another line related to our work is on the incremental tensor factorization [34]. Though tensor factorization has been widely studied in many domains, such as image processing [35] and computer vision [36], the incremental tensor factorization is still a challenging intellectual task [34]. Sun et al. proposed a general framework of incremental tensor analysis [34] for mining higher-order data streaming, which included three methods: dynamic tensor analysis, streaming tensor analysis, and window-based tensor analysis. Even though the higher-order data streaming can be effectively analyzed in such framework, the smooth evolution of latent patterns cannot be guaranteed.

3. Problem Formulation

Following the works by Sun et al. in [15] and our prior work [33], we first introduce some definitions of heterogeneous information networks and tensor construction from a given heterogeneous information network.

A *heterogeneous information network* [15] is a graph $G = (V, E)$ consisting of more than one type of objects V or links E. Assume that V belongs to N object types $\mathbb{V} = \{\mathcal{V}^{(n)}\}_{n=1}^{N}$, and E belongs to M link types $\mathbb{E} = \{\mathcal{R}^{(m)}\}_{m=1}^{M}$. That is, in a heterogeneous information network, $N > 1$ or $M > 1$. Otherwise, the network becomes a homogeneous information network.

The $\mathcal{V}^{(n)}$ indicates the set of objects from the nth type. We denote an arbitrary object in $\mathcal{V}^{(n)}$ as $v_{i_n}^{(n)}$, for $i_n = 1, 2, \ldots, I_n$; $n = 1, 2, \ldots, N$, where I_n is the number of objects in type $\mathcal{V}^{(n)}$; that is, $I_n = |\mathcal{V}^{(n)}|$. Thus, the total number of objects in the heterogeneous information network G is given by $I = \sum_{n=1}^{N} I_n$.

The *network schema* [15] for a given heterogeneous information network $G = (V, E)$ is a metatemplate that indicates the formation of object types $\mathbb{V}$ and link types $\mathbb{E}$ in the network. The network schema is denoted by $S_G = \{\mathbb{V}, \mathbb{E}\}$. In other words, $G = (V, E)$ is an instance of $S_G = \{\mathbb{V}, \mathbb{E}\}$.

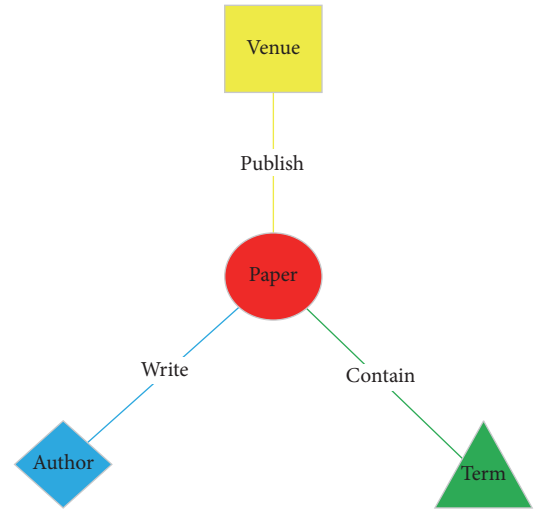

Figure 1: A typical star network schema extracted from DBLP network.

For example, the star network schema shown in Figure 1 is a typical network schema, in which four types of objects are contained, that is, author, paper, venue, and term. In Figure 1, paper is target object, and the others are attribute objects. The feature of star network schema is that the links in the network only appear between target object and attribute objects.

A *gene-network* [33], denoted by ϕ, is a minimum instance of $S_G = \{\mathbb{V}, \mathbb{E}\}$ in the set of subnetworks of $G = (V, E)$. It is noteworthy that a gene-network is an integrated semantic relation in the heterogeneous information network, which is quite different from gene regulatory network in Bioinformatics [37]. For example, a gene-network in DBLP network, denoted by $\phi = (\{v_i^{(A)}, v_j^{(P)}, v_l^{(V)}, v_m^{(T)}\}, \{\langle v_i^{(A)}, v_j^{(P)}\rangle, \langle v_j^{(P)}, v_l^{(V)}\rangle, \langle v_j^{(P)}, v_m^{(T)}\rangle\})$, represents an integrated semantic relation; that is, "an author $v_i^{(A)}$ writes a paper $v_j^{(P)}$, which contains the term $v_m^{(T)}$ and is published in the venue $v_l^{(V)}$." For simplicity, we can mark the gene-network ϕ by the subscripts of objects in ϕ, that is, $\phi_{i,j,l,m}$.

Following our prior work [33], a Nth order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be constructed according to the distribution of gene-networks, where each mode of $\mathcal{X}$ represents one type of objects in the network G. An arbitrary element $x_{i_1 i_2 \cdots i_N} \in \{0, 1\}$ is an indicator of whether the corresponding gene-network $\phi_{i_1, i_2, \ldots, i_N}$ exists, where $i_n = 1, 2, \ldots, I_n$, for $n = 1, 2, \ldots, N$, is the index of an object in type $\mathcal{V}^{(n)}$.

$$x_{i_1 i_2 \cdots i_N} = \begin{cases} 1, & \text{if } \exists \phi_{i_1, i_2, \ldots, i_N}; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The time-evolving heterogeneous information networks can be segmented into a *network sequence* according to a series of snapshots. The heterogeneous information network associated with timestamp t can be denoted as $G^{(t)} = (V^{(t)}, E^{(t)})$; then the network sequence is $\mathcal{GS} = (G^{(1)}, G^{(2)}, \ldots, G^{(t)}, \ldots)$. Thereby, the tensor representation

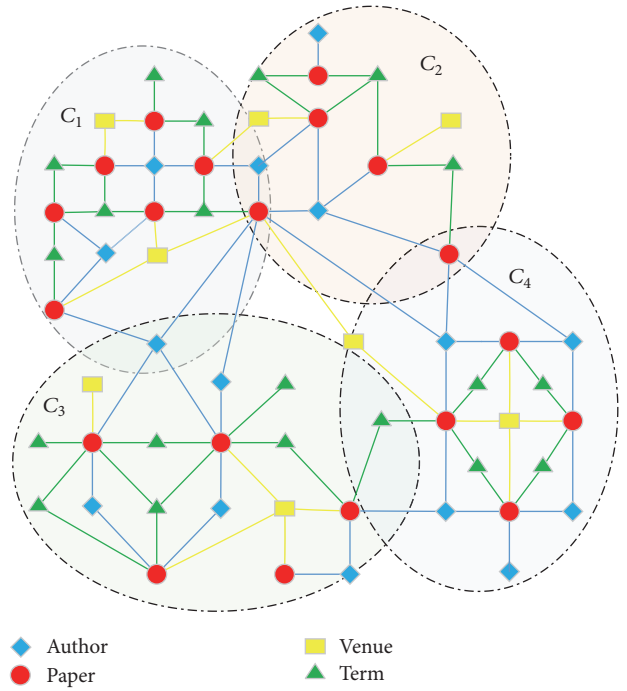

Figure 2: An instance of multityped communities in DBLP network.

of the network sequence is $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(t)}, \ldots$. Actually, $\mathcal{X}^{(t)} \in \{0,1\}^{I_1 \times I_2 \times \cdots \times I_N}$ is the hyper-adjacency tensor of the given heterogeneous information network at the tth timestamp, which indicates the distribution of gene-networks.

The community in heterogeneous information network is called *multityped community*, which is more complex than that in homogeneous information network. A multityped community is a set of gene-networks that share the same features and connect together. In other words, a multityped community contains all associated types of objects and links. As shown in Figure 2, the multityped communities about research areas in DBLP network consist of the authors with similar research interests, the papers they wrote, the conferences they attended, and the terms they used. In each multityped community, the authors, papers, venues, and terms are connected to each other and organized as gene-networks. In fact, the objects may belong to several multityped communities since some gene-networks coming from different multityped communities may share the same objects. For example, a famous scientist can cooperate with other researchers within different areas by publishing many interdisciplinary papers; that is, the famous scientist will be contained in many gene-networks across different multityped communities.

The problem of multityped community discovery from such a network sequence can be decomposed into two subproblems: (A) detect the multityped communities in each network snapshot, and (B) model the evolution of multityped communities over time.

(A) Multityped Community Discovery in Each Network Snapshot. Without loss of generality, we take the tth network snapshot $G^{(t)}$ as an example. Let $\{\mathscr{C}_k^{(t)}\}_{k=1}^K$ denote K hidden multityped communities in the network $G^{(t)}$ and $u_{i_n,k}^{(n,t)}$ represent the probability that the i_nth object in type $\mathscr{V}^{(n)}$ belongs to the kth community at the tth timestamp. Denote

$$\mathbf{u}_k^{(n,t)} = \left[u_{1,k}^{(n,t)}, u_{2,k}^{(n,t)}, \ldots, u_{I_n,k}^{(n,t)}\right]^\top \in \mathbb{R}^{I_n}. \tag{2}$$

Following our prior work [33], a multityped community can be represented as

$$\mathscr{C}_k^{(t)} = \mathbf{u}_k^{(1,t)} \circ \mathbf{u}_k^{(2,t)} \circ \cdots \circ \mathbf{u}_k^{(N,t)}, \tag{3}$$

where $\circ$ is the outer product of two vectors. Actually, the multityped community $\mathscr{C}_k^{(t)}$ is a rank-one tensor with the same size of $\mathcal{X}^{(t)}$. Equation (3) indicates the gene-networks and the probability of associated objects belonging to the kth community. Thereby, we can approximate $\mathcal{X}^{(t)}$ through a sum of K rank-one tensors; that is,

$$\mathcal{X}^{(t)} \approx \sum_{k=1}^{K} \mathscr{C}_k^{(t)} = \sum_{k=1}^{K} \mathbf{u}_k^{(1,t)} \circ \mathbf{u}_k^{(2,t)} \circ \cdots \circ \mathbf{u}_k^{(N,t)}. \tag{4}$$

Obviously, (4) is a tensor CP factorization. Let factor matrix $\mathbf{U}^{(n,t)} = [\mathbf{u}_1^{(n,t)}, \mathbf{u}_2^{(n,t)}, \ldots, \mathbf{u}_K^{(n,t)}] \in \mathbb{R}^{I_n \times K}$ be the latent community membership matrix for the nth type of objects at timestamp t, where $n = 1, 2, \ldots, N$. We denote

$$\begin{aligned} &\left[\!\left[\mathbf{U}^{(1,t)}, \mathbf{U}^{(2,t)}, \ldots, \mathbf{U}^{(N,t)}\right]\!\right] \\ &\quad \equiv \sum_{k=1}^{K} \mathbf{u}_k^{(1,t)} \circ \mathbf{u}_k^{(2,t)} \circ \cdots \circ \mathbf{u}_k^{(N,t)}. \end{aligned} \tag{5}$$

By minimizing the Frobenius norm of the difference between $\mathcal{X}^{(t)}$ and its CP approximation, the multityped community discovery in each network snapshot can be formulated as an optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathcal{X}^{(t)} - \left[\!\left[\mathbf{U}^{(1,t)}, \mathbf{U}^{(2,t)}, \ldots, \mathbf{U}^{(N,t)}\right]\!\right]\right\|_F^2, \\ \text{s.t.} \quad & \sum_{k=1}^{K} u_{i_n,k}^{(n,t)} = 1, \quad \forall n, \ \forall i_n, \\ & u_{i_n,k}^{(n,t)} \in [0,1], \quad \forall n, \ \forall i_n, \ \forall k, \\ & \sum_{i_n=1}^{I_n} u_{i_n,k}^{(n,t)} > 0, \quad \forall n, \ \forall k, \end{aligned} \tag{6}$$

where $i_n = 1, 2, \ldots, I_n$; $n = 1, 2, \ldots, N$; and $k = 1, 2, \ldots, K$. The first and second constraints in (6) guarantee that $u_{i_n,k}^{(n,t)}$ is the probability. The last constraint in (6) ensures that each multityped community consists of all associated types of objects.

(B) Multityped Community Evolution over Time. Equation (6) just performs the multityped community discovery at each timestamp independently and does not consider their

smooth evolution at two adjacent snapshots. We denote the objective function in (6) as $f^{(t)}$; that is,

$$f^{(t)} = \frac{1}{2}\left\|\mathcal{X}^{(t)} - \left[\!\left[\mathbf{U}^{(1,t)}, \mathbf{U}^{(2,t)}, \ldots, \mathbf{U}^{(N,t)}\right]\!\right]\right\|_F^2. \quad (7)$$

In order to ensure that the evolution of the multityped communities is smooth, a temporal evolution regularization term $g^{(t)}$ is introduced.

$$g^{(t)} = \frac{\lambda}{2}\sum_{n=1}^{N}\left\|\mathbf{U}^{(n,t)} - \mathbf{U}^{(n,t-1)}\right\|_F^2, \quad (8)$$

where $\lambda > 0$ is a temporally regularized parameter. Indeed, $g^{(t)}$ is a first-order Markov assumption, which forces the multityped communities at current timestamp to resemble that at previous snapshot.

Denote the objective function as

$$\begin{aligned}\mathcal{L}^{(t)} &= f^{(t)} + g^{(t)} \\ &= \frac{1}{2}\left\|\mathcal{X}^{(t)} - \left[\!\left[\mathbf{U}^{(1,t)}, \mathbf{U}^{(2,t)}, \ldots, \mathbf{U}^{(N,t)}\right]\!\right]\right\|_F^2 \\ &\quad + \frac{\lambda}{2}\sum_{n=1}^{N}\left\|\mathbf{U}^{(n,t)} - \mathbf{U}^{(n,t-1)}\right\|_F^2.\end{aligned} \quad (9)$$

Therefore, the problem of multityped community discovery in time-evolving heterogeneous information networks can be formulated as

$$\begin{aligned}\min_{\mathbf{U}^{(1,t)},\mathbf{U}^{(2,t)},\ldots,\mathbf{U}^{(N,t)}} \quad & \mathcal{L}^{(t)} \\ \text{s.t.} \quad & \sum_{k=1}^{K} u_{i_n,k}^{(n,t)} = 1, \quad \forall n,\ \forall i_n, \\ & u_{i_n,k}^{(n,t)} \in [0,1], \quad \forall n,\ \forall i_n,\ \forall k, \\ & \sum_{i_n=1}^{I_n} u_{i_n,k}^{(n,t)} > 0, \quad \forall n,\ \forall k.\end{aligned} \quad (10)$$

Here, $\{\mathbf{U}^{(n,t-1)}\}_{n=1}^{N}$ are constants at current timestamp t, which are solved at previous timestamp. When $t = 1$, we have no a priori knowledge about the multityped communities. We set $\mathbf{U}^{(n,t=0)} = \mathbf{0}$, for $n = 1, 2, \ldots, N$. Thus, $g^{(t=1)}$ becomes

$$g^{(t=1)} = \frac{\lambda}{2}\sum_{n=1}^{N}\left\|\mathbf{U}^{(n,t)}\right\|_F^2. \quad (11)$$

It is worth noting that $g^{(t=1)}$ is also a Tikhonov regularization term [38], which ensures the sparsity of the factor matrices and makes the optimization solution easy to be found. Moreover, when $t = 1$, problem (10) degrades into the same form as we proposed in [33]. That is, the work in [33] is the special case for static networks.

4. Algorithm

The stochastic gradient descent algorithm is an efficient tool for optimizing tensor factorization [33, 39]. However, the first-order stochastic gradient descent algorithm has a poor convergence speed near the optimal point. It has been proven that the second-order stochastic algorithm has not only a faster convergence speed but also better robustness with respect to the learning rate [33]. The SOSClus proposed in [33] is a second-order stochastic algorithm which has been well studied for the case of $t = 1$ in (10), that is, static heterogeneous information networks. Here, we present a second-order stochastic gradient descent algorithm, named SOSComm, for the time-evolving case, which is an extension of SOSClus. In this section, some multilinear operators and tensor algebra for tensor factorization will be used, which can be found in [40].

When $t > 1$, the snapshot of the current heterogeneous information network $\mathcal{X}^{(t)}$ and the previous community membership matrices $\{\mathbf{U}^{(n,t-1)}\}_{n=1}^{N}$ are known. To compute the factor matrix $\mathbf{U}^{(n,t)}$, we can rewrite $\mathcal{L}^{(t)}$ in (10) by matricization of $\mathcal{X}^{(t)}$ along the nth mode. According to (7), (8), and (9), we have

$$\mathcal{L}_{(n)}^{(t)} = f_{(n)}^{(t)} + g_{(n)}^{(t)}, \quad (12)$$

where

$$\begin{aligned} f_{(n)}^{(t)} &= \frac{1}{2}\left\|\mathcal{X}_{(n)}^{(t)} - \mathbf{U}^{(n,t)}\left(\odot^{(/n)}\mathbf{U}\right)^{\top}\right\|_F^2, \\ g_{(n)}^{(t)} &= g^{(t)}. \end{aligned} \quad (13)$$

The $\mathcal{X}_{(n)}^{(t)} \in \mathbb{R}^{I_n \times \prod_{m=1, m\neq n}^{N} I_m}$ is the matricization of $\mathcal{X}^{(t)}$ along the nth mode, and the symbol $\odot$ indicates the Khatri-Rao product of two matrices. Given two matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$, their Khatri-Rao product is a matrix of size $IJ \times K$ and defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{b}_{:,1} & a_{12}\mathbf{b}_{:,2} & \cdots & a_{1K}\mathbf{b}_{:,K} \\ a_{21}\mathbf{b}_{:,1} & a_{22}\mathbf{b}_{:,2} & \cdots & a_{2K}\mathbf{b}_{:,K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{b}_{:,1} & a_{I2}\mathbf{b}_{:,2} & \cdots & a_{IK}\mathbf{b}_{:,K} \end{bmatrix}, \quad (14)$$

where a_{ik}, $i = 1, 2, \ldots, I$, $k = 1, 2, \ldots, K$, is an element of $\mathbf{A}$, and $\mathbf{b}_{:,k} \in \mathbb{R}^J$, $k = k = 1, 2, \ldots, K$, is a column of $\mathbf{B}$. In particular, we denote the Khatri-Rao product of a series of matrices except $\mathbf{U}^{(n,t)}$ as

$$\odot^{(/n)}\mathbf{U} = \mathbf{U}^{(N,t)} \odot \cdots \odot \mathbf{U}^{(n+1,t)} \odot \mathbf{U}^{(n-1,t)} \odot \cdots \odot \mathbf{U}^{(1,t)}. \quad (15)$$

Since the partial derivative of $f_{(n)}^{(t)}$ with respect to $\mathbf{U}^{(n,t)}$ has been given in [33], we introduce the result directly.

$$\frac{\partial f_{(n)}^{(t)}}{\partial \mathbf{U}^{(n,t)}} = -\mathcal{X}_{(n)}^{(t)}\left(\odot^{(/n)}\mathbf{U}\right) + \mathbf{U}^{(n,t)}\Gamma^{(n,t)}, \quad (16)$$

where

$$\begin{aligned}\Gamma^{(n,t)} &\equiv \left(\odot^{(/n)}\mathbf{U}\right)^{\top}\left(\odot^{(/n)}\mathbf{U}\right)\\ &= \left(\left(\mathbf{U}^{(1,t)}\right)^{\top}\mathbf{U}^{(1,t)}\right) * \cdots * \left(\left(\mathbf{U}^{(n-1,t)}\right)^{\top}\mathbf{U}^{(n-1,t)}\right)\\ &\quad * \left(\left(\mathbf{U}^{(n+1,t)}\right)^{\top}\mathbf{U}^{(n+1)}\right) * \cdots\\ &\quad * \left(\left(\mathbf{U}^{(N,t)}\right)^{\top}\mathbf{U}^{(N,t)}\right),\end{aligned} \tag{17}$$

and symbol $*$ is Hadamard product, also named element-wise product of two matrices with the same dimension.

The partial derivative of $g_{(n)}^{(t)}$ with respect to $\mathbf{U}^{(n,t)}$ is

$$\begin{aligned}\frac{\partial g_{(n)}^{(t)}}{\partial \mathbf{U}^{(n,t)}} &= \frac{\lambda\partial\left\|\mathbf{U}^{(n,t)}-\mathbf{U}^{(n,t-1)}\right\|_F^2}{2\partial\mathbf{U}^{(n,t)}}\\ &= \frac{\lambda\partial \operatorname{Tr}\left(\left(\mathbf{U}^{(n,t)}-\mathbf{U}^{(n,t-1)}\right)\left(\mathbf{U}^{(n,t)}-\mathbf{U}^{(n,t-1)}\right)^{\top}\right)}{2\partial\mathbf{U}^{(n,t)}}\\ &= \frac{\lambda\partial \operatorname{Tr}\left(\mathbf{U}^{(n,t)}\left(\mathbf{U}^{(n,t)}\right)^{\top}\right)}{2\partial\mathbf{U}^{(n,t)}}\\ &\quad - \frac{\lambda\partial \operatorname{Tr}\left(\mathbf{U}^{(n,t)}\left(\mathbf{U}^{(n,t-1)}\right)^{\top}\right)}{\partial\mathbf{U}^{(n,t)}}\\ &\quad + \frac{\lambda\partial \operatorname{Tr}\left(\mathbf{U}^{(n,t-1)}\left(\mathbf{U}^{(n,t-1)}\right)^{\top}\right)}{2\partial\mathbf{U}^{(n,t)}}\\ &= \lambda\left(\mathbf{U}^{(n,t)}-\mathbf{U}^{(n,t-1)}\right).\end{aligned} \tag{18}$$

Therefore, the partial derivative of $\mathscr{L}_{(n)}^{(t)}$ with respect to $\mathbf{U}^{(n,t)}$ is given by

$$\begin{aligned}\frac{\partial\mathscr{L}_{(n)}^{(t)}}{\partial\mathbf{U}^{(n,t)}} &= -\mathscr{X}_{(n)}^{(t)}\left(\odot^{(/n)}\mathbf{U}\right) + \mathbf{U}^{(n,t)}\left(\Gamma^{(n,t)}+\lambda\mathbf{I}\right)\\ &\quad - \lambda\mathbf{U}^{(n,t-1)},\end{aligned} \tag{19}$$

where $\mathbf{I}$ is a unit matrix. And the second-order partial derivative of $\mathscr{L}_{(n)}^{(t)}$ with respect to $\mathbf{U}^{(n,t)}$ can be obtained as

$$\frac{\partial^2\mathscr{L}_{(n)}^{(t)}}{\partial^2\mathbf{U}^{(n,t)}} = \Gamma^{(n,t)} + \lambda\mathbf{I}. \tag{20}$$

Recalling the update rule of the second-order stochastic algorithm [33, 41], we have

$$\begin{aligned}\mathbf{U}^{(n,t)} &\longleftarrow \mathbf{U}^{(n,t)} - \eta\left(\frac{\partial^2\mathscr{L}_{(n)}^{(t)}}{\partial^2\mathbf{U}^{(n,t)}}\right)^{-1}\frac{\partial\mathscr{L}_{(n)}^{(t)}}{\partial\mathbf{U}^{(n,t)}}\\ &= \eta\left(\mathscr{X}_{(n)}^{(t)}\left(\odot^{(/n)}\mathbf{U}\right)+\lambda\mathbf{U}^{(n,t-1)}\right)\left(\Gamma^{(n,t)}+\lambda\mathbf{I}\right)^{-1}\\ &\quad + (1-\eta)\,\mathbf{U}^{(n,t)},\end{aligned} \tag{21}$$

where η is named learning rate or step size with a positive number.

When $t = 1$, (21) has the same form as SOSClus. That is, the SOSComm is an extension of SOSClus for time-evolving heterogeneous information networks. To satisfy the constraints in (10), the factor matrices derived by (21) should be normalized as

$$u_{i_n,k}^{(n,t)} \longleftarrow \frac{u_{i_n,k}^{(n,t)}}{\sum_{k=1}^{K} u_{i_n,k}^{(n,t)}}. \tag{22}$$

For the current network $G^{(t)}$, based on the tensor representation $\mathscr{X}^{(t)}$ and the previous community membership matrices $\{\mathbf{U}^{(n,t-1)}\}_{n=1}^{N}$, the alternating optimization can be used to update $\mathbf{U}^{(n,t)}$ according to (21) and (22), while all other variables are fixed. The community membership matrices $\{\mathbf{U}^{(n,t)}\}_{n=1}^{N}$ obtained by (21) and (22) are the approximations. We also need to recover the discrete community membership matrices from the approximations in some cases, which can be achieved by applying K-means to the factor matrices. Conveniently, we can simply assign each object to the multityped community which has the largest entry in the corresponding row of factor matrix. After that, the multityped communities consist of gene-networks that can be extracted according to (3). Therefore, the pseudocode of SOSComm is given in Algorithm 1.

5. Implementation Issues

5.1. New Objects Coming and Old Objects Vanishing. In realistic scenarios, objects in time-evolving heterogeneous information networks have various lifecycles. With the lifecycles beginning and end, new objects are born and join the network while old objects die and leave. The framework designed above does not consider the various lifecycles of objects, which assumes that the objects in a network remain unchanged and keep active. Here, we discuss more realistic cases that new objects coming and old objects vanishing in a time-evolving heterogeneous information network.

Note that the tensor representation $\mathscr{X}^{(t)}$ is a distribution of gene-networks in the heterogeneous information network, whose elements indicate whether the gene-networks exist or not. If the lifecycle of a new object $v_{I_n+1}^{(n)}$ begins at the tth timestamp, it will join the network and become active. Since the size of $\mathbf{U}^{(n,t)}$ becomes $(I_n + 1) \times K$ and only the previous factor matrix $\mathbf{U}^{(n,t-1)}$ is used to regularize the temporal smoothness, we can add an all-zero row to the corresponding position on $\mathbf{U}^{(n,t-1)}$ when updating $\mathbf{U}^{(n,t)}$.

If the lifecycle of a specified object $v_{i_n}^{(n)}$ ends at the tth timestamp, it will not appear in any gene-network in the network. According to (1), $x_{:,\ldots,i_n,\ldots,:} = 0$. That is, each element in the hyperplane, which is perpendicular to the nth dimensionality and passes the i_nth point of the nth dimensionality in the tensor space, is zero. Therefore, we set all entries in the i_nth row of $\mathbf{U}^{(n,t)}$ equal to zero; that is, $u_{i_n,k}^{(n,t)} = 0$ for $k = 1, 2, \ldots, K$. However, this operation makes the

Input: the tensor representation of the current network $\mathcal{X}^{(t)}$, the number of multi-typed communities K, temporally regularization parameter λ, the community membership matrices for the previous network snapshot $\{\mathbf{U}^{(n,t-1)}\}_{n=1}^{N}$, and maximum iterations MaxIter.
Output: community membership matrices for the current network snapshot $\{\mathbf{U}^{(n,t)}\}_{n=1}^{N}$ and multi-typed communities $\{\mathscr{C}_k^{(t)}\}_{k=1}^{K}$.
(1) Set $\{\mathbf{U}^{(n,t)}\}_{n=1}^{N} \leftarrow \{\mathbf{U}^{(n,t-1)}\}_{n=1}^{N}$;
(2) Set iter $\leftarrow$ 1;
(3) **repeat**
(4) **for** $n \leftarrow 1$ **to** N **do**
(5) Set $\eta \leftarrow 1/(\text{iter} + 1)$;
(6) Update $\mathbf{U}^{(n,t)}$ according to (21);
(7) Normalize $\mathbf{U}^{(n,t)}$ according to (22);
(8) **end for**
(9) Set iter $\leftarrow$ iter + 1;
(10) **until** $\mathscr{L}^{(t)}$ unchanged or iter = MaxIter
(11) Recover the discrete community membership matrices from $\{\mathbf{U}^{(n,t)}\}_{n=1}^{N}$ (optional).
(12) Extract the multi-typed communities $\{\mathscr{C}_k^{(t)}\}_{k=1}^{K}$ according to (3).

Algorithm 1: SOSComm.

factor matrix $\mathbf{U}^{(n,t)}$ dissatisfy the first constraint in (10). Since our framework is an approximation and the dead objects will never appear in any multityped community (according to (3)), we can loosen the first constraint in (10) as

$$\sum_{k=1}^{K} u_{i_n,k}^{(n,t)} \leq 1, \quad \forall n, \ \forall i_n, \tag{23}$$

which does not affect the performance of recovering the discrete community membership matrices from $\{\mathbf{U}^{(n,t)}\}_{n=1}^{N}$ and extracting the multityped communities $\{\mathscr{C}_k^{(t)}\}_{k=1}^{K}$.

5.2. Online Deployment. The snapshots in the network sequence of time-evolving heterogeneous information networks are coming in a stream way, which makes the storage of the whole network sequence unrealistic. Fortunately, we only use the new network snapshot and the previous community membership matrices to update the model, which makes SOSComm easy to deploy online. However, three issues should be taken into account.

Firstly, the initialization of factor matrices has a large impact on the efficiency of SOSComm. A good initialization may reduce the number of iterations significantly. In practice, previous community membership matrices served as the start when updating the current factor matrices is a good choice. That is, set

$$\left\{\mathbf{U}^{(n,t)}\right\}_{n=1}^{N} \longleftarrow \left\{\mathbf{U}^{(n,t-1)}\right\}_{n=1}^{N}, \tag{24}$$

in the beginning of the algorithm. See line (1) in Algorithm 1.

Secondly, the second-order stochastic gradient descent algorithm has a fast convergence speed [33, 41] with good initialization, which will be proven in the experiments in Section 6. And the factor matrices obtained by SOSComm are the approximations to community membership matrices. Therefore, we can set the maximum iteration to be a very small positive integer.

Finally, the sparsity of heterogeneous information network should be used to speed up the calculation. According to (21), the primary computation cost for updating $\mathbf{U}^{(n,t)}$ is calculating a series of Khatri-Rao products, that is, $(\odot^{(/n)}\mathbf{U})$. If we store all the elements of $\mathcal{X}^{(t)}$ and calculate the Khatri-Rao product of the $N-1$ factor matrices orderly, it will be a very expensive calculation because the largest scale of intermediate results will reach $K \times \prod_{n=1}^{N} I_n$. Actually, the heterogeneous information networks are usually very sparse; namely, a great amount of elements in tensor $\mathcal{X}$ are zeros. By considering $\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U}) \in \mathbb{R}^{I_n \times K}$ as a whole, the elements of $\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U})$ are given by

$$\left(\mathcal{X}_{(n)}^{(t)}\left(\odot^{(/n)}\mathbf{U}\right)\right)_{i_n,k} = \sum_{\substack{\{i_m\}_{l=1}^{N} \\ l\neq n}} \left(x_{i_n,\prod_{\substack{l=1 \\ l\neq n}}^{N} i_l} \prod_{\substack{l=1 \\ l\neq n}}^{N} u_{i_l,k}^{(l,t)} \right). \tag{25}$$

Obviously, when $x_{i_n,\prod_{\substack{l=1 \\ l\neq n}}^{N} i_l} = 0$, we can directly set $(\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U}))_{i_n,k} = 0$; that is, the following calculation of Khatri-Rao products is unnecessary. Thus, by considering the sparsity, only nonzero elements in $\mathcal{X}$ need to be stored and calculated.

5.3. Time Complexity Analysis. The primary computation cost for updating the factor matrices in each iteration of SOSComm is calculating three part: $\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U})$, $(\Gamma^{(n,t)} + \lambda\mathbf{I})^{-1}$,

and the product of them. Firstly, for calculating $\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U})$, only nonzero elements in $\mathcal{X}$ need to be concerned. Therefore, the time complexity is $O(nnz(\mathcal{X})I_nK)$, where $nnz(\mathcal{X})$ is the number of nonzero elements in $\mathcal{X}$ and also is the total number of gene-networks in the network. Secondly, according to (17), since a series of matrix-matrix multiplications and Hadamard products are used to replace numerous Khatri-Rao products, calculating $\Gamma^{(n,t)}$ costs $O((I - I_n)K^2)$, where $I = \sum_n I_n$ is the total number of objects in the network. Thus, the time complexity for calculating the inverse matrix of $(\Gamma^{(n,t)} + \lambda\mathbf{I})$ is $O(((I - I_n)K^2) + K^3)$. Finally, the product of $\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U}) + \lambda\mathbf{U}^{(n,t-1)}$ and $(\Gamma^{(n,t)} + \lambda\mathbf{I})^{-1}$ is a matrix-matrix multiplication, where $(\mathcal{X}_{(n)}^{(t)}(\odot^{(/n)}\mathbf{U}) + \lambda\mathbf{U}^{(n,t-1)}) \in \mathbb{R}^{I_n \times K}$ and $((\Gamma^{(n,t)} + \lambda\mathbf{I})^{-1}) \in \mathbb{R}^{K \times K}$, so, the time complexity is $O(I_nK^2)$.

To summarize, the time complexity for SOSComm in each iteration is $O(nnz(\mathcal{X})IK + NIK^2 + NK^3)$, where $nnz(\mathcal{X})$ is the total number of gene-networks, I is the total number of objects, N is the number of object types, and K is the number of multityped communities. Since $K \ll I$ and $N \ll I$, the time complexity for SOSComm is nearly $O(nnz(\mathcal{X})I)$.

6. Experiments and Results

In this section, the proposed SOSComm is evaluated on both synthetic and real-world datasets. We demonstrate the efficiency of SOSComm for multityped community discovery in time-evolving heterogeneous information networks with general network schemas and further compare the performances with several other state-of-the-art community discovery methods. The experiments are simulated by MATLAB R2015a (version 8.5.0, 64-bit), with the MATLAB Tensor Toolbox (version 2.6, http://www.sandia.gov/~tgkolda/TensorToolbox/). The code and datasets used in experiments are available online https://github.com/tianshuideyu/SOSComm.

6.1. Experiments on Synthetic Datasets

6.1.1. Dataset Description. Typically, the real-world heterogeneous information networks are often without ground-truth of community membership. Furthermore, due to the large scale and sparsity, it is impossible to manually assign the community labels to objects in a real-world network. Therefore, several synthetic networks with detailed community structures are resorted to demonstrate the effectiveness of SOSComm.

We construct four synthetic networks with different parameters as the initial networks, that is, the network snapshots at $t = 1$. In order to obtain more realistic synthetic networks, the interactions between objects are assumed to follow Zipf's law (see details online: https://en.wikipedia.org/wiki/Zipf's_law), which denotes the distribution of gene-networks in networks. The parameters are as follows, and the details of the synthetic networks at the first timestamp are shown in Table 1:

(i) N is the number of object types in networks.

(ii) K is the number of multityped communities.

Table 1: The synthetic datasets.

Synthetic datasets	N	K	S	D
Syn1	2	2	$1M = 1000 \times 1000$	0.1%
Syn2	2	4	$10M = 1000 \times 10000$	0.01%
Syn3	4	2	$100M = 100 \times 100 \times 100 \times 100$	0.1%
Syn4	4	4	$1000M = 100 \times 100 \times 100 \times 1000$	0.01%

(iii) S is the network scale, and $S = I_1 \times I_2 \times \cdots \times I_N$.

(iv) D is the tensor density, and $D = nnz(\mathcal{X})/S$.

To simulate the smooth evolution of multityped communities, each synthetic network is evolved into a network sequence with 10 timestamps. Within each evolution, a percentage (from 5% to 10%) of the objects from each type change their community memberships by interacting with other objects in different communities randomly at each timestamp.

For completeness, we also randomly generate from 10% to 15% new objects coming and old objects vanishing in Syn4 at each timestamp. With new objects coming and interacting with other objects, many new gene-networks are generated. Meanwhile, with old objects vanishing, they will not appear in any gene-network in the network.

6.1.2. Comparative Methods and Experimental Setting. The performances of SOSComm on synthetic networks are compared with two state-of-the-art baselines:

(1) SOSClus (see [33]): an offline clustering framework for static heterogeneous information networks, which treats every snapshot in the network sequence independently without the temporal evolution regularization term.

(2) CEMNTR (see [26, 27]): a framework of community evolution in multimode network with temporal evolution regularization term, denoted as CEMNTR. CEMNTR partitions the multimode network into a set of bityped networks and detects communities in each bityped network via block model approximation with temporal regularization.

Both the baselines and SOSComm share the same stopping conditions; that is, the change of corresponding objective function is less than 10^{-6} and the maximum iterations MaxIter = 10000. The experiments in our prior work [33] have shown that the second-order stochastic gradient descent has good robustness with respect to the learning rate. Hence, we set the learning rate $\eta = 1/(\text{iter} + 1)$ for both SOSClus and SOSComm. As CEMNTR needs to partition the networks into a set of bityped networks, we divide each network snapshot in Syn3 and Syn4 into 3 bityped networks and construct the adjacent matrices for each pair of object types.

Since the ground-truth of the community structures in the synthetic networks is known, we adopt the Normalized

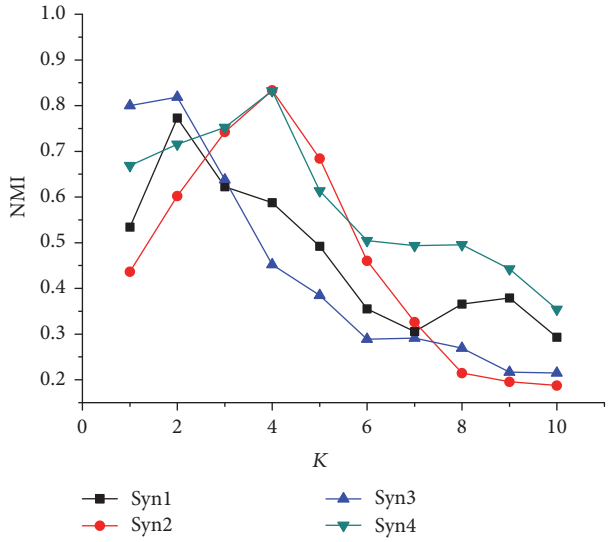

FIGURE 3: The performances of SOSComm on 4 synthetic networks with different K.

Mutual Information (NMI) [42] as the metric to evaluate the performances. NMI is a measurement of mutual dependence information between multityped community membership and the ground-truth, which ranges from 0 to 1. The larger the value of NMI is, the better the result is.

6.1.3. Experimental Results. We set the temporally regularized parameter $\lambda = 1.0$ for SOSComm and CEMNTR. Since the number of multicommunities K is an important parameter for SOSComm, we evaluate the performance with different K on the 4 synthetic networks firstly. With K varying from 1 to 10, the average values of NMIs of SOSComm on the 4 synthetic networks are shown in Figure 3. Obviously, on Syn1 and Syn3, SOSComm performs best when $K = 2$, and on Syn2 and Syn4, SOSComm performs best when $K = 4$. The results are consistent with the real setting for synthetic networks in Table 1; that is, the real number of multityped communities is 2 for Syn1 and Syn3, and 4 for Syn2 and Syn4. With the widening gap between K and the real number of multityped communities, SOSComm performs worse and worse in all synthetic networks.

In the following experiments, we fix K as the real number of multityped communities in each synthetic network. The comparison of NMIs for SOSComm and two baselines on the 4 synthetic networks is shown in Figure 4. In Figure 4, each subgraph shows NMIs of the three methods on each network snapshot in corresponding synthetic network. The tendency of the NMI curve turns out the ability of tracing communities evolution. From the 4 subgraphs in Figure 4, we find that SOSComm performs best on NMI and tracing communities evolution. Since no knowledge of previous community membership at the first timestamp is available, SOSComm and SOSClus share the same starting point on the 4 synthetic networks. Moreover, with the time evolving, SOSComm can trace the evolution of multityped communities closely, while the NMIs of SOSClus and CEMNTR on the 4 synthetic networks decline steadily.

As shown in Figure 4(d), with the new objects coming and old objects vanishing in the network at each timestamp, the NMIs of SOSClus and CEMNTR on Syn4 drop sharply; in detail, NMI of SOSClus drops from 1.0 to 0.2865 and NMI of CEMNTR drops from 0.8099 to 0.0976. Meanwhile, NMI of SOSComm keeps smooth relatively. This reveals that SOSComm can handle the time-evolving heterogeneous information networks with new objects coming and old objects vanishing effectively.

The convergence speed is also a significant focus for studying the performances of our framework. We run SOSComm on Syn3 and Syn4 with $\lambda = 1.0$ and analyze the changes of the objective function $\mathscr{L}^{(t)}$ between adjacent iterations, denoted as error $= |\mathscr{L}^{(t)}_{\text{iter}+1} - \mathscr{L}^{(t)}_{\text{iter}}|/\mathscr{L}^{(t)}_{\text{iter}}$, for all timestamps in the two network sequences. When the errors almost keep constant, the algorithm converges. Figure 5 shows the experimental results of error, where each subgraph displays the convergence speed of SOSComm on Syn3 and Syn4 at corresponding timestamp. In Figure 5, we can see that SOSComm converges quickly on both Syn3 and Syn4 at all timestamps. Particularly, SOSComm has converged when the iterations are less than 10 in all subgraphs, which is a good property for online deployment.

The temporally regularized parameter λ in (10) controls the impact of historical information on the current community distribution. The larger the λ is, the more significant the impact is. To study the influence of temporally regularized parameter λ tuning, we apply SOSComm on Syn4 with λ varying from 0.1 to 100. The average values of NMIs and iterations on all network snapshots over all timestamps are shown in Figure 6, where the coordinates of x-axis are based on a logarithmic transformation. As shown in Figure 6, the NMIs and iterations maintain the satisfactory results when λ is less than 10. However, when $\lambda > 10$ and keeps increasing, the performances of NMIs and iterations become worse quickly. That is, the historical information dominates and the algorithm consumes more resources to smooth the time-evolving communities, when the temporally regularized parameter λ is too large. Certainly, the temporally regularized parameter contributes to multityped communities detection by considering the temporal information when λ ranges from 0.1 to 10.

To conclude, the experiments on the 4 time-evolving synthetic networks demonstrate that SOSComm outperforms the SOSClus and CEMNTR. With a fast convergence speed, SOSComm can trace the evolution of multityped communities in the 4 synthetic networks accurately. In particular, on Syn4, with the new objects coming and old objects vanishing in the network, SOSComm can detect the multityped communities evolution well, while the performances of SOSClus and CEMNTR deteriorate rapidly as time goes on. The performances of NMIs for SOSComm on the 4 synthetic networks with different K show that SOSComm is sensitive to K. The K is closer to the real number of multityped of communities, so SOSComm performs better. Moreover,

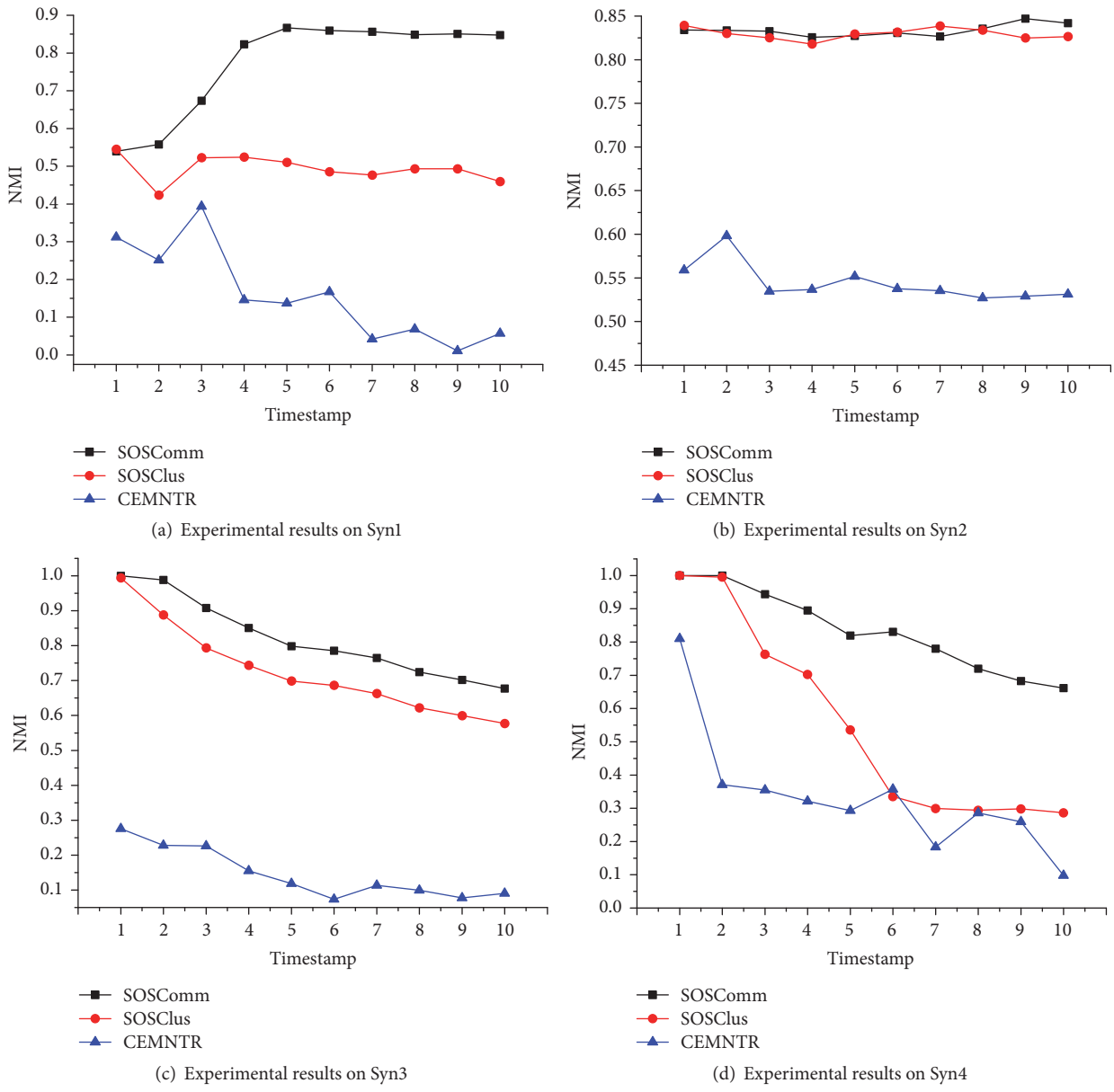

FIGURE 4: The comparison of the performances of NMIs for SOSComm, SOSClus, and CEMNTR at each timestamp on the 4 synthetic networks.

when λ ranges from 0.1 to 10, the performances of SOSComm are satisfactory.

6.2. Experiments on Real-World Dataset

6.2.1. Dataset Description. Here, we compare the performances of SOSComm with the baselines on real-world dataset. The real-world dataset is a 25-year DBLP network sequence, which is collected by Tang et al. [27] and available online: http://www.leitang.net/heterogeneous_network.html. In the 25-year DBLP dataset, the papers published from 1980 to 2004 are extracted, and all related authors, terms (words contained in the papers' titles), and venues (the conferences or journals the papers published in) are included. The low frequency used and stop words have been abandoned. In the real-world dataset, the 25-year DBLP network is segmented into 25 network snapshots according to the publication year associated with each paper. After that, we construct a 4-mode tensor for each network snapshot, where the 4 modes of the tensors represent the papers, authors, venues, and terms, respectively. Table 2 shows the number of papers, authors, venues, terms, and gene-networks in each network snapshot of the 25-year DBLP dataset. Meanwhile, each row in Table 2 indicates the size of the corresponding tensor. For example, the size of the tensor for year = 2004 is 69,021 × 105,292 × 1,238 × 9,153, with 1,182,458 nonzero elements. It is worth noting that there is no ground-truth of community memberships in the real-world dataset, because it is difficult

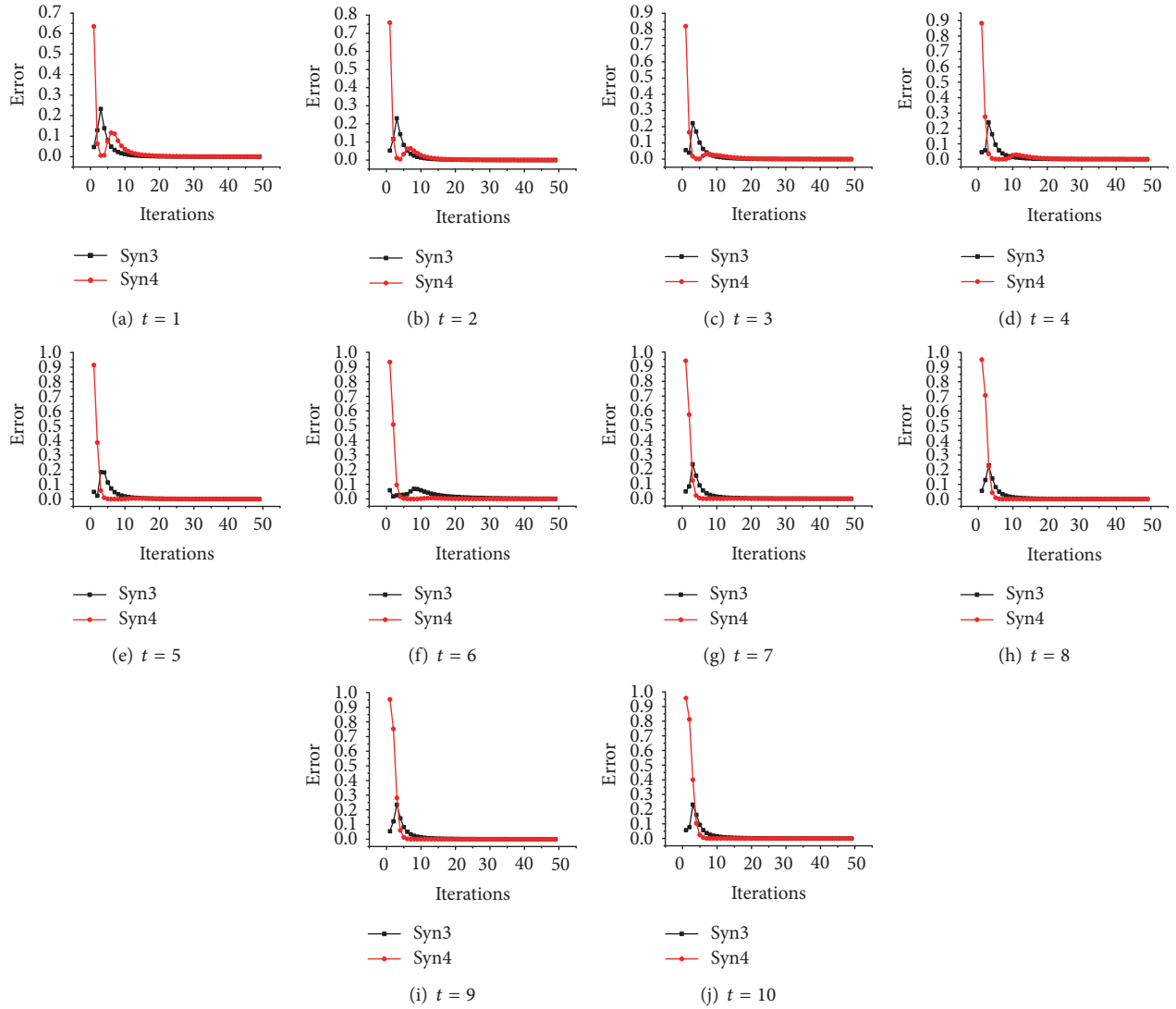

Figure 5: The changes of the objective function between adjacent iterations for SOSComm on Syn3 and Syn4 at each timestamp.

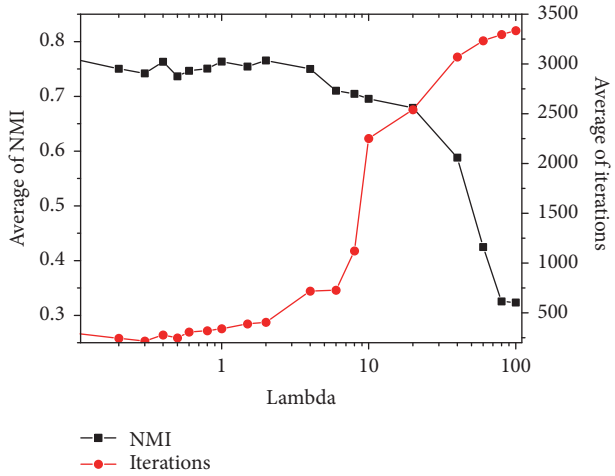

Figure 6: The average values of NMIs and iterations of SOSComm on Syn4 with different λs.

and unrealistic to label the massive objects in a real-world network automatically or even manually.

6.2.2. Evaluation Metrics. Different from the synthetic networks, NMI cannot be adopted as the metric to evaluate the performances due to the lack of ground-truth of community membership in the real-world dataset. In fact, to evaluate the detection of community evolution is challenging. Alternatively, we extend the modularity Q [43, 44], a widely used metric of measuring the quality of communities in a homogenous network, to the high-order tensor space, so that the extended modularity Q is suitable for the heterogeneous information networks. In a network, the high modularity reflects dense connections among vertices within a community and sparse connections among vertices across different communities.

Following the work of [44], modularity Q is defined as the fraction of the edges that fall within the given communities

Table 2: The details of the real-world dataset.

Year	Type				
	Paper	Author	Venue	Term	Gene-network
			Number		
1980	2783	3400	80	2994	21814
1981	3693	4630	95	3511	31156
1982	3525	4418	89	3451	30228
1983	3872	5066	100	3742	33790
1984	4299	5674	106	3917	38060
1985	5076	6630	124	4238	45081
1986	5531	7539	145	4505	51625
1987	6368	8871	170	4929	60488
1988	7522	10415	195	5169	72363
1989	8665	11856	213	5562	85161
1990	10332	14801	243	6005	105975
1991	11435	16107	268	6134	116875
1992	13654	19546	323	6609	147849
1993	15183	22130	363	6870	171968
1994	16860	25160	380	7197	201015
1995	18532	27737	418	7469	225549
1996	21611	31828	472	7828	266749
1997	25492	38684	551	8233	338227
1998	26133	40595	584	8352	351770
1999	29082	45201	634	8515	401446
2000	34500	53735	718	8721	492333
2001	40402	62770	852	8948	601616
2002	47322	72126	957	9059	720623
2003	60833	92843	1144	9198	978413
2004	69021	105292	1238	9153	1182458

minus the expected fraction of randomization of these edges with the fixed degree of each vertex. We directly give the calculation of modularity Q in [44]:

$$Q = \frac{1}{2e}\sum_{v,w}\left(a_{vw} - \frac{d_v d_w}{2e}\right)\delta(v,w), \tag{26}$$

where e is the total number of edges in the whole network, a_{vm} is an element of adjacent matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, and d_v denotes the degree of vertices v. The function $\delta(v,w)$ indicates whether the vertices v and w are in the same community or not. The value of Q falls in the range $[-0.5, 1)$, which can be negative. In practice, when the value of Q ranges from 0.3 to 0.7, the quality of community is satisfactory.

Without loss of generality, we take the heterogeneous information network at the tth timestamp as an example and ignore the superscript of timestamp in the following discussion. In our framework, each nonzero element of tensor $\mathcal{X}$ maps a gene-network in the given heterogeneous information network, while the outer product of a series of the kth column in the corresponding factor matrices indicates the distribution of the kth multityped community for gene-networks; that is, $\mathcal{C}_k = \mathbf{u}_k^{(1)} \circ \mathbf{u}_k^{(2)} \circ \cdots \circ \mathbf{u}_k^{(N)}$. In other words, a gene-network is the minimum unit in our framework. Then, a new graph Φ reflecting the connections of gene-networks is formed, in which each gene-network in the original heterogeneous information network is treated as a vertex. In other words, the vertices in Φ are the gene-networks in original heterogeneous information network. If two vertices ϕ and φ are connected or an edge between ϕ and φ exists in Φ, this means that the gene-networks denoted by ϕ and φ in the original heterogeneous information network share one or more same objects.

Accordingly, the modularity Q can be used to evaluate the quality of communities in Φ. Since the vertices in Φ are in one-to-one correspondence with gene-networks in original heterogeneous information network, the multityped communities $\mathcal{C}_k|_{k=1}^K$ consisting of gene-networks in original heterogeneous information network are also the partition of communities in Φ. Let J denote the total number of vertices in Φ; that is, $J = nnz(\mathcal{X})$. The adjacent matrix of Φ becomes $\mathbf{A} \in \{0,1\}^{J\times J}$, whose element $a_{\phi\varphi}$ indicates whether ϕ connects to φ or not. Here, the adjacent matrix $\mathbf{A}$ is a symmetric matrix with all zeros diagonal; that is, $a_{\phi\varphi} = a_{\varphi\phi}$ and $a_{\phi\phi} = 0$.

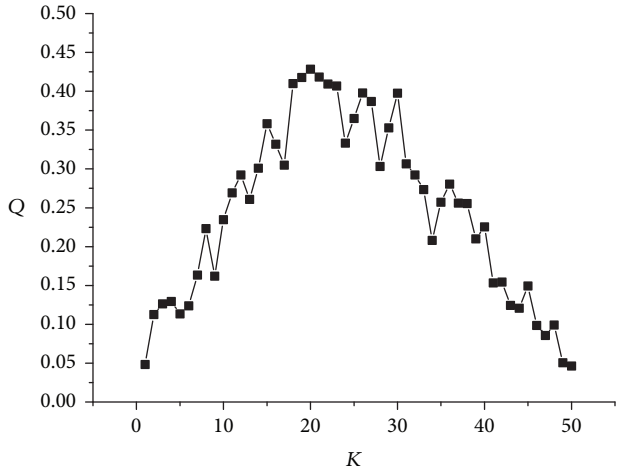

Figure 7: The performances of SOSComm on the 25-year DBLP network with different K.

Thereby, the total number of edges in Φ is $e = (1/2)\sum_{\phi\varphi} a_{\phi\varphi}$, and the degree of ϕ is $d_\phi = \sum_\varphi a_{\phi\varphi}$. According to (26), the extended modularity Q (also denoted by Q) can be calculated by

$$Q = \frac{1}{2e}\sum_{\phi,\varphi}\left(a_{\phi\varphi} - \frac{d_\phi d_\varphi}{2e}\right)\delta(\phi,\varphi). \quad (27)$$

If ϕ and φ are in the same multityped community, $\delta(\phi,\varphi) = 1$. Otherwise, $\delta(\phi,\varphi) = 0$.

6.2.3. Experimental Results. Firstly, The baselines and SOSComm are deployed in offline mode in order to learn their best performances on multityped communities discovery. That is, the baselines and SOSComm are iterated on each network snapshot until they converge. In the offline mode, we share the same comparative methods and experimental setting as that in experiments on synthetic networks; that is, the change of corresponding objective function is less than 10^{-6} and the maximum iterations MaxIter = 10000. We set the temporally regularized parameter $\lambda = 1.0$ for SOSComm and CEMNTR.

To seek out the suitable number of multityped communities, we perform the SOSComm on the 25-year DBLP network with different K. Figure 7 gives the average values of modularity Q on the 25 timestamps with K varying from 1 to 50. In Figure 7, when $14 \le K \le 31$, $Q \ge 0.3$. Though the average values of modularity Q are almost equal when $K = 18, 19, 20, 21, 22$, and 23, the maximum of Q is obtained when $K = 20$. Therefore, in the following experiments, the number of multityped communities in the 25-year DBLP network is fixed to 20.

The comparison of modularity Q for the baselines and SOSComm in offline mode is shown in Figure 8. SOSComm performs the best modularity Q on each network snapshot. With the time evolving, SOSComm traces the evolution of multityped communities more and more closely, while the modularity Q of SOSClus keeps low all the time and the modularity Q of CEMNTR declines steadily.

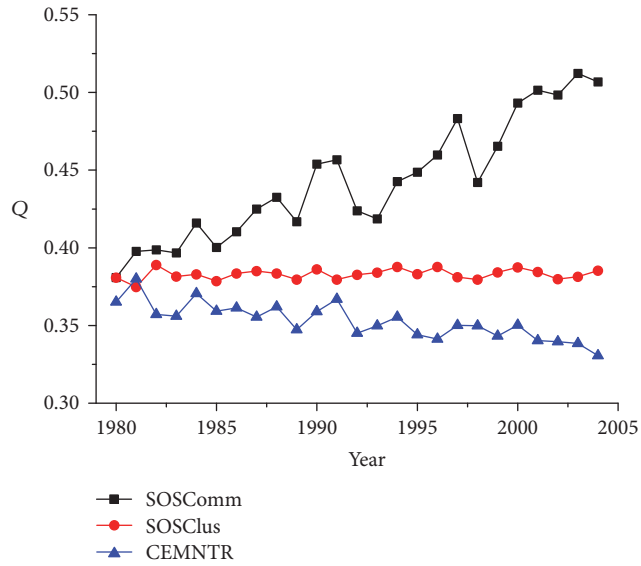

Figure 8: The comparison of modularity Q for SOSComm, SOSClus, and CEMNTR in offline mode.

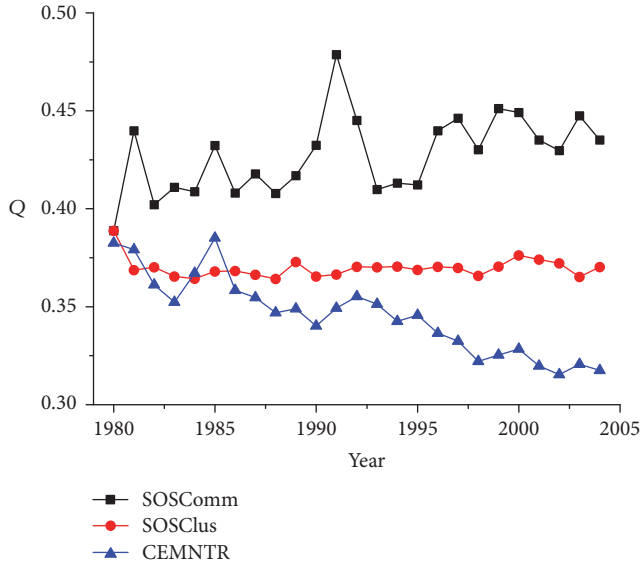

Figure 9: The comparison of modularity Q for SOSComm, SOSClus, and CEMNTR in online mode.

Secondly, we learn the performances of SOSComm in online mode. In the online mode, the maximum iteration is limited to 5. The comparison of modularity Q for the baselines and SOSComm in online mode is shown in Figure 9. Although the modularity Q of SOSComm has declined relatively to that in offline mode, its performance is still the best.

In addition, Figure 10 shows the comparison of modularity Q for SOSComm in offline mode and online mode. In Figure 10, we can find that the performance of SOSComm in online mode is not worse than that in offline mode. Before 2000, the two curves almost overlap. With the explosive growth of the tensors in the last 5 years, the modularity Q of

Table 3: The running time of the three methods on the real-world dataset in online mode.

Year	Method		
	SOSClus	CEMNTR	SOSComm
		Time (s)	
1980	10.27	**0.9472**	1.59
1981	6.14	1.4157	**1.24**
1982	6.10	1.41	**1.25**
1983	6.19	1.53	**1.33**
1984	6.07	1.63	**1.36**
1985	6.20	1.76	**1.35**
1986	6.27	1.98	**1.35**
1987	6.36	2.16	**1.66**
1988	6.38	2.66	**1.37**
1989	6.55	**2.92**	3.10
1990	6.75	**3.90**	6.73
1991	6.85	**3.49**	5.46
1992	8.31	4.18	**4.13**
1993	8.83	**4.70**	7.37
1994	9.48	6.08	**4.34**
1995	11.34	**5.82**	12.06
1996	12.66	12.12	**8.26**
1997	14.02	14.00	**4.92**
1998	17.94	14.52	**8.31**
1999	20.36	9.02	**5.19**
2000	33.79	**11.31**	19.50
2001	24.96	12.87	**12.26**
2002	27.54	49.05	**15.52**
2003	28.93	19.18	**17.90**
2004	36.71	21.94	**10.67**

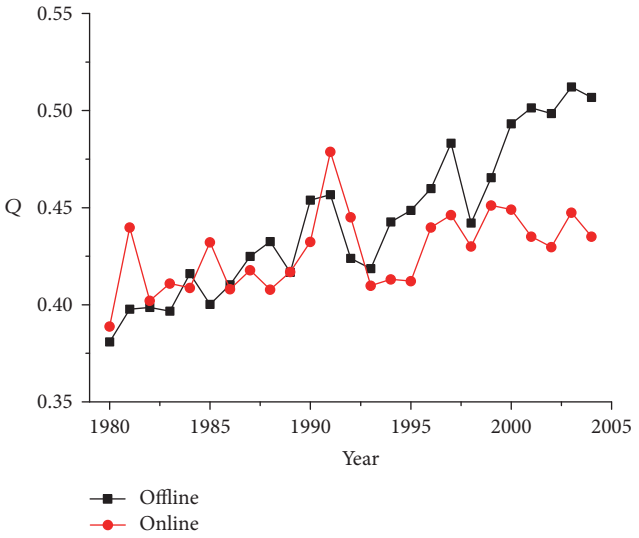

Figure 10: The comparison of modularity Q for SOSComm in offline mode and online mode.

SOSComm in online mode is slightly less than that in offline mode. Table 3 summarizes the running time of the baselines and SOSComm in online mode. CEMNTR and SOSComm, as shown in Table 3, yield the obvious advantages. Most of the time, SOSComm is the fastest.

To summarize, the experiments on the 25-year DBLP dataset show that SOSComm outperforms the SOSClus and CEMNTR. With a larger modularity Q, SOSComm can detect the multityped communities and trace their evolution in the 25-year DBLP network. In particular, the experimental results of online mode demonstrate that SOSComm has the best performances on modularity Q and running time. That is, SOSComm has a good property of online deployment.

7. Conclusion

In this paper, a novel online framework for multityped community discovery in time-evolving heterogeneous information network without the restriction of network schema is proposed. Each snapshot of the network sequence is expressed as a tensor, and the multityped community is modeled as a rank-one tensor. Then, the problem of multityped community discovery is formalized as a tensor decomposition, which integrates the tensor CP factorization with a temporal evolution regularization term. In addition, a second-order stochastic gradient descent algorithm, named SOSComm, is designed to address the tensor decomposition. In this framework, the community membership matrices

of all types of objects, the multityped communities, and their evolutions over time can be obtained simultaneously. Whether in offline or online mode, the proposed algorithm outperformed the other state-of-the-art methods.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Science Foundation of China (no. 61401482 and no. 61401483).

References

[1] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Mining hidden community in heterogeneous social network," in *Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD '05)*, pp. 58–65, USA.

[2] H. Ma, H. Yang, M. R. Lyu, and I. King, "Mining social networks using heat diffusion processes for marketing candidates selection," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pp. 233–242, ACM, Napa Valley, California, CA, USA, October 2008.

[3] A. H. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Ontology matching: A machine learning approach," in *International Handbooks on Information Systems*, pp. 397–416, 2003.

[4] F. Tao, G. Brova, J. Han et al., "NewsNetExplorer: Automatic construction and exploration of news information networks," in *Proceedings of the International Conference on Management of Data (SIGMOD '14)*, ACM, USA, 2014.

[5] M. Gomez Rodriguez and L. Song, "Diffusion in social and information networks: Research problems, probabilistic models & machine learning methods," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '15)*, pp. 2315-2316, Australia, 2015.

[6] X. Yu, X. Ren, Y. Sun et al., "Recommendation in heterogeneous information networks with implicit user feedback," in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*, pp. 347–350, China, October 2013.

[7] X. Yu, X. Ren, Y. Sun et al., "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining*, pp. 283–292, New York, NY, USA, Feburary 2014.

[8] X. Wang, L. Tang, H. Liu, and L. Wang, "Learning with multi-resolution overlapping communities," *Knowledge and Information Systems*, vol. 36, no. 2, pp. 517–535, 2013.

[9] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

[10] R. Jin, C. Kou, and R. Liu, "Improving community detection in time-evolving networks through clustering fusion," *Cybernetics and Information Technologies*, vol. 15, no. 2, pp. 63–74, 2015.

[11] C. C. Aggarwal and P. S. Yu, "Online analysis of community evolution in data streams," *Sdm Lars Backstrom Dan Huttenlocher Jon Kleinberg and Xiangyang*, 2005.

[12] M. Revelle, C. Domeniconi, M. Sweeney, and A. Johri, "Finding community topics and membership in graphs," *ECML PKDD*, pp. 625–640, 2015.

[13] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 797–806, Paris, France, July 2009.

[14] Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao, "Community evolution detection in dynamic heterogeneous information networks," in *Proceedings of the 8th Workshop on Mining and Learning with Graphs (MLG '10)*, pp. 137–146, July 2010.

[15] Y. Sun, J. Tang, J. Han, C. Chen, and M. Gupta, "Co-evolution of multi-typed objects in dynamic star networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2942–2955, 2014.

[16] P. W. Holland and K. B. Laskey, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[17] K. Nowicki, "Estimation and prediction for stochastic block-structures," *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 1077–1087, 2001.

[18] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. 5, pp. 1981–2014, 2008.

[19] J. Sun, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Community evolution and change point detection in time-evolving graphs," in *Link Mining: Models, Algorithms, and Applications*, pp. 73–104, Springer, New York, NY, USA, 2010.

[20] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng, "Blog community discovery and evolution based on mutual awareness expansion," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07)*, pp. 48–56, USA, November 2007.

[21] G. Palla, A. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[22] A. Cuzzocrea, F. Folino, and C. Pizzuti, "Dynamicnet: An effective and efficient algorithm for supporting community evolution detection in time-evolving information networks," in *Proceedings of the 17th International Database Engineering and Applications Symposium (IDEAS '13)*, pp. 148–153, ACM, New York, NY, USA, 2013.

[23] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, pp. 1–31, 2009.

[24] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A framework for community identification in dynamic social networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 717–726, USA, August 2007.

[25] F. Folino and C. Pizzuti, "An evolutionary multiobjective approach for community discovery in dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1838–1852, 2014.

[26] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 677–685, USA, August 2008.

[27] L. Tang, H. Liu, and J. Zhang, "Identifying evolving groups in dynamic multimode networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 72–85, 2012.

[28] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "MetaFac: Community discovery via relational hypergraph factorization," in *Proceedings of the 15th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining (KDD '09), pp. 527–535, France, July 2009.

[29] Y.-R. Lin, J. Sun, H. Sundaram, A. Kelliher, P. Castro, and R. Konuru, "Community discovery via MetaGraph Factorization," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 3, article 17, 2011.

[30] A. Animashree, G. Rong, H. Daniel, and M. K. Sham, "A tensor spectral approach to learning mixed membership community models," in *Proceedings of the in JMLR: Workshop and Conference Proceedings*, 2013.

[31] J. Wu, Y. Wu, S. Deng, and H. Huang, "Multi-way clustering for heterogeneous information networks with general network schema," in *Proceedings of the 16th IEEE International Conference on Computer and Information Technology (CIT '16)*, pp. 339–346, December 2016.

[32] J. Wu, Q. Meng, S. Deng, H. Huang, Y. Wu, and A. Badii, "Generic, network schema agnostic sparse tensor factorization for single-pass clustering of heterogeneous information networks," *PLoS ONE*, vol. 12, no. 2, Article ID e0172323, 2017.

[33] J. Wu, Z. Wang, Y. Wu, L. Liu, S. Deng, and H. Huang, "A Tensor CP decomposition method for clustering heterogeneous information networks via stochastic gradient descent algorithms," *Scientific Programming*, vol. 2017, Article ID 2803091, 13 pages, 2017.

[34] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: theory and applications," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 2, no. 3, article 11, 2008.

[35] M. Zhang and C. Ding, "Robust tucker tensor decomposition for effective image representation," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2448–2455, Australia, December 2013.

[36] X. Cao, X. Wei, Y. Han, and D. Lin, "Robust face clustering via tensor decomposition," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2546–2557, 2015.

[37] E. Davidson and M. Levine, "Gene regulatory networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 102, no. 14, p. 4935, 2005.

[38] P. Paatero, "Construction and analysis of degenerate PARAFAC models," *Journal of Chemometrics*, vol. 14, no. 3, pp. 285–299, 2000.

[39] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *Journal of Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[40] T. G. Kolda, "Multilinear operators for higher-order decompositions.," Tech. Rep. SAND2006-2081, 2006.

[41] B. L. Bottou and N. Murata, "Stochastic approximations and efficient learning," *The Handbook of Brain Theory and Neural Networks, Second edition*, 2002.

[42] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003.

[43] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, 2004.

[44] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.