Hindawi
Scientific Programming
Volume 2017, Article ID 2803091, 13 pages
https://doi.org/10.1155/2017/2803091

Research Article

A Tensor CP Decomposition Method for Clustering Heterogeneous Information Networks via Stochastic Gradient Descent Algorithms

Jibing Wu, Zhifei Wang, Yahui Wu, Lihua Liu, Su Deng, and Hongbin Huang

Science and Technology on Information System Engineering Laboratory, National University of Defense Technology, Changsha, China

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

Received 1 December 2016; Revised 14 February 2017; Accepted 16 February 2017; Published 30 April 2017

Academic Editor: Fabrizio Riguzzi

Clustering analysis is a basic and essential method for mining heterogeneous information networks, which consist of multiple types of objects and rich semantic relations among different object types. Heterogeneous information networks are ubiquitous in the real-world applications, such as bibliographic networks and social media networks. Unfortunately, most existing approaches, such as spectral clustering, are designed to analyze homogeneous information networks, which are composed of only one type of objects and links. Some recent studies focused on heterogeneous information networks and yielded some research fruits, such as RankClus and NetClus. However, they often assumed that the heterogeneous information networks usually follow some simple schemas, such as bityped network schema or star network schema. To overcome the above limitations, we model the heterogeneous information network as a tensor without the restriction of network schema. Then, a tensor CP decomposition method is adapted to formulate the clustering problem in heterogeneous information networks. Further, we develop two stochastic gradient descent algorithms, namely, SGDClus and SOSClus, which lead to effective clustering multityped objects simultaneously. The experimental results on both synthetic datasets and real-world dataset have demonstrated that our proposed clustering framework can model heterogeneous information networks efficiently and outperform state-of-the-art clustering methods.

1. Introduction

Heterogeneous information networks are ubiquitous in the real-world applications. Generally, heterogeneous information networks consist of multiple types of objects and rich semantic relations among different object types. The bibliographic network extracted from the DBLP database (http://www.informatik.uni-trier.de/~ley/db/) is a typical heterogeneous information network, as shown in Figure 1. The DBLP database is an open resource that contains most bibliographic information on computer science. The bibliographic network contains four types of objects: author (A), paper (P), venue (i.e., conference or journal) (V), and term (T). The edges are labeled by "write" or "written by" between author and paper or labeled by "publish" or "published by" between paper and venue or labeled by "contain" or "contained in" between paper and term.

Clustering analysis is a basic and essential method for mining such networks, which can help us better understand the semantic information and interpretable structure in the network. Unfortunately, most existing approaches, such as spectral clustering, are designed to analyze *homogeneous information networks* [1] that consist of only a single type of objects and links, while the real-world situations are often *heterogeneous information networks* [2] in nature with more than one type of objects and links. The mission of clustering such a heterogeneous information network is more difficult than that in a homogeneous information network, as we cannot directly measure the similarity among the different types of objects and relations.

Though some recent studies have focused on clustering heterogeneous information networks, such as RankClus [1] and NetClus [2], they can only be applied to some specific simple network schemas. RankClus can only be used to model bityped networks, where only two different types of objects exist in the network. NetClus was developed for the star network schema, where the links only appear between

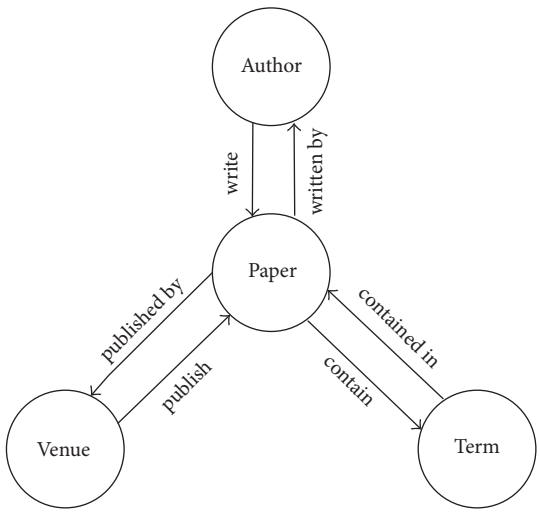

Figure 1: An example of heterogeneous information network: a bibliographic network extracted from the DBLP database.

target objects and attribute objects. The network schema is a metatemplate of a heterogeneous information network, which shows how many types of objects and links are there in the network [3]. Figure 1 shows a typical star network schema, where the paper (P) is the target object and others are attribute objects.

A tensor is a generalization of the matrix in the high-dimensional space. It is a natural expression of complicated and interpretable structures in high-mode data. In this paper, we model a heterogeneous information network as a tensor without the restriction of network schema. Each type of objects maps onto one mode of the tensor, and the semantic relations among different object types map onto the elements in the tensor. Then, a tensor CP decomposition method is adapted to formulate the clustering problem in heterogeneous information networks. And two stochastic gradient descent algorithms are developed, which lead to effective clustering multityped objects simultaneously. The experimental results on both synthetic datasets and real-world dataset show that the proposed clustering framework can model the heterogeneous information networks efficiently and outperform the state-of-the-art clustering methods.

The rest of this paper is organized as follows. In Section 2, we discuss the related work on clustering for heterogeneous information networks and the tensor factorization. Section 3 gives some notations and definitions used in this paper. In Section 4, we formulate the clustering problem and describe two stochastic gradient descent algorithms. The experimental results on both synthetic datasets and real-world dataset are presented in Section 5. Finally, the conclusions are drawn in Section 6.

2. Related Work

Our work mainly focuses on the clustering heterogeneous information networks and the tensor factorization.

2.1. Clustering Heterogeneous Information Networks. Clustering is an unsupervised learning method to recognize the distribution and hidden structures in the data, which is a basic and significant mission for pattern recognition and machine learning. Since MacQueen first proposed K-means [4] in 1967, many subtle algorithms have been developed for clustering in the past decades. However, most existing algorithms, such as hierarchical clustering algorithm [5], density-based clustering [6], mesh-based clustering [7], fuzzy clustering algorithm [8], and spectral clustering [9], are designed to analyze point sets or homogeneous information networks, which are composed of only one type of objects and links.

In real-world applications, the datasets are often organized as heterogeneous information networks, where objects and the relations between them are of more than one type. In recent years, researchers have made a significant progress on clustering for heterogeneous information networks [10, 11], which largely focused on four main directions: the first is to use a ranking based clustering algorithm [1]; it developed the RankClus algorithm that integrated clustering with ranking for clustering bityped networks. Its extension, NetClus [2], was developed for the star network schema. They have proven that ranking and clustering can mutually enhance each other. GPNRankClus [12] assumed that edges in heterogeneous information networks follow a Poisson distribution. This method can simultaneously achieve both clustering and ranking in a heterogeneous information network. In addition, FctClus [13] achieved a higher computational speed and had a greater clustering accuracy when applied to heterogeneous information networks. But, same as NetClus, FctClus algorithm can only handle the star network schema. For a general network schema, HeProjI [14] projected the network into a number of bityped or star schema subnetworks and performed the ranking based clustering in each subnetwork.

The second direction involves metapath based clustering algorithms [15, 16]. A metapath is a connected path defined on the network schema of a heterogeneous information network, which represents a composite semantic relation between two objects. PathSim (metapath based top-k similarity search) [3] measured the similarity between the same types of objects based on metapath in heterogeneous information networks. However, it has a limitation: the metapath must be symmetric; that is, PathSim could not work on different types of objects. The PathSelClus algorithm in [15–17] integrated metapath selection with user guidance to cluster objects in networks, where user provided seeds for each cluster acted as guidance.

The third direction is structural-based clustering. Sun et al. proposed a probabilistic clustering method [18] to deal with heterogeneous information networks with incomplete attributes, which integrated the incomplete attribute information and the network structure information. NetSim [19] is a structural-based similarity measurement between objects for x-star network. Xu proposed a Bayesian probabilistic model based on network structural information for clustering heterogeneous information networks.

The final direction is a clustering algorithm based on social network features. Zhou and Liu designed the SI-Cluster algorithm [20], which adopted the heat diffusion procedure

to model the social influence and then measure the similarity between objects.

2.2. Tensor Factorization. A tensor is a multidimensional array, in which the elements are addressed by more than two indices. Tensor factorization has been studied since the early 20th century [21–25]. Two of the most popular tensor factorization methods are Tucker decomposition [21, 24] and canonical decomposition using parallel factors (CANDECOMP/PARAFAC) [23, 24]. The CANDECOMP/PARAFAC is also named CP decomposition. It is worth noting that CP decomposition is a special case of Tucker decomposition.

The clustering issues based on tensor factorization are often modeled as the optimization problems [26]. But it has been proven in [27] that tensor clustering formulations are NP-hard. In the past years, many approximation algorithms for tensor clustering are proposed [28–30]. These theories provide a new perspective for us as to clustering heterogeneous information networks. Tensor factorization based clustering has also been used in some specific applications. Examples include link prediction in higher-order network structures [31, 32], collaborative filtering in recommendation systems [33], community detection in multigraphs [34], graph clustering [35], and modeling multisource datasets [36].

The Alternating Least Squares (ALS) algorithm [22, 37, 38] is one of the most famous and commonly used algorithms to solve the tensor factorization, which updates one component iteratively at each round, while holding the others constant. However, ALS suffers from some limitations; for example, ALS may converge to a local minimum and the memory consumption may explode when the scale of tensor is large. Nonlinear optimization approach is another option to obtain the tensor factorization, such as nonlinear conjugate gradient method [39], Newton based optimization [40], randomized block sampling method [41], and stochastic gradient descent [42]. In this paper, we adopt a stochastic gradient descent algorithm with Tikhonov regularization item loss function to process the tensor CP decomposition based clustering.

3. Preliminaries

First, we introduce some related concepts and notations of tensors used in this paper. More details about tensor algebra can be found in [24, 43]. The order of a tensor is the number of dimensions, also known as ways or modes. We will follow the convention used in [23] to denote scalars by lowercase letters, for example, a, b, c, vectors (one mode) by boldface lowercase letters, for example, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, matrices (two modes) by boldface capital letters, for example, $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and tensors (three modes or more) by boldface calligraphic letters, for example, $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. Elements of a matrix or a tensor are denoted by lowercase letters with subscripts; that is, the $(i_1, i_2, \ldots, i_N)$th element of an Nth-order tensor $\mathcal{X}$ is denoted by $x_{i_1, i_2, \ldots, i_N}$. The notations about tensor algebra used in this paper are summarized in Notations.

If an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be written as the outer product of N vectors, that is, $\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$ and $\mathbf{a}^{(n)} \in \mathbb{R}^{I_n}; n = 1, 2, \ldots, N$, tensor $\mathcal{X}$ is named rank-one tensor. The CP decomposition represents a tensor as a sum of R rank-one tensors; that is, the CP decomposition of $\mathcal{X}$ is $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}$, where R is a positive integer and $\mathbf{a}_r^{(n)} \in \mathbb{R}^{I_n}; r = 1, 2, \ldots, R; n = 1, 2, \ldots, N$. The rank of a tensor is defined as the smallest number of rank-one tensors for which the equality holds in the CP decomposition and denoted as $\text{rank}(\mathcal{X}) = \min R$. In fact, the problem of tensor rank determination is NP-hard [27].

Let factor matrices $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$, for $n = 1, 2, \ldots, N$. We denote $[\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \equiv \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}$. By minimizing the Frobenius norm of the difference between $\mathcal{X}$ and its CP approximation, the CP decomposition can be formulated as an optimization problem:

$$\min \quad \frac{1}{2} \left\| \mathcal{X} - [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \right\|_F^2. \tag{1}$$

Then, we give the definitions for the information network and the network schema, which are based on the work by Sun et al. [3].

Definition 1 (information network [3]). An *information network* is a graph $G = (V, E)$ defined on a set of objects V and a set of links E, where V belongs to T objects types $\mathbb{V} = \{\mathcal{V}_t\}_{t=1}^{T}$ and E belongs to L links types $\mathbb{E} = \{\mathcal{R}_l\}_{l=1}^{L}$.

Specifically, when $T > 1$ or $L > 1$, the information network is called *heterogeneous information network*; otherwise, it is called *homogeneous information network*.

We denote the object set of type $\mathcal{V}_t$ as $\{v_n^t\}_{n=1}^{N_t}$, where N_t is the number of objects in type $\mathcal{V}_t$; that is, $N_t = |\mathcal{V}_t|$ and $t = 1, 2, \ldots, T$. The total number of objects in the network is given by $N = \sum_{t=1}^{T} N_t$.

Definition 2 (network schema [3]). The *network schema* is a metatemplate for a heterogeneous information network $G = (V, E)$, which is a graph defined over object types $\mathbb{V}$ and links types $\mathbb{E}$, denoted by $S_G = \{\mathbb{V}, \mathbb{E}\}$.

A network schema $S_G = \{\mathbb{V}, \mathbb{E}\}$ shows how many types of objects are there in the network $G = (V, E)$ and which type the links between different object types belong to. Figure 1 shows the network schema of DBLP, which follows a star network schema.

4. Tensor CP Decomposition Based Clustering Framework

4.1. Tensor Construction. According to Definition 2, we know that the network schema $S_G = \{\mathbb{V}, \mathbb{E}\}$ is a metatemplate for the given heterogeneous information network $G = (V, E)$. In other words, $G = (V, E)$ is an instance of $S_G = \{\mathbb{V}, \mathbb{E}\}$. So we can find at least one subnetwork of $G = (V, E)$, which follows the schema $S_G = \{\mathbb{V}, \mathbb{E}\}$.

Definition 3 (gene-network). A gene-network, denoted by $G' = (V', E')$, is the minimum subnetwork of $G = (V, E)$, which follows the schema $S_G = \{\mathbb{V}, \mathbb{E}\}$.

It is easy to see that a gene-network is one of the smallest instances of $S_G = \{\mathbb{V}, \mathbb{E}\}$ in the set of subnetworks of $G = (V, E)$. For example, a gene-network in

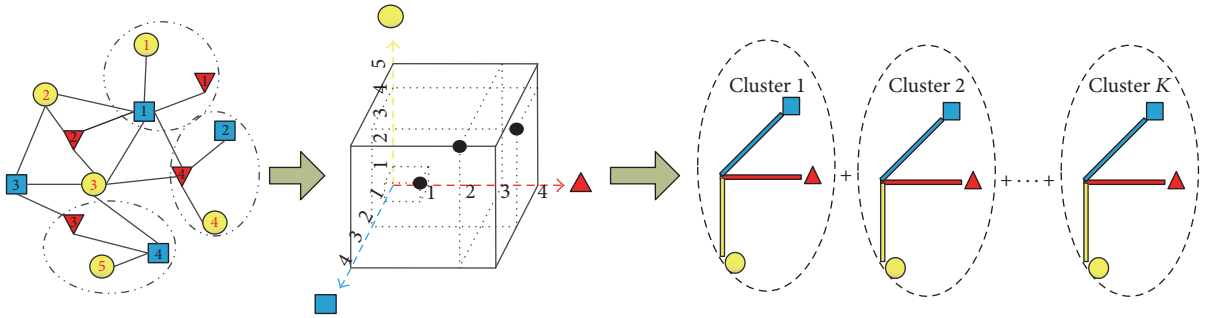

Figure 2: An illustration of tensor CP decomposition method for clustering heterogeneous information network.

DBLP network (in Figure 1, which contains four types of objects, $\{A, P, V, T\}$), denoted by $G' = (\{v_i^A, v_j^P, v_m^V, v_n^T\}, \{\langle v_i^A, v_j^P\rangle, \langle v_j^P, v_m^V\rangle \langle v_j^P, v_n^T\rangle\})$, represents a semantic relation of *"an Author v_i^A writes a Paper v_j^P published in the Venue v_m^V and containing the Term v_n^T."* For simplicity, we can use the subscript of each object in G' to mark the corresponding gene-network. In the example, the gene-network G' can be marked by $G'_{i,j,m,n}$.

Let $\mathcal{X}$ be a Tth-order tensor of size $N_1 \times N_2 \times \cdots \times N_T$; each mode of $\mathcal{X}$ represents one type of objects in the network G. An arbitrary element, $x_{n_1 n_2 \cdots n_T} \in \{0, 1\}$, for $n_t = 1, 2, \ldots, N_t$, is an indicator of whether the corresponding gene-network $G'_{n_1, n_2, \ldots, n_T}$ exists; that is,

$$x_{n_1 n_2 \cdots n_T} = \begin{cases} 1 & \text{if } \exists G'_{n_1, n_2, \ldots, n_T}; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Then, the heterogeneous information network $G = (V, E)$ can be represented by the form of tensor as $\mathcal{X}$.

4.2. Problem Formulation. Using the tensor representation $\mathcal{X}$ of $G = (V, E)$, we can partition the multityped objects into different clusters by the CP decomposition. We assume that there are K clusters in $G = (V, E)$ and denote $\mathbf{U}^{(t)} \in \mathbb{R}^{N_t \times K}$; $t = 1, 2, \ldots, T$ as the cluster indication matrix of the tth type of objects. Then, the CP decomposition of $\mathcal{X}$ is

$$\min \quad \left\| \mathcal{X} - \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)} \right]\!\right] \right\|_F^2. \tag{3}$$

Each row $\mathbf{u}_k^{(t)} = [u_{k,1}^{(t)}, u_{k,2}^{(t)}, \ldots, u_{k,N_t}^{(t)}]^\top$ of the factor matrix $\mathbf{U}^{(t)}$ is the probability vector for each object from tth type belonging to the kth cluster. In other words, the kth cluster is composed of the kth rank-one tensor in the CP decomposition; that is, $\mathbf{u}_k^{(1)} \circ \mathbf{u}_k^{(2)} \circ \cdots \circ \mathbf{u}_k^{(T)}$.

Figure 2 gives an example of tensor CP decomposition method for clustering heterogeneous information network. The left one is the original network with three types of objects, the middle cube is a 3-mode tensor, and the right one is the CP decomposition of the 3-mode tensor and also is the partition of the original network. In addition, the three types of objects are the yellow round, the blue square, and the red triangle, respectively. The number within each object is the identifier of the object. Each element (black dot in the middle cube) in the tensor represents a gene-network in the network (black dashed circle in the left). Each component (black dashed circle in the right) in the CP decomposition shows one cluster of the original network.

The problem in (3) is an NP-hard nonconvex optimization problem, which has a continuous manifold of equivalent solutions [39]. In other words, the global minimum is drowned in many local minima, which makes it difficult to be found. In real-world scenarios, the objects in the heterogeneous information networks may belong to more than one cluster; that is, the clusters are overlapping. However, the number of clusters that the vast majority of objects may belong to is much smaller than the total number of clusters. That is, most of the elements in $\mathbf{U}^{(t)}$ should be zero; that is, $\mathbf{U}^{(t)}$ should be sparse. To overcome the two challenges, we can introduce a Tikhonov regularization term, proposed by Paatero in [44, 45], in the objective function and replace the objective function by the following loss function:

$$\begin{aligned} &\mathscr{L}\left(\mathcal{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)}\right) \\ &\quad = \frac{1}{2}\left\| \mathcal{X} - \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)} \right]\!\right] \right\|_F^2 + \frac{\lambda}{2}\sum_{t=1}^{T} \left\| \mathbf{U}^{(t)} \right\|_F^2, \end{aligned} \tag{4}$$

where $\lambda > 0$ is a regularization parameter. Let

$$\begin{aligned} &f\left(\mathcal{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)}\right) \\ &\quad = \frac{1}{2}\left\| \mathcal{X} - \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)} \right]\!\right] \right\|_F^2 \end{aligned} \tag{5}$$

be the first squared loss function component in $\mathscr{L}$ and let

$$g\left(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)}\right) = \frac{\lambda}{2}\sum_{t=1}^{T} \left\| \mathbf{U}^{(t)} \right\|_F^2 \tag{6}$$

be the Tikhonov regularization term in $\mathscr{L}$, respectively. Then

$$\mathscr{L} = f + g. \tag{7}$$

The Tikhonov regularization term g in the loss function $\mathscr{L}$ has an encouraging property, which makes the Frobenius

norms of all factor matrices in the optimization be equal; that is,

$$\left\|\mathbf{U}^{(1)}\right\|_F = \left\|\mathbf{U}^{(2)}\right\|_F = \cdots = \left\|\mathbf{U}^{(T)}\right\|_F. \tag{8}$$

Therefore, the local minima of loss function $\mathscr{L}$ become isolated, and any replacement and scaling of the satisfactory solutions will escape from the optimization. The details of proof can be found in [39]. Meanwhile, the Tikhonov regularization term can ensure the sparsity of the factor matrices by penalizing the number of nonzero elements.

Therefore, the tensor CP decomposition method for clustering heterogeneous information networks can be formalized as

$$\begin{aligned} \min_{\mathbf{U}^{(1)},\mathbf{U}^{(2)},\ldots,\mathbf{U}^{(T)}} \quad & \mathscr{L}\left(\mathscr{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)}\right) \\ \text{s.t.} \quad & \sum_{k=1}^{K} u_{n,k}^{(t)} = 1, \quad \forall t, \forall n, \\ & u_{n,k}^{(t)} \in [0,1], \quad \forall t, \forall n, \forall k, \\ & \sum_{n=1}^{N_t} u_{n,k}^{(t)} > 0, \quad \forall t, \forall k, \end{aligned} \tag{9}$$

where $n = 1, 2, \ldots, N_t$; $t = 1, 2, \ldots, T$; $k = 1, 2, \ldots, K$, and $K < \min\{N_1, N_2, \ldots, N_T\}$ is the total number of clusters. In (9), we divide $\mathscr{X}$ into K clusters and obtain the structure of each cluster, which includes the distribution of each object. The first constraint in (9) guarantees that the sum of probabilities for each object belonging to all clusters is 1. The second constraint in (9) represents that each probability should be in the range of $[0, 1]$. The last constraint in (9) makes sure that there is no empty cluster for any mode.

4.3. The Stochastic Gradient Descent Algorithms. Stochastic gradient descent is a mature and widely used tool for optimizing various models in machine learning, such as artificial neural networks, support vector machines, and logistic regression. In this section, the regularized clustering problem in (9) will be addressed by the stochastic gradient descent algorithms. The details of tensor algebra and properties used in this section can be found in [43].

First, we review the stochastic gradient descent algorithm. To solve an optimization problem, $\min_x Q(x)$, where $Q(x)$ is a differentiable object function to be minimized and x is a variable, the stochastic gradient descent method to update x can be described as

$$x \longleftarrow x - \eta \nabla Q(x), \tag{10}$$

where η is a positive number, named learning rate or step size. The convergence speed of stochastic gradient descent algorithm depends on the choice of learning rate η and initial solution.

Though the stochastic gradient descent algorithm may converge to a local minimum at a linear speed, the efficiency of the algorithm near the optimal point is not all roses [46]. To speed up the final optimization phase, an extension method named second-order stochastic algorithm is designed in [46], which replaces the learning rate η by the inverse of second-order derivative of the object function; that is,

$$x \longleftarrow x - \eta \left(\nabla^2 Q(x)\right)^{-1} \nabla Q(x). \tag{11}$$

Now, we apply the stochastic gradient descent and the second-order stochastic algorithm to the clustering problem in (9) and propose two algorithms, named SGDClus (**S**tochastic **G**radient **D**escent for **Clus**tering) and SOSClus (**S**econd-**O**rder **S**tochastic for **Clus**tering), respectively.

4.3.1. SGDClus. In SGDClus, we apply the stochastic gradient descent algorithm to the clustering problem in (9). According to (10), each factor matrix $\mathbf{U}^{(t)}$, for $t = 1, 2, \ldots, T$, is updated by the rule

$$\mathbf{U}^{(t)} \longleftarrow \mathbf{U}^{(t)} - \eta \frac{\partial \mathscr{L}}{\partial \mathbf{U}^{(t)}} = \mathbf{U}^{(t)} - \eta \left(\frac{\partial f}{\partial \mathbf{U}^{(t)}} + \frac{\partial g}{\partial \mathbf{U}^{(t)}}\right). \tag{12}$$

Actually, $\partial g / \partial \mathbf{U}^{(t)}$ is easy to be obtained according to (6); that is,

$$\frac{\partial g}{\partial \mathbf{U}^{(t)}} = \lambda \mathbf{U}^{(t)}. \tag{13}$$

To compute $\partial f / \partial \mathbf{U}^{(t)}$, $f(\mathscr{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)})$ can be rewritten by matricization of $\mathscr{X}$ along the tth mode as

$$f_{(t)} = \frac{1}{2}\left\|\mathscr{X}_{(t)} - \mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\right\|_F^2, \tag{14}$$

where $\odot^{(/t)}\mathbf{U} = \mathbf{U}^{(T)} \odot \cdots \odot \mathbf{U}^{(t+1)} \odot \mathbf{U}^{(t-1)} \odot \cdots \odot \mathbf{U}^{(1)}$. Then, we have

$$\begin{aligned} &\frac{\partial f_{(t)}}{\partial \mathbf{U}^{(t)}} \\ &= \frac{\partial Tr\left(\left(\mathscr{X}_{(t)} - \mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\right)\left(\mathscr{X}_{(t)} - \mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\right)^{\top}\right)}{2\partial \mathbf{U}^{(t)}} \\ &= \frac{\partial Tr\left(\mathscr{X}_{(t)}\mathscr{X}_{(t)}^{\top}\right)}{2\partial \mathbf{U}^{(t)}} - \frac{\partial Tr\left(2\mathscr{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right)\left(\mathbf{U}^{(t)}\right)^{\top}\right)}{2\partial \mathbf{U}^{(t)}} \\ &\quad + \frac{\partial Tr\left(\left(\mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\right)\left(\mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\right)^{\top}\right)}{2\partial \mathbf{U}^{(t)}} \\ &= -\mathscr{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right) + \mathbf{U}^{(t)}\left(\odot^{(/t)}\mathbf{U}\right)^{\top}\left(\odot^{(/t)}\mathbf{U}\right). \end{aligned} \tag{15}$$

Another way to compute $\partial f_{(t)} / \partial \mathbf{U}^{(t)}$ is given in [39], which has the same result as (15). Following the works in [39], we denote

$$\begin{aligned} \Gamma^{(t)} &= \left(\odot^{(/t)}\mathbf{U}\right)^{\top}\left(\odot^{(/t)}\mathbf{U}\right) \\ &= \left(\left(\mathbf{U}^{(1)}\right)^{\top}\mathbf{U}^{(1)}\right) * \cdots * \left(\left(\mathbf{U}^{(t-1)}\right)^{\top}\mathbf{U}^{(t-1)}\right) \\ &\quad * \left(\left(\mathbf{U}^{(t+1)}\right)^{\top}\mathbf{U}^{(t+1)}\right) * \cdots * \left(\left(\mathbf{U}^{(T)}\right)^{\top}\mathbf{U}^{(T)}\right). \end{aligned} \tag{16}$$

Input: $\mathcal{X}$, K, λ.
Output: $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$.
(1) Initialize $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$;
(2) Set iter $\leftarrow$ 1;
(3) **repeat**
(4) **for** $t \leftarrow 1$ **to** T **do**
(5) Set η;
(6) Update $\mathbf{U}^{(t)}$ according to (18);
(7) Normalize $\mathbf{U}^{(t)}$ according to (19);
(8) **end for**
(9) Set iter $\leftarrow$ iter + 1;
(10) **until** $\mathscr{L}(\mathcal{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)})$ unchanged or iter reaches the maximum iterations

Algorithm 1: The SGDClus algorithm.

Therefore, the partial derivative of $\mathscr{L}$ is given by

$$\frac{\partial \mathscr{L}}{\partial \mathbf{U}^{(t)}} = -\mathcal{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right) + \mathbf{U}^{(t)}\Gamma^{(t)} + \lambda\mathbf{U}^{(t)}. \tag{17}$$

And (12) can be rewritten as

$$\begin{aligned}\mathbf{U}^{(t)} &\longleftarrow \mathbf{U}^{(t)} - \eta\frac{\partial \mathscr{L}}{\partial \mathbf{U}^{(t)}} \\ &= \mathbf{U}^{(t)}\left(\mathbf{I} - \eta\left(\Gamma^{(t)} + \lambda\mathbf{I}\right)\right) + \eta\mathcal{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right),\end{aligned} \tag{18}$$

where $\mathbf{I}$ is an identity matrix. Note that $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$ derived by (18) do not satisfy the first and second constraints in (9). To satisfy these two constraints, we can normalize each row of $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$ by

$$u_{n,k}^{(t)} \longleftarrow \frac{u_{n,k}^{(t)}}{\sum_{k=1}^{K} u_{n,k}^{(t)}}. \tag{19}$$

Furthermore, the pseudocode of SGDClus is given in Algorithm 1.

4.3.2. SOSClus. In SOSClus, we apply the second-order stochastic algorithm to the clustering problem in (9). According to (11), each factor matrix $\mathbf{U}^{(t)}$, for $t = 1, 2, \ldots, T$, is updated by the rule

$$\mathbf{U}^{(t)} \longleftarrow \mathbf{U}^{(t)} - \eta\left(\frac{\partial^2 \mathscr{L}}{\partial^2 \mathbf{U}^{(t)}}\right)^{-1}\frac{\partial \mathscr{L}}{\partial \mathbf{U}^{(t)}}. \tag{20}$$

According to (17), we can obtain the second-order partial derivative of $\mathscr{L}$ with respect to $\mathbf{U}^{(t)}$; that is,

$$\frac{\partial^2 \mathscr{L}}{\partial^2 \mathbf{U}^{(t)}} = \Gamma^{(t)} + \lambda\mathbf{I}. \tag{21}$$

By substituting (17) and (21) into (20), we have

$$\mathbf{U}^{(t)} \longleftarrow \left(1 - \eta\right)\mathbf{U}^{(t)} + \eta\mathcal{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right)\left(\Gamma^{(t)} + \lambda\mathbf{I}\right)^{-1}. \tag{22}$$

Input: $\mathcal{X}$, K, λ.
Output: $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$.
(1) Initialize $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$;
(2) Set iter $\leftarrow$ 1;
(3) **repeat**
(4) **for** $t \leftarrow 1$ **to** T **do**
(5) Set η;
(6) Compute $\mathbf{U}_{\text{opt}}^{(t)}$ according to (23)
(7) Update $\mathbf{U}^{(t)}$ according to (24);
(8) Normalize $\mathbf{U}^{(t)}$ according to (19);
(9) **end for**
(10) Set iter $\leftarrow$ iter + 1;
(11) **until** $\mathscr{L}(\mathcal{X}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(T)})$ unchanged or iter reaches the maximum iterations

Algorithm 2: The SOSClus algorithm.

Note that $\mathcal{X}_{(t)}(\odot^{(/t)}\mathbf{U})(\Gamma^{(t)}+\lambda\mathbf{I})^{-1}$ is the General Gradient-based Optimization (OPT) [39] solution for updating $\mathbf{U}^{(t)}$ in the regularized CP decomposition by making (17) equal to zero. See the details of proof in [39]. Let

$$\mathbf{U}_{\text{opt}}^{(t)} = \mathcal{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right)\left(\Gamma^{(t)} + \lambda\mathbf{I}\right)^{-1}. \tag{23}$$

Therefore, the updating rule of $\mathbf{U}^{(t)}$ in SOSClus is a weighted average of the current solution and the OPT solution intuitively; that is,

$$\mathbf{U}^{(t)} \longleftarrow \left(1 - \eta\right)\mathbf{U}^{(t)} + \eta\mathbf{U}_{\text{opt}}^{(t)}. \tag{24}$$

Actually, SOSClus is a general extension of General Gradient-based Optimization (OPT) [39] and the ALS with step size restriction in randomized block sampling method [41]. In (24), when the learning rate $\eta = 1$, we get the OPT solution. When the regularization parameter $\lambda = 0$, SOSClus becomes the ALS with step size restriction in randomized block sampling method.

Similar to SGDClus, $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$ derived by (24) in SOSClus also do not satisfy the first and second constraints in (9). We should normalize each row of $\{\mathbf{U}^{(t)}\}_{t=1}^{T}$ according to (19). The pseudocode of SOSClus is given in Algorithm 2.

4.4. Feasibility Analysis

Theorem 4. *The CP decomposition of $\mathcal{X}$ obtains the clustering of multityped objects in the heterogeneous information network $G = (V, E)$ simultaneously.*

Proof. Since the proofs for different types of objects in the heterogeneous information network $G = (V, E)$ are similar, we will simply describe the process for a single type of objects. Without loss of generality, we detail the proof on the tth type of objects.

Given a heterogeneous information network $G = (V, E)$ and its tensor representation $\mathcal{X}$, the nonzero elements in $\mathcal{X}$ represent the input gene-networks in $G = (V, E)$, which

we want to partition into K clusters $\{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_K\}$. The centre of the cluster $\mathscr{C}_k$ is denoted by $\mathbf{c}_k$. By using the coordinate format [47] as the sparse representation of $\mathscr{X}$, the gene-networks can be denoted as a matrix $\mathbf{M} \in \mathbb{R}^{nnz(\mathscr{X})\times T}$, where $nnz(\mathscr{X})$ is the number of nonzero elements in $\mathscr{X}$. Each row $\mathbf{m}_n \in \mathbf{M}, n = 1, 2, \ldots, nnz(\mathscr{X})$ gives the subscripts of corresponding nonzero element in $\mathscr{X}$. In other words, $\mathbf{m}_n$ represents a gene-network and the entries $m_{n,t} \in \mathbf{m}_n, t = 1, 2, \ldots, T$ are the subscripts of the objects contained in the gene-network.

The traditional clustering approach, such as K-means, minimizes the sum of differences between individual gene-network in each cluster and the corresponding cluster centres; that is,

$$\min_{p_{n,k},\mathbf{c}_k} \sum_{n=1}^{nnz(\mathscr{X})} \left\| \mathbf{m}_n - \sum_{k=1}^{K} p_{n,k}\mathbf{c}_k \right\|_F^2, \tag{25}$$

where $p_{n,k}$ is the probability of gene-network $\mathbf{m}_n$ belonging to the kth cluster. Also we can rewrite the problem by a new perspective of clustering individual object in the gene-network as follows:

$$\min_{p_{n_t,k},c_{k,t}} \sum_{n=1}^{nnz(\mathscr{X})} \sum_{t=1}^{T} \left\| m_{n,t} - \sum_{k=1}^{K} p_{n_t,k}c_{k,t} \right\|_F^2, \tag{26}$$

where $p_{n_t,k}$ is the probability of object v_n^t belonging to the kth cluster.

In the matrix form, K-means can be formalized as

$$\min_{\mathbf{P},\mathbf{C}} \|\mathbf{M} - \mathbf{P}\mathbf{C}\|_F^2, \tag{27}$$

where $\mathbf{P}$ is the cluster indication matrix and $\mathbf{C}$ is the cluster centres.

By matricization of $\mathscr{X}$ along the tth mode, the CP decomposition of $\mathscr{X}$ in (3) can be rewritten as

$$\min_{\mathbf{U}^{(1)},\mathbf{U}^{(2)},\ldots,\mathbf{U}^{(T)}} \left\| \mathscr{X}_{(t)} - \mathbf{U}^{(t)} \left(\odot^{(/t)}\mathbf{U}\right)^\top \right\|_F^2 \tag{28}$$

$\mathscr{X}_{(t)}$ is the matricization of $\mathscr{X}$ along the tth mode and $\mathbf{M}$ is the sparse representation of $\mathscr{X}$. Let $\mathbf{U}^{(t)} = \mathbf{P}$ and let $(\odot^{(/t)}\mathbf{U})^\top = \mathbf{C}$. That is, $\mathbf{U}^{(t)}$ is the cluster indication matrix for the tth type of objects and $(\odot^{(/t)}\mathbf{U})^\top$ is the cluster centres. So, the CP decomposition in (3) is equivalent to the K-means clustering for the tth type of objects in heterogeneous information network $G = (V, E)$.

By matricization of $\mathscr{X}$ in (3) along different modes, we can prove that the CP decomposition is equivalent to the K-means clustering for other types of objects. So, the CP decomposition of $\mathscr{X}$ obtains the clustering of multityped objects in the heterogeneous information network $G = (V, E)$ simultaneously. □

It is worth noting that the CP decomposition based clustering is a soft clustering method; the factor matrices indicate the probability of objects belonging to corresponding clusters. The soft clustering is more in line with reality, because many objects in the heterogeneous information networks may belong to several clusters. In other words, the clusters are overlapping. In some cases, the overlapping clusters need to be translated into nonoverlapping clusters, which can be achieved by using different approaches, such as K-means, to cluster the rows of factor matrices. Usually, the nonoverlapping clusters can be obtained by simply assigning each object to the cluster which has the largest entry in the corresponding row of factor matrix.

4.5. Time Complexity Analysis. The main time consumption of updating each factor matrix $\mathbf{U}^{(t)}$ in SGDClus is computing $\partial\mathscr{L}/\partial\mathbf{U}^{(t)}$. According to (17), we need to calculate $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U})$ and $\mathbf{U}^{(t)}\Gamma^{(t)}$, respectively. Since $\mathbf{U}^{(t)} \in \mathbb{R}^{N_t\times K}$ and $\mathscr{X}_{(t)} \in \mathbb{R}^{N_t\times\prod_{\substack{i=1\\i\neq t}}^{T} N_i}$, we have $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U}) \in \mathbb{R}^{N_t\times K}$, $\Gamma^{(t)} \in \mathbb{R}^{K\times K}$, and $\mathbf{U}^{(t)}\Gamma^{(t)\in\mathbb{R}^{N_t\times K}}$.

Firstly, if we successively calculate the Khatri-Rao product of $T-1$ matrices and a matrix-matrix multiplication when computing $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U})$, the intermediate results will be of very large size and the computational cost will be very expensive. In practice, we can reduce the complexity by ignoring the unnecessary calculation. Let us observe the element of $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U})$; that is,

$$\left(\mathscr{X}_{(t)}\left(\odot^{(/t)}\mathbf{U}\right)\right)_{n_t,k} = \sum_{\{n_i\}_{\substack{i=1\\i\neq t}}^{T}} \left(x_{n_t,\prod_{\substack{i=1\\i\neq t}}^{T} n_i} \prod_{\substack{i=1\\i\neq t}}^{T} u_{n_i,k}^{(i)} \right); \tag{29}$$

$x_{n_t,\prod_{\substack{i=1\\i\neq t}}^{T} n_i}$ is an element in the matricization of $\mathscr{X}$, which represents a corresponding gene-network in the heterogeneous information network. When $x_{n_t,\prod_{\substack{i=1\\i\neq t}}^{T} n_i} = 0$, we can ignore the following Khatri-Rao product. Hence, only nonzero elements in $\mathscr{X}$ need to be computed. Therefore, the time complexity for computing $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U})$ is $O(nnz(\mathscr{X})N_tK)$.

Secondly, the element of $\mathbf{U}^{(t)}\Gamma^{(t)}$ is given by

$$\left(\mathbf{U}^{(t)}\Gamma^{(t)}\right)_{n_t,k} = \sum_{j=1}^{K} \left(u_{n_t,j}^{(t)} \prod_{\substack{i=1\\i\neq t}}^{T} \sum_{n_i=1}^{N_i} u_{n_i,j}^{(i)} u_{n_i,k}^{(i)} \right). \tag{30}$$

So, the time complexity of computing $\mathbf{U}^{(t)}\Gamma^{(t)}$ is $O((N - N_t)K^2)$, where $N = \sum_{t=1}^{T} N_t$ is the total number of objects in the networks.

Above all, the time complexity of each iteration in SGDClus is $O(nnz(\mathscr{X})NK + (T-1)NK^2)$. Note that, in the real-world heterogeneous information networks, the number of clusters K and the number of object types T are usually far less than N; that is, $K \ll N$ and $T \ll N$.

According to (22), the time complexity of updating each factor matrix $\mathbf{U}^{(t)}$ in SOSClus is composed of three components: $\mathscr{X}_{(t)}(\odot^{(/t)}\mathbf{U})$, $(\Gamma^{(t)} + \lambda\mathbf{I})^{-1}$, and the product of them. Compared to SGDClus, only the time consumption of computing an inverse matrix for $(\Gamma^{(t)} + \lambda\mathbf{I})$ is additional in

SOSClus. Since $(\Gamma^{(t)} + \lambda \mathbf{I})$ is a $K \times K$ square matrix, computing the inverse of such matrix costs $O(K^3)$. Nevertheless, $O(K^3)$ is usually negligible because $K \ll N$.

5. Experiments and Results

In this section, we present several experiments on synthetic and real-world datasets for heterogeneous information networks and compare the performance with a number of state-of-the-art clustering methods.

5.1. Evaluation Metrics and Experimental Setting

5.1.1. Evaluation Metrics. In the experiments, we adopt the Normalized Mutual Information (NMI) [48] and Accuracy (AC) as our performance measurements.

NMI is used to measure the mutual dependence information between the clustering result and the ground truth. Given N objects, K clusters, one clustering result, and the ground truth classes for the objects, let $n(i, j)$, $i, j = 1, 2, \ldots, K$, be the number of objects that labeled i in clustering result but labeled j in the ground truth. The joint distribution can be defined as $p(i, j) = n(i, j)/N$, the marginal distribution of rows can be calculated as $p_1(j) = \sum_{i=1}^{K} p(i, j)$, and the marginal distribution of columns can be calculated as $p_2(i) = \sum_{j=1}^{K} p(i, j)$. Then, the NMI is defined as

$$\mathrm{NMI} = \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} p(i, j) \log\left(p(i, j) / p_1(j)\, p_2(i)\right)}{\sqrt{\sum_{j=1}^{K} p_1(j) \log p_1(j) \sum_{i=1}^{K} p_2(i) \log p_2(i)}}. \quad (31)$$

The NMI ranges from 0 to 1: the larger value of NMI, the better the clustering result.

AC is used to compute the clustering accuracy that measures the percent of the correct clustering result. AC is defined as

$$\mathrm{AC} = \frac{\sum_{t=1}^{T} \sum_{n=1}^{N_t} \delta\left(\mathrm{map}\left(v_n^t\right), \mathrm{label}\left(v_n^t\right)\right)}{\sum_{t=1}^{T} N_t}, \quad (32)$$

where $\mathrm{map}(v_n^t)$ is the cluster label of the object v_n^t and the $\mathrm{label}(v_n^t)$ is the ground truth class of the object v_n^t. And $\delta(\cdot)$ is an indicator function:

$$\delta(\cdot) = \begin{cases} 1 & \text{if } \mathrm{map}\left(v_n^t\right) = \mathrm{label}\left(v_n^t\right), \\ 0 & \text{if } \mathrm{map}\left(v_n^t\right) \neq \mathrm{label}\left(v_n^t\right). \end{cases} \quad (33)$$

Since both of NMI and AC are used to measure the performance of clustering one type of object, the weighted average NMI and AC are also used to measure the performance of STFClus and other state-of-the-art methods:

$$\overline{\mathrm{NMI}} = \frac{\sum_{t=1}^{T} N_t \left(\mathrm{NMI}\right)_t}{\sum_{t=1}^{T} N_t}, \qquad \overline{\mathrm{AC}} = \frac{\sum_{t=1}^{T} N_t \left(\mathrm{AC}\right)_t}{\sum_{t=1}^{T} N_t}. \quad (34)$$

Table 1: The synthetic datasets.

Synthetic datasets	T	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%

T is the number of object types in the heterogeneous information network and also the number of modes in the tensor. K is the number of clusters. S is the network scale, and $S = N_1 \times N_2 \times \cdots \times N_T$. D is the density of the tensor, that is, the percentage of nonzero elements in the tensor, and $D = nnz(\mathcal{X})/S$.

5.1.2. Experimental Setting. In order to compare the performance of our proposed SGDClus and SOSClus with others impartially, all methods share a common stopping condition, that is,

$$\frac{\left|\mathcal{L}_{\mathrm{iter}} - \mathcal{L}_{\mathrm{iter-1}}\right|}{\mathcal{L}_{\mathrm{iter-1}}} \leq 10^{-6}, \quad (35)$$

or iter reaches the maximum iterations. $\mathcal{L}_{\mathrm{iter}}$ and $\mathcal{L}_{\mathrm{iter-1}}$ are the values of function $\mathcal{L}$ at the current, that is, (iter)th, iteration, and the previous, that is, (iter − 1)th, iteration, respectively. And we set the maximum iterations to be 1000. Throughout the experiments, the regularization parameter λ in SGDClus and SOSClus is fixed as $\lambda = 0.001$.

All experiments are implemented in the MATLAB R2015a (version 8.5.0), 64-bit. And the MATLAB Tensor Toolbox (version 2.6, http://www.sandia.gov/~tgkolda/TensorToolbox/) is used in our experiments. Since the heterogeneous information networks are often sparse in real-world scenarios, that is, most elements in the tensor $\mathcal{X}$ are zeros, we use the sparse format of $\mathcal{X}$ as proposed in [47], which has been supported by MATLAB Tensor Toolbox. The experimental results are the average values obtained by running the algorithms ten times on corresponding datasets.

5.2. Experiments on Synthetic Datasets

5.2.1. The Synthetic Datasets Description. The purpose of using synthetic datasets is to examine whether the proposed tensor CP decomposition clustering framework can work well, since the detailed cluster structures of the synthetic datasets are known. In order to make the synthetic datasets similar to a realistic situation, we assume that the distribution for different types of objects that appear in a gene-network follows Zipf's law (see details online: https://en.wikipedia.org/wiki/Zipf's_law). Zipf's law is defined by $f(r; \rho, N) = r^{-\rho} / \sum_{n=1}^{N} n^{-\rho}$, where N is the number of objects, r is the object index, and ρ is the parameter characterizing the distribution. Zipf's law denotes the frequency of the rth object appearing in the gene-network. We set $\rho = 0.95$ and generate 4 synthetic datasets with different parameters. The details of these synthetic datasets are shown in Table 1.

5.2.2. Experimental Results. In the beginning of the experiments, we set a common learning rate, $\eta = 1/(\mathrm{iter} + 1)$, for

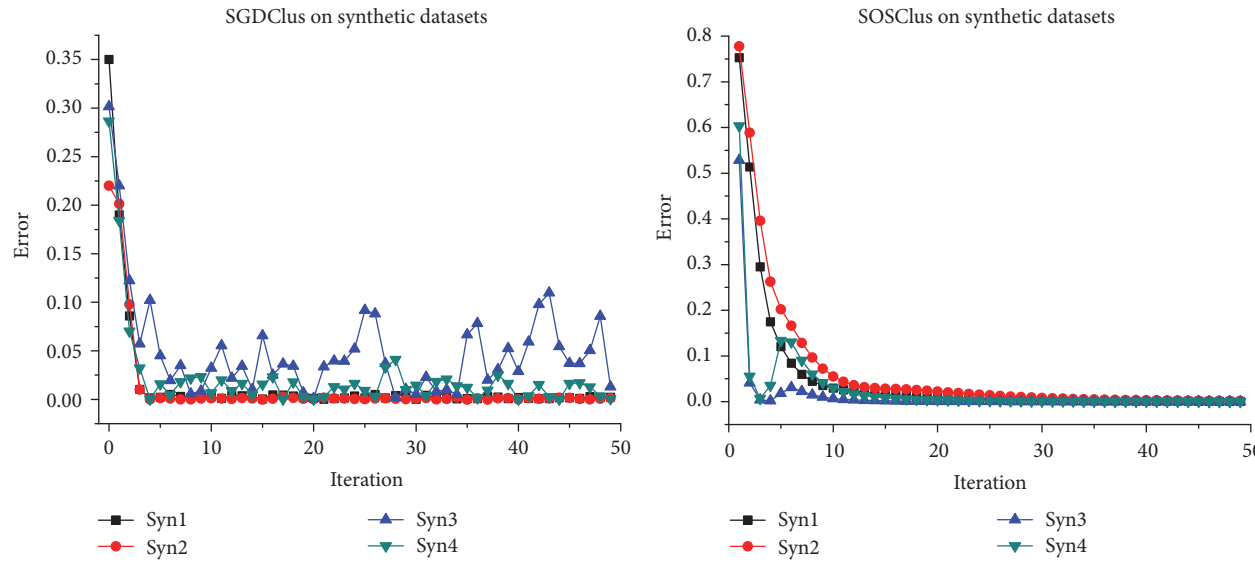

Figure 3: Convergence speed of SGDClus and SOSClus on the 4 synthetic datasets with a common learning rate, that is, $\eta = 1/(\text{iter} + 1)$.

SGDClus and SOSClus. We find that SOSClus has a faster convergence speed and better robustness with respect to the learning rate, which is clearly shown in Figure 3. Although SGDClus may eventually converge to a local minimum, the efficiency of the optimization near a local minimum is not all roses. As shown in Figure 3, the solutions of SGDClus swing around a local minimum. This phenomenon proves that the convergence speed of SGDClus is sensitive to the choice of learning rate η.

Then, we modify the learning rate to $\eta = 1/(\text{iter} + c)$ for SGDClus, where c is a constant optimized in the experiments. In practice, $c = 27855$ for SGDClus running on Syn3 and $c = 430245$ for SGDClus running on Syn4. The performance comparison of SGDClus with learning rate $\eta = 1/(\text{iter} + c)$ and SOSClus with learning rate $\eta = 1/(\text{iter} + 1)$ on Syn3 and Syn4 is shown in Figure 4. By employing the optimized learning rate, SGDClus converges to a local minimum quickly. However, compared to SOSClus, SGDClus still has no advantage. The hand drawing blue circles over the curve of SOSClus in Figure 4 shows that SOSClus can escape from a local minimum and find the global minimum, while SGDClus just obtains the first reaching local minimum.

According to (23) and (24), we accessorily obtain the solutions of OPT by running SOSClus. So, we compare the AC and NMI of OPT, SGDClus, and SOSClus on the 4 synthetic datasets, which are shown in Figure 5. With the increase of object types in the heterogeneous information networks, the AC and NMI of SOSClus and OPT increase distinctly, while performance of SGDClus almost does not change. When $T = 2$, AC and NMI of these three methods on Syn1 and Syn2 are almost equal and low. However, AC and NMI of SOSClus increase to 1 when $T = 4$. Since the histograms of OPT, SGDClus, and SOSClus on Syn1 and Syn2 are almost the same and on Syn3 and Syn4 are also similar, we know that the parameters K and S have no significant effect on the performance. Generally, the larger density D and the number of object types T in the network result in higher AC and NMI of SOSClus.

Obviously, in the experiments on the 4 synthetic datasets, SOSClus shows an excellent performance. SOSClus has a faster convergence speed and better robustness with respect to the learning rate. Meanwhile, SOSClus performs better on AC and NMI, because it can escape from a local minimum and find the global minimum.

5.3. Performance Comparison on Real-World Dataset

5.3.1. Real-World Dataset Description. The experiments on the real-world dataset are used to compare the performance of the tensor CP decomposition clustering framework with other state-of-the-art methods.

The real-world dataset extracted from the DBLP database is the DBLP-four-area dataset, which can be downloaded from http://web.cs.ucla.edu/~yzsun/data/DBLP_four_area.zip. It is a four research-area subset of DBLP and is used in [2, 3, 12, 13, 15, 16, 18]. The four research areas in DBLP-four-area dataset are database (DB), data mining (DM), machine learning (ML), and information retrieval (IR), respectively. There are five representative conferences in each area. And all related authors, papers published in these conferences, and terms contained in these papers' titles are included. The DBLP-four-area dataset contains 14,376 papers with 100 labeled, 14,475 authors with 4,057 labeled, 20 labeled conferences, and 8,920 terms. The density of the DBLP-four-area dataset is 9.01935×10^{-9}, so we construct a 4-mode tensor with size of $14{,}376 \times 14{,}475 \times 20 \times 8{,}920$ and 334832 nonzero elements. We compare the performance of tensor CP decomposition clustering framework with several other methods on the labeled record in this dataset.

5.3.2. Comparative Methods

(i) NetClus (see [2]). An extended version of RankClus [1], which can deal with the network, follows the star network schema. The time complexity of NetClus for clustering each

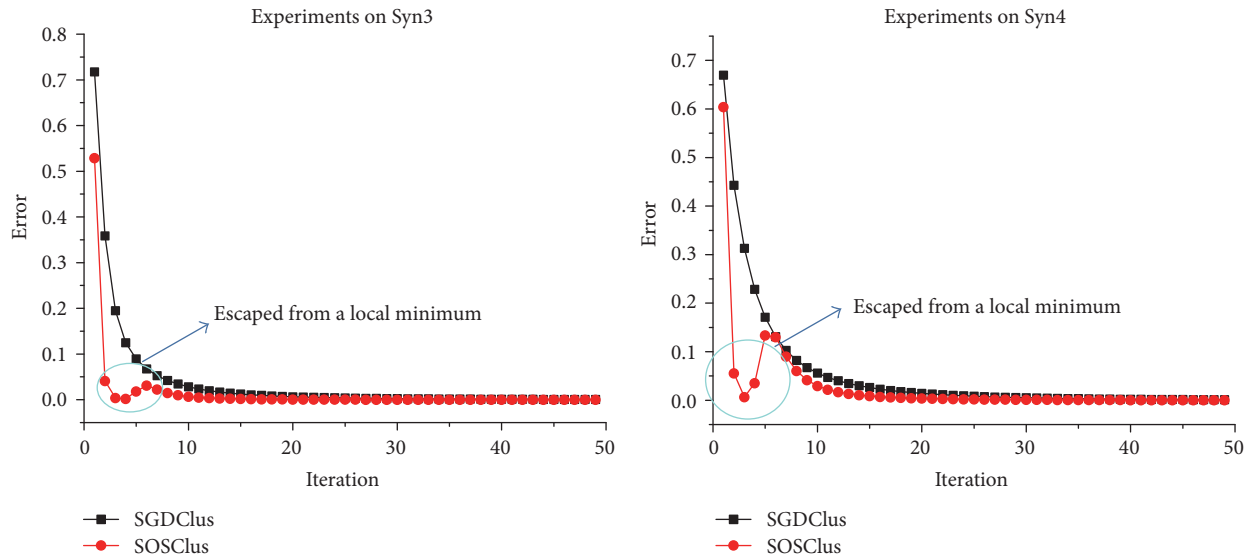

Figure 4: Convergence speed of SGDClus and SOSClus on Syn3 and Syn4 with different learning rate. The learning rate for SOSClus is $\eta = 1/(\text{iter} + 1)$, while that for SGDClus is $\eta = 1/(\text{iter} + c)$, where c is a constant optimized in the experiments.

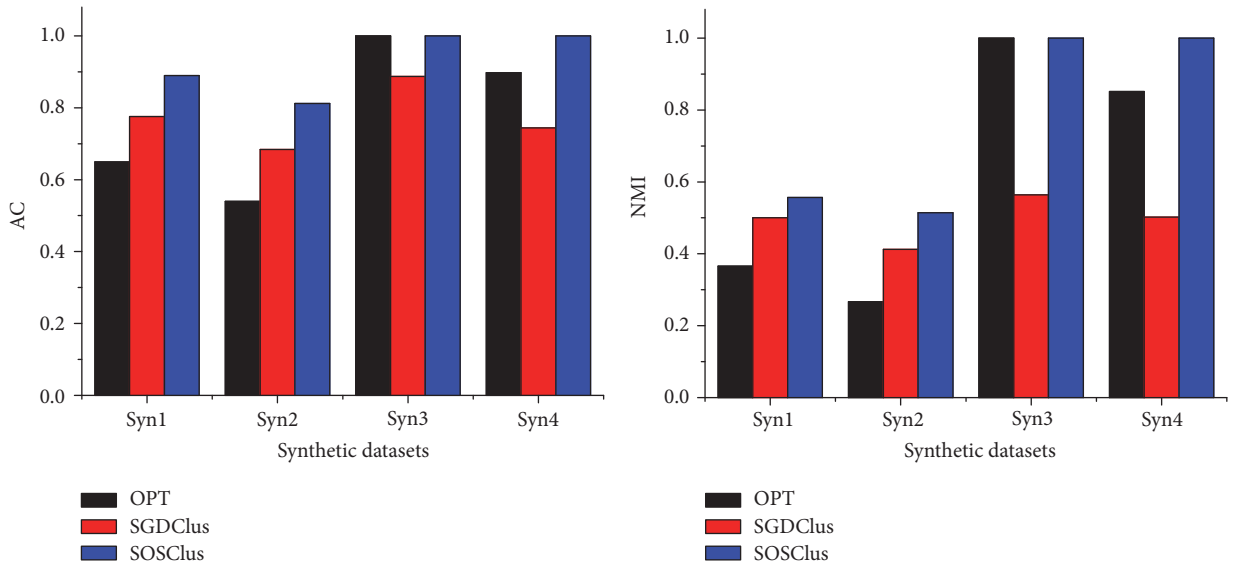

Figure 5: The AC and NMI of OPT, SGDClus, and SOSClus on the 4 synthetic datasets.

object type in each iteration is $O(K|E| + (K^2 + K)N)$, where K is the number of clusters, $|E|$ is the number of edges in the network, and N is the total number of objects in the network.

(ii) PathSelClus (see [15, 16]). A clustering method based on the predefined metapath requires a user guide. In PathSelClus, the distance between the same type objects is measured by PathSim [3], and the method starts with the given seeds by user. The time complexity of PathSelClus for clustering each object type in each iteration is $O((K + 1)|\mathscr{P}| + KN)$, where $|\mathscr{P}|$ is the number of metapath instances in the network. And, the time complexity of PathSim used by PathSelClus for clustering each object type is $O(Nd)$, where d is the average degree of objects.

(iii) FctClus (see [13]). It is a recently proposed clustering method for heterogeneous information networks. As with NetClus, the FctClus method can deal with networks following the star network schema. The time complexity of FctClus for clustering each object type in each iteration is $O(K|E| + TKN)$.

5.3.3. Experimental Results. As the baseline methods can only deal with specific schema heterogeneous information networks, here we must construct different subnetworks for them. For NetClus and FctClus, the network is organized as a star network schema like in [2, 13], where the paper (P) is the centre type, and author (A), conference (C), and term (T) are

Table 2: AC of experiments on DBLP-four-area dataset.

AC	OPT	SGDClus	SOSClus	NetClus	PathSelClus	FctClus
Paper	0.5882	0.8476	**0.9007**	0.7154	0.7551	0.7887
Author	0.5872	0.8486	**0.9486**	0.7177	0.7951	0.8008
Conference	**1**	0.99	**1**	0.9172	0.9950	0.9031
avg (AC)	0.5892	0.8493	**0.9477**	0.7186	0.7951	0.8010

Table 3: NMI of experiments on DBLP-four-area dataset.

NMI	OPT	SGDClus	SOSClus	NetClus	PathSelClus	FctClus
Paper	0.6557	0.6720	**0.8812**	0.5402	0.6142	0.7152
Author	0.6539	**0.8872**	0.8822	0.5488	0.6770	0.6012
Conference	**1**	0.8497	**1**	0.8858	0.9906	0.8248
avg (NMI)	0.6556	0.8778	**0.8827**	0.5503	0.6770	0.6050

Table 4: Running time of experiments on DBLP-four-area dataset.

Running time (s)	OPT	SGDClus	SOSClus	NetClus	PathSelClus	FctClus
Paper	—	—	—	802.6	542.3	808.4
Author	—	—	—	743.7	681.1	774.9
Conference	—	—	—	658.4	629.3	669.8
Total time	672.6	**432.4**	818.4	2204.7	1852.7	2253.1

the attribute types. For PathSelClus, we select the metapath of P-T-P, A-P-C-P-A, and C-P-T-P-C to cluster the papers, authors, and conferences, respectively. And in PathSelClus, we give each cluster one seed to start.

We model the DBLP-four-area dataset as a 4-mode tensor, where each mode represents one object type. The 4 modes are author (A), paper (P), conference (C), and term (T), respectively. Actually, the sequence of the object types is insignificant. And each element of the tensor represents a gene-network in the heterogeneous information network. In the experiments, we set the learning rate for SOSClus to be $\eta = 1/(\text{iter}+1)$ and an optimized learning rate $\eta = 1/(\text{iter}+c)$ with $c = 1000125$ for SGDClus. By running SOSClus, we accessorily obtain the solutions of OPT. So, we compare the experimental results of OPT, SGDClus, and SOSClus on the DBLP-four-area dataset with the three baseline methods. See the details in Tables 2, 3, and 4.

In Tables 2 and 3, SOSClus performs best on average AC and NMI, and SGDClus takes the second place. All methods achieve satisfactory AC and NMI on the conference, since there are only 20 conferences in the network. SGDClus takes the shortest running time, and OPT and SOSClus have an obvious advantage on running time compared with other baselines. The time complexity of SGDClus is $O(nnz(\mathcal{X})NK + (T-1)K^2N)$, and the time complexity of SOSClus is $O(nnz(\mathcal{X})NK + (T-1)K^2N + K^3)$, where $nnz(\mathcal{X})$ is the number of nonzero elements in $\mathcal{X}$, that is, the number of gene-networks in the heterogeneous information network. We have $nnz(\mathcal{X}) < |\mathcal{P}| \ll |E|$, $K \ll N$, and $T \ll N$. Compared with the time complexity of the three baselines, SGDClus and SOSClus have a little disadvantage. However, it is worth noting that the three baselines can only cluster one type of objects in the network in each running, while OPT, SGDClus, and SOSClus can obtain the clusters of all types of objects simultaneously by running once. This is the reason why only the total time is shown for OPT, SGDClus, and SOSClus in Table 4. Moreover, the total time of OPT, SGDClus, and SOSClus is on the same order of magnitude as the running time of other baselines for clustering each type of objects, which is consistent with the comparison of the time complexity.

6. Conclusion

In this paper, a tensor CP decomposition method for clustering heterogeneous information networks is presented. In tensor CP decomposition clustering framework, each type of objects in heterogeneous information network is modeled as one mode of tensor, and the gene-networks in the network are modeled as the elements in tensor. In other words, tensor CP decomposition clustering framework can model different types of objects and semantic relations in the heterogeneous information network without the restriction of network schema. In addition, two stochastic gradient descent algorithms, named SGDClus and SOSClus, are designed. SGDClus and SOSClus can cluster all types of objects and the gene-networks simultaneously by running once. The proposed algorithms outperformed other state-of-the-art clustering methods in terms of AC, NMI, and running time.

Notations

a: A scalar (lowercase letter)
$\mathbf{a}$: A vector (boldface lowercase letter)
$\mathbf{A}$: A matrix (boldface capital letter)
$\mathcal{X}$: A tensor (calligraphic letter)

$x_{i_1,i_2,\ldots,i_N}$: The $(i_1, i_2, \ldots, i_N)$th element of an Nth-order tensor $\mathcal{X}$
$\mathcal{X}_{(n)}$: Matricization of $\mathcal{X}$ along the nth mode
$*$: Hadamard product (element-wise product) of two tensors (or matrices or vectors) with the same dimension
$\odot$: Khatri-Rao product of two matrices
$\|\cdot\|_F$: Frobenius norm of a tensor (or matrices or vectors)
$\mathbf{a} \circ \mathbf{b}$: Outer product of two vectors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

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

References

[1] Y. Sunt, J. Hant, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "RankClus: integrating clustering with ranking for heterogeneous information network analysis," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '09)*, pp. 565–576, Saint Petersburg, Russia, March 2009.

[2] 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. 797–805, Paris, France, July 2009.

[3] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: meta path-based top-K similarity search in heterogeneous information networks," *PVLDB*, vol. 4, no. 11, pp. 992–1003, 2011.

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[5] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[6] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, E. Simoudis, J. Han, and U. M. Fayyad, Eds., pp. 226–231, AAAI Press, Portland, Ore, USA, 1996.

[7] W. Wang, J. Yang, and R. R. Muntz, "STING: a statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB '97)*, M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, Eds., pp. 186–195, Morgan Kaufmann, Athens, Greece, August 1997, http://www.vldb.org/conf/1997/P186.PDF.

[8] E. H. Ruspini, "New experimental results in fuzzy clustering," *Information Sciences*, vol. 6, no. 73, pp. 273–284, 1973.

[9] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[10] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[11] J. Han, Y. Sun, X. Yan, and P. S. Yu, "Mining knowledge from data: an information network analysis approach," in *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE '12)*, pp. 1214–1217, IEEE, Washington, DC, USA, April 2012.

[12] J. Chen, W. Dai, Y. Sun, and J. Dy, "Clustering and ranking in heterogeneous information networks via gamma-poisson model," in *Proceedings of the SIAM International Conference on Data Mining (SDM '15)*, pp. 424–432, May 2015.

[13] J. Yang, L. Chen, and J. Zhang, "FctClus: a fast clustering algorithm for heterogeneous information networks," *PLoS ONE*, vol. 10, no. 6, Article ID e0130086, 2015.

[14] C. Shi, R. Wang, Y. Li, P. S. Yu, and B. Wu, "Ranking-based clustering on general heterogeneous information networks by network projection," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*, pp. 699–708, ACM, Shanghai, China, November 2014.

[15] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 1348–1356, ACM, Beijing, China, August 2012.

[16] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "PathSelClus: integrating meta-path selection with user-guided Object clustering in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 7, no. 3, pp. 723–724, 2013.

[17] X. Yu, Y. Sun, B. Norick, T. Mao, and J. Han, "User guided entity similarity search using meta-path selection in heterogeneous information networks," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, pp. 2025–2029, ACM, November 2012.

[18] Y. Sun, C. C. Aggarwal, and J. Han, "Relation strength-aware clustering of heterogeneous information networks with incomplete attributes," *Proceedings of the VLDB Endowment*, vol. 5, no. 5, pp. 394–405, 2012.

[19] M. Zhang, H. Hu, Z. He, and W. Wang, "Top-k similarity search in heterogeneous information networks with x-star network schema," *Expert Systems with Applications*, vol. 42, no. 2, pp. 699–712, 2015.

[20] Y. Zhou and L. Liu, "Social influence based clustering of heterogeneous information networks," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 338–346, ACM, Chicago, Ill, USA, August 2013.

[21] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[22] R. A. Harshman, "Foundations of the parafac procedure: model and conditions for an 'explanatory' multi-mode factor analysis," UCLA Working Papers in Phonetics, 1969.

[23] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *Journal of Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.

[24] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[25] A. Cichocki, D. Mandic, L. De Lathauwer et al., "Tensor decompositions for signal processing applications: from two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[26] W. Peng and T. Li, "Tensor clustering via adaptive subspace iteration," *Intelligent Data Analysis*, vol. 15, no. 5, pp. 695–713, 2011.

[27] J. Hastad, "Tensor rank is NP-complete," *Journal of Algorithms. Cognition, Informatics and Logic*, vol. 11, no. 4, pp. 644–654, 1990.

[28] S. Metzler and P. Miettinen, "Clustering Boolean tensors," *Data Mining & Knowledge Discovery*, vol. 29, no. 5, pp. 1343–1373, 2015.

[29] 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.

[30] I. Sutskever, R. Salakhutdinov, and J. B. Tenenbaum, "Modelling relational data using Bayesian clustered tensor factorization," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 1821–1828, British Columbia, Canada, December 2009.

[31] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining & Knowledge Discovery*, vol. 29, no. 1, pp. 203–236, 2015.

[32] A. R. Benson, D. F. Gleiche, and J. Leskovec, "Tensor spectral clustering for partitioning higher-order network structures," in *Proceedings of the SIAM International Conference on Data Mining (SDM '15)*, pp. 118–126, Vancouver, Canada, May 2015.

[33] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proceedings of the 10th SIAM International Conference on Data Mining (SDM '10)*, pp. 211–222, Columbus, Ohio, USA, May 2010.

[34] E. E. Papalexakis, L. Akoglu, and D. Ience, "Do more views of a graph help? Community detection and clustering in multi-graphs," in *Proceedings of the 16th International Conference of Information Fusion (FUSION '13)*, pp. 899–905, Istanbul, Turkey, July 2013.

[35] M. Vandecappelle, M. Bousse, F. V. Eeghem, and L. D. Lathauwer, "Tensor decompositions for graph clustering," Internal Report 16-170, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2016, ftp://ftp.esat.kuleuven.be/pub/SISTA/sistakulak/reports/2016_Tensor_Graph_Clustering.pdf.

[36] W. Shao, L. He, and P. S. Yu, "Clustering on multi-source incomplete data via tensor modeling and factorization," in *Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '15)*, pp. 485–497, Ho Chi Minh City, Vietnam, 2015.

[37] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[38] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1, R2, . . . , RN) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 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] S. Hansen, T. Plantenga, and T. G. Kolda, "Newton-based optimization for Kullback-Leibler nonnegative tensor factorizations," *Optimization Methods & Software*, vol. 30, no. 5, pp. 1002–1029, 2015.

[41] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE Journal on Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 284–295, 2016.

[42] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," http://arxiv.org/abs/1503.02101.

[43] T. G. Kolda, "Multilinear operators for higher-order decompositions," Tech. Rep. SAND2006-2081, Sandia National Laboratories, 2006, http://www.osti.gov/scitech/biblio/923081=0pt.

[44] P. Paatero, "A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis," *Chemometrics & Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 223–242, 1997.

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

[46] B. L. Bottou and N. Murata, "Stochastic approximations and efficient learning," in *The Handbook of Brain Theory and Neural Networks*, 2nd edition, 2002.

[47] B. W. Bader and T. G. Kolda, "Efficient MATLAB computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, 2007.

[48] 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.