PAPER *Special Section on Data Engineering and Information Management*

# Detecting Communities and Correlated Attribute Clusters on Multi-Attributed Graphs

**Hiroyoshi ITO**[†a)], ***Nonmember***, **Takahiro KOMAMIZU**[††b)], **Toshiyuki AMAGASA**[†††c)], ***Members***, ***and*** **Hiroyuki KITAGAWA**[†††d)], ***Fellow***

**SUMMARY** Multi-attributed graphs, in which each node is characterized by multiple types of attributes, are ubiquitous in the real world. Detection and characterization of communities of nodes could have a significant impact on various applications. Although previous studies have attempted to tackle this task, it is still challenging due to difficulties in the integration of graph structures with multiple attributes and the presence of noises in the graphs. Therefore, in this study, we have focused on clusters of attribute values and strong correlations between communities and attribute-value clusters. The graph clustering methodology adopted in the proposed study involves **C**ommunity detection, **A**ttribute-value clustering, and deriving **R**elationships between communities and attribute-value clusters (CAR for short). Based on these concepts, the proposed multi-attributed graph clustering is modeled as CAR-clustering. To achieve CAR-clustering, a novel algorithm named CARNMF is developed based on non-negative matrix factorization (NMF) that can detect CAR in a cooperative manner. Results obtained from experiments using real-world datasets show that the CARNMF can detect communities and attribute-value clusters more accurately than existing comparable methods. Furthermore, clustering results obtained using the CARNMF indicate that CARNMF can successfully detect informative communities with meaningful semantic descriptions through correlations between communities and attribute-value clusters.

***key words:*** *clustering, community detection, non-negative matrix factorization*

## 1. Introduction

Community detection is a task to detect densely connected subgraphs as communities. Nodes in a community tend to share same or similar properties, such phenomenon is called *homophily effect* [1], [2], meaning that nodes having similar properties tend to link together. Because diverse applications are derived from the nature of real communities, community detection is important in graph/network analyses. Examples include node property estimations [3]–[5], community-wise information recommendations [6], and semantic reasoning for nodes/edges [7].

Moreover, using the attributes in a graph is advantageous to realize high-quality community detection as well as to understand the characteristics of communities. Multi-attributed graphs are reasonable models of real-world networks such as social networks, co-author networks, protein-protein interaction networks, etc. In fact, several works have proposed algorithms that employ attribute information (i.e., shared interests or functional behaviors of each community) to detect not only communities but also their semantic meanings [8]–[11].

However, community detection and extraction of semantics in multi-attributed graphs remain challenging due to difficulties on integrating graph structures and multiple attributes of different types. Community detection and extraction of semantics involve multiple steps. First, useful information from each attribute must be extracted because certain node attributes describe different aspects. Second, all extracted information must be exploited to enhance community detection by effectively integrating heterogeneous information. Notice that the previous works [8]–[11] do not differentiate multiple attributes, that is, they consider multiple attributes equally. Moreover, real-world graphs are often incomplete and noisy. That is, some edges or nodes may be missing or attribute values may contain incorrect values, leading to inappropriate results.

To overcome these difficulties, we propose a novel clustering scheme based on the following two assumptions:
(1) *Relevant attribute values form clusters by attribute type.* This is based on the observation that an attribute reflects a node's interests in a network. Hence, an attribute tends to be associated to a specific group of values related to an interest. For example, in a co-author network where the nodes correspond to the authors (researchers), each author typically has specific research interests (e.g., AI, data mining, and database). Thus, attributes (e.g., paper title and conference) present biased values according to interests. Consequently, it is possible to identify clusters of attributes values (attribute-value clusters) reflecting a node's interests.
(2) *Communities are strongly correlated with attribute-value clusters.* This is related to the previous assumption. Consider the example above. The nodes in a community share similar interests (e.g., research interests) and consequently, similar attribute-value clusters (e.g., research topics, and conferences). Conversely, if some nodes (researchers) have similar attribute values, they should share similar interests and can be grouped in the same commu-

Manuscript received June 27, 2018.
Manuscript revised October 17, 2018.
Manuscript publicized February 4, 2019.

[†] The author is with the Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305–8577 Japan.

[††] The author is with Information Technology Center, Nagoya University, Nagoya-shi, 464–8601 Japan.

[†††] The authors are with the Center for Computational Sciences, University of Tsukuba, Tsukuba-shi, 305–8577 Japan.

a) E-mail: hiro.3188@kde.cs.tsukuba.ac.jp
b) E-mail: taka-coma@acm.org
c) E-mail: amagasa@cs.tsukuba.ac.jp
d) E-mail: kitagawa@cs.tsukuba.ac.jp

DOI: 10.1587/transinf.2018DAP0022

nity.

Exploiting the correlation between communities and multiple attributes should improve the quality of community detection as well as attribute-value clustering. Using the information from different sources (attributes) to alleviate the effect of noise (e.g., missing values and errors), we simultaneously implement community detection and attribute-value clustering.

Based on the aforementioned ideas, we study a novel clustering scheme for multi-attributed graphs, called CAR-clustering. CAR includes **C**ommunity detection, **A**ttribute-value clustering, and deriving **R**elationships between communities and attribute-value clusters for multi-attributed graphs. Additionally, we develop a novel clustering algorithm called CARNMF, which employs a non-negative matrix factorization (NMF).

The contributions of this paper are summarized as follows:

- We propose a novel clustering scheme CAR-clustering to address two technical questions. (i) Given a multi-attributed graph, how can community detection and attribute-value clustering be performed for different types of attributes in a cooperative manner? (ii) How should reasonable relationships be determined between communities and attribute-value clusters for each type of attribute?
- We develop a novel algorithm CARNMF, which achieves CAR-clustering. Specifically, a dedicated loss function is designed to perform multiple NMFs simultaneously.
- We conduct experiments using real-world datasets (DBLP computer science bibliography and arXiv physics bibliography). The accuracy of CARNMF with respect to community detection and attribute-value clustering and a comparison to other methods are examined. Relative to comparative methods, CARNMF achieves a better accuracy of up to 11% for community detection and up to 22% for attribute-value clustering. Furthermore, CARNMF detects informative communities and their rich semantic descriptions by correlating multiple types of attribute-value clusters.

A preliminary version of this paper appeared in [12]. More surveys, detailed explanation of proposed method and experiments are included in the journal version. The rest of the paper is organized as follows: In Sect. 2, we summarize the related works. We provide formal deffinitions of input graph model and our research objectives in Sect. 3. We propose our method CARNMF in Sect. 4. We examine CARNMF in several experiments in Sect. 5 and conclude the article in Sect. 6.

## 2. Related Work

Community detection in graphs is a current topic of interest in graph analysis and AI research. Existing works for non-attributed graphs can be categorized according to the techniques used: graph separation [4], [13], probabilistic generative model [14], and matrix factorization [5], [15], [16]. [4] defined *modularity*, which indicates how separated a community is from other nodes. More comprehensive surveys can be found in [17], [18].

Recently, several works have addressed the problem of detecting communities and their semantic descriptions on node-attributed graphs. [10] proposed *CESNA*, where communities and their attributes are simultaneously detected in an efficient manner. [8] proposed *SCI* to detect communities and their semantics using NMF. [9] proposed a probabilistic generative model called the *author-topic model* to model communities and related topics. [19] proposed *COMODO* to detect communities with shared properties using subgroup discovery techniques. [20] proposed a method for detecting communities and their descriptions from an attributed graph where detection of communities and induction of description are alternated. [21] proposed a joint community profiling and detection model which characterizes communities with user published contents and user diffusion links. Likewise, [11] proposed *LCTA*, where communities and their topics are modeled separately, and then their relationships are modeled using a probabilistic generative model. A comprehensive survey over these works can be found in [22].

The aforementioned works only consider single textual attributes or uniformly handle multiple attributes without any distinction. In reality, each attribute represents different aspects of the nodes. In our research, we deal with heterogeneous attributes individually. In addition to community detection, we perform clustering over attribute values for each attribute, which, in turn, can be used to improve the quality of communities detected.

Some works have investigated clustering over networks containing different types of nodes and/or edges. [23] studied community detection with characterization from multi-dimensional networks, which is defined as a graph consisting of a set of nodes and multiple types of edges. [24] studied subgraph detection from multi-layer graphs with edge labels. In contrast, we assume a different model where each node is characterized by multiple attributes. As we shall see later, we model multiple attributes using different types of nodes, and community detection as well as attribute-value clusterings can be described on such a graph consisting of different types of nodes (nodes and multiple types of attribute values), and try to detect communities over the nodes as well as the clusters over other types of attribute values. [25] proposed a scheme of ranking-based clustering for multi-typed heterogeneous networks, where two or more types of nodes are included. Similarly, [26] proposed an NMF-based method for such networks. These methods differ from ours in that they define a cluster consisting of all types of nodes. In other words, these methods cannot handle each attribute in a unique way. In contrast, our work deals with different attributes individually, but solves community detection and attribute-value clustering in a unified manner.

## 3. Problem Statement

In this work, we deal with multi-attributed graphs, where each node is characterized by two or more attributes. Given such a graph, *CAR-clustering* is used to solve the following three sub-problems: community detection, attribute-value clustering, and derivation of relationships between communities and attribute-value clusters, which have been independently studied. Below, we provide the formal definitions which are necessary to define the clustering scheme.

### 3.1 Multi-Attributed Graph

Multi-attributed graph $\mathbb{G}$ is defined by extending weighted graph $\mathbb{G}'$ with several attributed graphs $\mathbb{G}_t$ for attribute $t \in \mathbb{T}$. The following are formal definitions.

**Definition 1** (Weighted graph): Weighted graph $\mathbb{G}'$ is defined by a triplet, $\langle \mathbb{V}, \mathbb{E}, \mathbb{W} \rangle$, where $\mathbb{V}$ is a set of nodes, $\mathbb{E}(\subseteq \mathbb{V} \times \mathbb{V})$ is a set of edges, and $\mathbb{W} : \mathbb{E} \rightarrow \mathbb{R}^{+}$ is a map of edge weights. □

**Definition 2** (Attributed graph): Attributed graph $\mathbb{G}_t = \langle \mathbb{V} \cup \mathbb{X}_t, \mathbb{E}_t, \mathbb{W}_t \rangle$ of attribute $t \in \mathbb{T}$ is a bipartite graph consisting of set $\mathbb{V}$ of nodes, set $\mathbb{X}_t$ of attribute-values, a set of edges $\mathbb{E}_t \subseteq \mathbb{V} \times \mathbb{X}_t$, and $\mathbb{W}_t : \mathbb{E}_t \rightarrow \mathbb{R}^{+}$ is a map of edge weights. □

**Definition 3** (Multi-attributed graph): Given weighted graph $\mathbb{G}' = \langle \mathbb{V}, \mathbb{E}, \mathbb{W} \rangle$ and a set of attributed graphs $\{\mathbb{G}_t\}_{t \in \mathbb{T}}$ where $\mathbb{G}_t = \langle \mathbb{V} \cup \mathbb{X}_t, \mathbb{E}_t, \mathbb{W}_t \rangle$, multi-attributed graph $\mathbb{G} = \langle \mathbb{G}', \{\mathbb{G}_t\}_{t \in \mathbb{T}} \rangle$ is a union of these graphs. □

### 3.2 CAR-Clustering

Given a multi-attributed graph, information can be extracted from different perspectives. In this work, we extract *communities*, *attribute-value clusters*, and the *relationship* between them.

**Community.** For a multi-attributed graph, a set of nodes with the following properties is regarded as a community. (1) Nodes in a community are densely connected with each other and sparsely connected with other nodes. (2) Nodes in a community tend to share common values in distinct attributes. This study assumes that communities can overlap. That is, each node belongs to more than one community. This assumption is reasonable for real applications. Formally, given the number of communities $\ell$, node $n \in \mathbb{V}$ belonging to community $c \in \mathbb{C}$ is described by probability distribution $p(n \mid c)$, where $|\mathbb{C}| = \ell$.

**Attribute-value cluster.** For attribute $t \in \mathbb{T}$ in a multi-attributed graph, similar or highly correlated attribute values can be grouped into attribute-value clusters. Herein, we assume overlapping clusters. That is, each attribute-value belongs to more than one cluster. Formally, given the number of clusters $k_t$ of attribute $t \in \mathbb{T}$, cluster member $x \in \mathbb{X}_t$ for attribute-value cluster $s_t \in \mathbb{S}_t$ is described by probability distribution $p(x \mid s_t)$, where $|\mathbb{S}_t| = k_t$.

**Relationship between a community and an attribute-value cluster.** Nodes in a community often share common attribute-value clusters. Detecting such relationship is useful in many applications. Given community $c \in \mathbb{C}$ and attribute-value cluster $s_t \in \mathbb{S}_t$ of attribute $t \in \mathbb{T}$, the probability that $c$ is related to $s_t$ is described as the relationship between $c$ and $s_t$. In this work, a community may be related to more than one attribute-value cluster. Formally, this is described by probability distribution $p(s_t \mid c)$.

**CAR-clustering.** CAR-clustering is formally defined by Definition 4.

**Definition 4** (CAR-clustering): Given a multi-attributed graph $\mathbb{G}$, CAR-clustering is to perform community detection, attribute-value clustering, and detection of the relationship between the communities and the attribute-value clusters simultaneously. □

Solving these sub-problems simultaneously is more beneficial than evaluating each one independently because, in many cases, communities and attribute-value clusters are mutually correlated. Solving the problems simultaneously exploits this correlation, leading to improved results.

## 4. CARNMF – Algorithm for CAR-Clustering

In this section, we propose an NMF (non-negative matrix factorization)-based algorithm, called CARNMF, for CAR-clustering. CARNMF models communities and attribute-value clusters. Additionally, we introduce an auxiliary matrix to maintain the relationship between the communities and the attribute-value clusters. A unified loss function is used to solve the different NMFs in a unified manner. It is assumed that the user gives the number $\ell$ of communities and the number $k_t$ of clusters for each attribute $t \in \mathbb{T}$.

### 4.1 Matrix Representation

We represent a multi-attributed graph by two sorts of matrices: an adjacency matrix $A \in \mathbb{R}^{|\mathbb{V}| \times |\mathbb{V}|}$ and attribute matrices $X^{(t)} \in \mathbb{R}^{|\mathbb{V}| \times |\mathbb{X}_t|}$ for $t \in \mathbb{T}$. An element $A_{u,v}$ of $A$ corresponds to an edge $e_{u,v} = (u, v) \in \mathbb{E}$. $A_{u,v} = \mathbb{W}(e_{u,v}) / \sum_{e_{i,j} \in \mathbb{E}} \mathbb{W}(e_{i,j})$, indicating the joint probability for the presence of edge $e_{u,v}$. Similarly, for $t \in \mathbb{T}$, an element $X^{(t)}_{u,x}$ in $X^{(t)}$ corresponds to an edge $e^{(t)}_{u,x} \in \mathbb{E}_t$. $X^{(t)}_{u,x} = \mathbb{W}_t(e^{(t)}_{u,x}) / \sum_{v,y \in \mathbb{E}_t} \mathbb{W}_t(e^{(t)}_{v,y})$, indicating the joint probability of the presence of edge $e^{(t)}_{u,x}$.

### 4.2 Loss Function

We achieve CAR-clustering in terms of several NMFs, which correspond to the aforementioned sub-problems. To achieve CAR-clustering, we introduce loss functions for the sub-problems followed by a unified loss function.

**Loss function for community detection.** In CARNMF, communities $\mathbb{C}$ are denoted by a matrix $U^{*} \in \mathbb{R}^{|\mathbb{V}| \times \ell}$, where each row and column correspond to a node

$u \in \mathbb{V}$ and a community $c \in \mathbb{C}$, respectively. A cell $U^*_{u,c}$ represents joint probability of node $u$ and community $c$ $p(u, c)$. In probability $p(u, v, c)$, $u$ and $v$ are connected through community $c$, and is represented by $U^*_{u,c}U^*_{v,c}$. Moreover, joint probability $p(u, v)$, or the existence of edge $e_{u,v} \in \mathbb{E}$, is expressed as $\sum_{c\in\mathbb{C}} U^*_{u,c}U^*_{v,c}$. Therefore, when $U^*$ minimizes the following loss function, $U^*$ is the best approximation of the edges in the graph.

$$\underset{U^*\geq 0}{\arg\min} \left\| A - U^*(U^*)^T \right\|_F^2 , \tag{1}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm. Notice that this loss function is equivalent to the symmetric NMF based graph clustering [15].

**Loss function for attribute-value clustering.** In CARNMF, attribute-value clusters $\mathbb{S}_t$ of attribute $t \in \mathbb{T}$ are represented as a matrix $V^{(t)} \in \mathbb{R}^{|\mathbb{X}_t|\times k_t}$, where each row and column correspond to an attribute $x \in \mathbb{X}_t$ and an attribute cluster $s_t \in \mathbb{S}_t$, respectively. A cell $V^{(t)}_{x,s_t}$ represents probability $p(x \mid s_t)$.

To derive $V^{(t)}$ from $X^{(t)}$, we introduce a matrix $U^{(t)} \in \mathbb{R}^{|\mathbb{V}|\times k_t}$, which denotes the relationships between the nodes and attribute-value clusters with probability $p(u, s_t)$. Using both matrices $U^{(t)}$ and $V^{(t)}$, probability $p(u, x, s_t)$, which is the existence of edge $e^{(t)}_{u,x} \in \mathbb{E}_t$ in terms of attribute-value cluster $s_t$, is calculated as $U^{(t)}_{u,s_t}V^{(t)}_{x,s_t}$. Moreover, probability $p(u, x)$ is derived as $\sum_{s_t\in\mathbb{S}_t} U^{(t)}_{u,s_t}V^{(t)}_{x,s_t}$. Therefore, when $U^{(t)}$, $V^{(t)}$ minimize loss function, $U^{(t)}$, $V^{(t)}$ represent the best approximation of the edges in the graph.

$$\underset{U^{(t)},V^{(t)}\geq 0}{\arg\min} \left\| X^{(t)} - U^{(t)}(V^{(t)})^T \right\|_F^2 . \tag{2}$$

**Loss function for relationship detection.** In CARNMF, the relationships between communities and attribute-value clusters of attribute $t \in \mathbb{T}$ are represented as a matrix $R^{(t)} \in \mathbb{R}^{\ell\times k_t}$, where each row and column corresponds to a community $c \in \mathbb{C}$ and an attribute-value cluster $s_t \in \mathbb{S}_t$, respectively. The cell contains the probability $p(s_t \mid c)$. We assume $R^{(t)}$ is a linear transformation that maps $U^*$ into $U^{(t)}$, where $U^*$ and $U^{(t)}$ derived by Eq. (1) and Eq. (2), respectively. Moreover, the joint probability $p(u, s_t) = U^{(t)}_{u,s_t}$ can also be calculated as $\sum_c p(u, c)p(s_t \mid c) = \sum_c U^*_{u,c}R^{(t)}_{c,s_t}$ Therefore, when $R^{(t)}$ minimizes the loss function, $R^{(t)}$ represents the relationships between the communities and the attribute-value clusters.

$$\underset{U^{(t)},U^*,R^{(t)}\geq 0}{\arg\min} \left\| U^{(t)} - U^*R^{(t)} \right\|_F^2 . \tag{3}$$

Equation (3) can be regarded as an NMF that decomposes the matrix of the node-by-attribute value cluster into node-by-community and community-by-attribute value cluster matrices. In other words, Eq. (3) indicates the effect of the relationship between nodes and attribute-value clusters against communities.

**Unified loss function.** To achieve CAR-clustering, the aforementioned three sub-problems must be solved. In this work, we attempt to solve them simultaneously by introducing a unified loss function, which is expressed as

$$\begin{aligned} L = &\underset{U^*,\, \{U^{(t)},\, V^{(t)},\, R^{(t)}\}_{t\in\mathbb{T}}}{\arg\min} \left\| A - U^*(U^*)^T \right\|_F^2 \\ &+ \sum_{t\in\mathbb{T}} \left\{ \left\| X^{(t)} - U^{(t)}(V^{(t)})^T \right\|_F^2 + \lambda_t \left\| U^{(t)} - U^*R^{(t)} \right\|_F^2 \right\}, \end{aligned} \tag{4}$$

where $\lambda_t$ for attribute $t \in \mathbb{T}$ is a user-defined parameter to control the effect of attribute-value clusters for community detection. Higher $\lambda_t$ yields a stronger effect of the attribute-value clusters in community detection.

### 4.3 Optimization

Similar to the ordinary NMF, the loss function in Eq. (4) is not simultaneously convex for all variables. Hence, we consider the NMF to be a Frobenius norm optimization, where update equations are derived based on [27]. From the Karush-Kuhn-Tucker (KKT) conditions, we derive update rules corresponding to the variables as follows:

$$U^* \leftarrow U^* \odot \frac{A^T U^* + \sum_{t\in\mathbb{T}} \lambda_t U^{(t)}(R^{(t)})^T}{2U^*(U^*)^T U^* + \sum_{t\in\mathbb{T}} \lambda_t U^* R^{(t)}(R^{(t)})^T}, \tag{5}$$

$$U^{(t)} \leftarrow U^{(t)} \odot \frac{X^{(t)}V^{(t)} + \lambda_t U^* R^{(t)}}{U^{(t)}(V^{(t)})^T V^{(t)} + \lambda_t U^{(t)}}, \tag{6}$$

$$V^{(t)} \leftarrow V^{(t)} \odot \frac{(X^{(t)})^T U^{(t)}}{(V^{(t)})^T (U^{(t)})^T U^{(t)}}, \tag{7}$$

$$R^{(t)} \leftarrow R^{(t)} \odot \frac{(U^*)^T U^{(t)}}{(U^*)^T U^* R^{(t)}}. \tag{8}$$

The detailed explanations for the derivation of update rules are described in appendix A, B, C, and D.

The aforementioned update rules monotonically decrease the unified loss function (Eq. (4)). It should be noticed that the updated variables may have quite large values, leading to inconsistent results. To avoid such situations, we normalize them immediately after each update according to the following formulas:

$$U^* \leftarrow U^*(Q^*)^{-1}, \tag{9}$$

$$V^{(t)} \leftarrow V^{(t)}(Q^{(t)})^{-1}, \tag{10}$$

$$U^{(t)} \leftarrow U^{(t)}Q^{(t)}, \tag{11}$$

$$R^{(t)} \leftarrow R^{(t)}(Q^R)^{-1}. \tag{12}$$

where $Q^* = Diagonalize(U^*)$, $Q^{(t)} = Diagonalize(V^{(t)})$, and $Q^{R^{(t)}} = Diagonalize(R^{(t)})$.

$$Diagonalize(Z \in \mathbb{R}^{a\times b}) = Diag\left( \sum_{i=1}^{a} Z_{i,1} \cdots , \sum_{i=1}^{a} Z_{i,b} \right). \tag{13}$$

$Diag(\cdot)$ provides a diagonal matrix where the diagonals are the input sequence. Algorithm 1 shows the optimization algorithm based on the aforementioned update rules.

**Algorithm 1** Optimization Algorithm

```
Input: $A, \{X^{(t)}\}_{t\in\mathbb{T}}, \{\lambda_t\}_{t\in\mathbb{T}}, \delta$
Output: $U^*, \{U^{(t)}, V^{(t)}, R^{(t)}\}_{t\in\mathbb{T}}$
 1: $U^*, \{U^{(t)}, V^{(t)}, R^{(t)}\}_{t\in\mathbb{T}} \leftarrow$ random non-negative init
 2: $\epsilon' \leftarrow maxFloat, \epsilon \leftarrow \frac{\epsilon'}{2}$
 3: while $abs(\epsilon' - \epsilon) \geq \delta$ do
 4:     $U^* \leftarrow U^* \odot \frac{A^T U^* + \sum_{t\in\mathbb{T}} \lambda_t U^{(t)} (R^{(t)})^T}{2U^*(U^*)^T U^* + \sum_{t\in\mathbb{T}} U^* R^{(t)} (R^{(t)})^T}$
 5:     $U^* \leftarrow U^* (Q^*)^{-1}$
 6:     for $t \in \mathbb{T}$ do
 7:         $U^{(t)} \leftarrow U^{(t)} \odot \frac{X^{(t)} V^{(t)} + \lambda_t U^* R^{(t)}}{U^{(t)} (V^{(t)})^T V^{(t)} + \lambda_t U^{(t)}}$
 8:         $V^{(t)} \leftarrow V^{(t)} \odot \frac{(X^{(t)})^T U^{(t)}}{(V^{(t)})^T (U^{(t)})^T U^{(t)}}$
 9:         $U^{(t)} \leftarrow U^{(t)} Q^{(t)}$
10:         $V^{(t)} \leftarrow V^{(t)} \left(Q^{(t)}\right)^{-1}$
11:         $R^{(t)} \leftarrow R^{(t)} \odot \frac{(U^*)^T U^{(t)}}{(U^*)^T U^* R^{(t)}}$
12:         $R^{(t)} \leftarrow R^{(t)} \left(Q^R\right)^{-1}$
13:     end for
14:     $\epsilon' \leftarrow \epsilon$
15:     $\epsilon \leftarrow L\left(U^*, \{U^{(t)}, V^{(t)}, R^{(t)}\}_{t\in\mathbb{T}}\right)$
16: end while
```

### 4.4 Complexity Analysis

Here, we analyze the computational complexity of the proposed algorithm. The equations in our algorithm have the following complexities:

- Updating $U^*$ (Eqs. (5) and (9)) needs $O(|\mathbb{E}|\ell + |\mathbb{V}|\ell^2 \sum_t k_t)$.
- Updating $U^{(t)}$ (Eqs. (6) and (11)) and $V^{(t)}$ (Eqs. (7) and (10)) needs $O\left((|\mathbb{V}| + |\mathbb{X}_t|)k_t^2 + |\mathbb{E}_t|k_t\right)$.
- Updating $R^{(t)}$ (Eqs. (8) and (12)) needs $O\left(|\mathbb{V}|(\ell k_t + \ell^2)\right)$.

In summary, the time complexity of our algorithm is follows, where *iter* is the number of outer iterations (lines 3–16 in our algorithm).

$$O\left(iter \sum_t \left(|\mathbb{V}|(\ell^2 k_t + k_t^2) + |\mathbb{X}_t|k_t^2 + |\mathbb{E}|\ell + |\mathbb{E}_t|k_t\right)\right). \tag{14}$$

## 5. Experimental Evaluations

To demonstrate the applicability and effectiveness of CARNMF, we conducted a set of experiments using real-world datasets. Specifically, the performance of the proposed scheme was compared to simple baseline and the state-of-the-art methods.

The experiments were performed on a PC with an Intel Core i7 (3.3 GHz) CPU with 16 GB RAM running Ubuntu14.04. CARNMF was implemented by Python 2.7.6 with Numpy 1.9.0.

### 5.1 Datasets

We used two datasets: DBLP and arXiv.

- *DBLP*: Digital Bibliography Project[†] is a bibliographic database in the computer science area. DBLP contains publication information, such as authors and conferences. We used a part of the dataset by extracting conferences similar to [28]. We extracted four research areas: data mining, databases, machine learning, and information retrieval, and five major conferences for each area. Consequently, 10,491 papers in 20 conferences (shown in Table 1) were selected.
- *arXiv*: arXiv[††] is a repository of electronic preprints in various scientific fields. Similar to above, we chose four research areas: mathematical physics (math-ph), nuclear (nucl-th), astrophysics (astro-ph), and materials (part of cond-mat), and four major journals for each area. Consequently, 12,547 papers in 16 journals (shown in Table 2) were selected.

**Table 1** Selected conferences on four research areas.

| DB | DM | ML | IR |
|---|---|---|---|
| SIGMOD, VLDB<br>PODS, EDBT<br>ICDT | KDD, ICDM<br>PKDD, SDM<br>PAKDD | NIPS, ICML<br>ECML, UAI<br>COLT | SIGIR, ECIR<br>JCDL, ECDL<br>TREC |

**Table 2** Selected journals on four research areas.

| **math-ph** |
|---|
| Communications in Mathematical Physics<br>Reviews in Mathematical Physics<br>Letters in Mathematical Physics<br>Journal of Mathematical Physics |
| **nucl-th** |
| Annual Review of Nuclear and Particle Science<br>Progress in Particle and Nuclear Physics<br>Atomic Data and Nuclear Data Tables<br>Journal of Nuclear Materials |
| **astro-ph** |
| Research in Astronomy and Astrophysics<br>Annual Review of Astronomy and Astrophysics<br>New Astronomy Reviews<br>Space Science Review |
| **cond-mat** |
| Nature Nanotechnology<br>Nature Materials<br>Nano Letters<br>Journal of Materials Science |

Multi-attributed graphs were constructed from the datasets as follows: The nodes correspond to the authors. If two authors co-author a paper, we placed a weighted edge between the authors. The weighting denotes the number of co-authored papers. Each author has attributes *term*, *paper*, and *conference/journal*, which are defined below:

- *term*: Each term is regarded as a node. An edge is generated between an author and a term if the author uses the term in the titles of at least one paper. The edge weight denotes the term frequency for each author. As a preprocessing, we applied stop-word elimination and stemming.
- *paper*: Each paper is regarded as a node. An edge is generated if the author publishes the paper. The edge

[†] http://dblp.uni-trier.de/

[††] https://arxiv.org

weight is always 1.0 because each paper can only be published once.

- *conference/journal*: Each conference or journal corresponds with a node. An edge is created between an author and a conference/journal if the author publishes at least one paper at the conference/journal. The edge weight is the total number of publications at the conference/journal.

### 5.2 Results of CAR-Clustering

Figure 1 shows examples of the detected communities and their associated attribute-value clusters in DBLP detected by the proposed method. The number of communities and that of term clusters were each 50, whereas the number of conference clusters and that of paper clusters were each 4. The red, blue, and gray rectangles correspond to communities, term clusters, and conference clusters, respectively, showing the top nodes in community/cluster in terms of the contribution. The edge weights show the strengths of the relationships between the communities and the corresponding term and conference clusters.

The result includes the communities containing researchers in database and data mining areas, namely, "jiawei han", "hans-peter kriegel," and "jennifer widom." We can observe: (1) the communities of "jiawei han" and "hans-peter kriegel" are mainly related to data mining conferences (i.e., KDD, ICDM, SDM, PAKDD, and PKDD), while that of "jennifer widom" is related to database conferences (i.e., SIGMOD, VLDB, PODS, EDBT, and ICDT); (2) "jiawei han"'s community is strongly related to topics "frequent pattern mining" and "kind of matching", "hans-peter kriegel"'s community is related to "clustering" and "frequent pattern mining", and "jennifer widom"'s community is relates to "clustering" and "databases"; and (3) the communities of "jiawei han" and "hans-peter kriegel" are related in terms of data mining conferences and the topic "frequent pattern mining" while the communities "hans-peter kriegel" and "jennifer widom" are related in terms of topic "clustering". From the results, it seems that the proposed scheme successfully extract representative communities and their related topics along with their relationships.

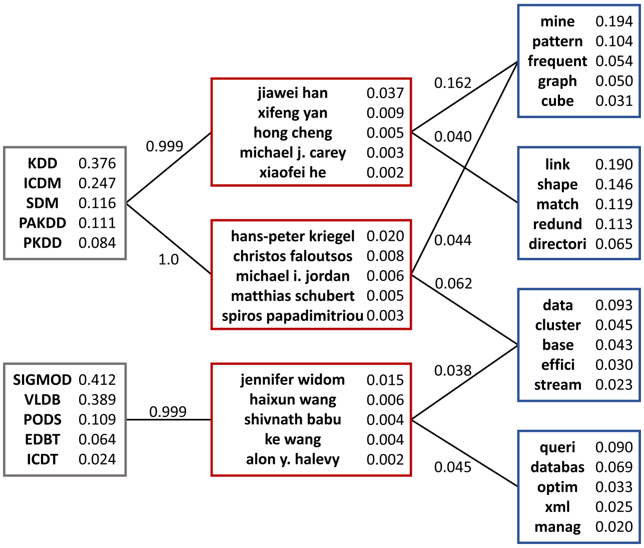


**Fig. 1** Example communities with attribute-value clusters. The red, blue and gray rectangles correspond to communities, term clusters, and conference clusters, respectively.

### 5.3 Accuracy Comparison

The proposed scheme is compared to a baseline method as well as the state-of-the-art methods to quantitatively evaluate the performance of community detection and attribute-value clustering. In this experiment, we find the hyper parameters which bring the highest accuracy for each method using grid search. The comparison methods include:

- **NMF** [29]: Baseline approaches that apply NMF for binary relationships between graph components, including author-term (A-T), author-paper (A-P), author-conference (A-C), term-paper (T-P), and term-conference (T-C)[†].
- **LCTA** [11]: A probabilistic generative model for communities, topics of textual attributes, and their relationships. We set hyper parameter $\lambda$ to 0.0 for all dataset.
- **SCI** [8]: An NMF based method for detecting communities as well as their semantic descriptions via node's attribute values. We set hyper parameters $\alpha$ and $\beta$ to 80 for DBLP dataset, and 80 and 0.05 for arXiv dataset, respectively.
- **HINMF** [26]: A model that clusters objects and attributes simultaneously and takes the consensus among the binary NMFs. This work is the most similar to our proposal. We set hyper parameter $\alpha$ to 0.01 for all dataset.

Note that, LCTA and SCI deal with a single concatenated feature of multiple attributes. Therefore, we prepare concatenated feature consisting of *term*, *document* and *conference/journal*, and apply these approaches on the feature. As for CARNMF, we set parameters $\lambda_t$ to all 0.01 for DBLP dataset, and all 0.05 for arXiv dataset.

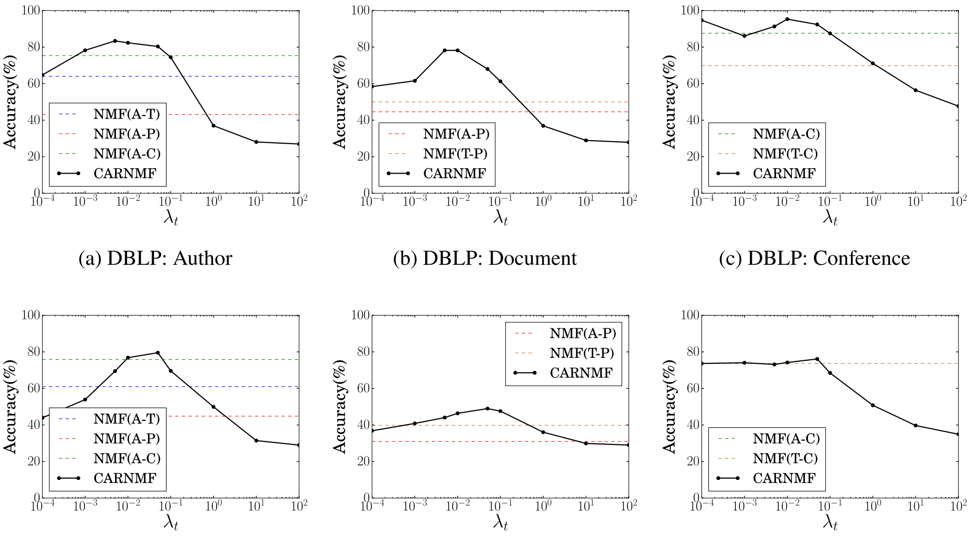
To evaluate the qualities of these methods, we compared the *accuracy* [11] of community and attribute-value clustering w.r.t. *paper* and *conference/journal*. We designed a ground truth to measure the *accuracy*. To derive the ground truth, each author is labeled based on research areas of their papers, in other words, if the author mostly published papers for the specific area, the author is labeled as that area. Similarly, the labels for *conference/journal* and *paper* were manually given by referring to the conference categories.

**Definition 5** (Accuracy): Given a set $\mathbb{S}$ of elements, for

[†] Because NMF assumes the co-occurrences of binary relationships, paper-conference (one-to-one relationship) is excluded.

**Table 3** Quality evaluations of community detection and attribute clustering.

| | DBLP dataset | | | arXiv dataset | | |
|---|---|---|---|---|---|---|
| | Author | Paper | Conference | Author | Paper | Journal |
| NMF(A-T) | 64.02 ± 5.73 | N/A | N/A | 60.99 ± 0.07 | N/A | N/A |
| NMF(A-P) | 43.12 ± 5.17 | 44.58 ± 5.89 | N/A | 44.84 ± 5.06 | 30.94 ± 1.15 | N/A |
| NMF(A-C) | 75.35 ± 6.85 | N/A | 87.60 ± 1.73 | 75.85 ± 7.29 | N/A | 73.68 ± 2.33 |
| NMF(T-P) | N/A | 50.02 ± 7.93 | N/A | N/A | 39.80 ± 5.05 | N/A |
| NMF(T-C) | N/A | N/A | 69.88 ± 6.68 | N/A | N/A | **100.00 ± 0.0** |
| LCTA | 48.90 ± 7.57 | 26.13 ± 4.36 | 68.50 ± 12.46 | 46.72 ± 5.72 | 31.50 ± 1.17 | 56.87 ± 6.53 |
| SCI | 54.78 ± 8.79 | 22.31 ± 1.48 | 58.20 ± 7.40 | 35.42 ± 4.01 | 29.79 ± 1.11 | 47.49 ± 6.37 |
| HINMF | 68.90 ± 9.08 | 56.46 ± 3.08 | 90.10 ± 12.63 | 74.30 ± 7.99 | 29.68 ± 0.95 | 73.12 ± 8.86 |
| **CARNMF** | **86.34 ± 2.39** | **78.19 ± 9.87** | **97.20 ± 5.21** | **77.64 ± 2.88** | **44.05 ± 3.14** | 75.00 ± 5.23 |



(a) DBLP: Author (b) DBLP: Document (c) DBLP: Conference

(d) arXiv: Author (e) arXiv: Document (f) arXiv: Journal

**Fig. 2** Accuracy for different $\lambda_t$ values.

each element $n \in \mathbb{S}$, the true label and the cluster label generated by a method are denoted by $s_n$ and $r_n$, respectively. Then, the *accuracy* is defined as:

$$Accuracy = \frac{\sum_{n \in \mathbb{S}} \delta(s_n, map(r_n))}{|\mathbb{S}|}$$

where $|\cdot|$ is the cardinality of a set; $\delta(x, y)$ is a delta function which returns 1 if $x = y$, otherwise 0; and $map(r_n)$ is a mapping function that maps $r_n$ to the equivalent label in the dataset. The best mapping can be found by Kuhn-Munkres algorithm [30]. □

Table 3 summarizes the evaluation results. The number of communities and the number of attribute-value clusters for each attribute are each four. Each cell shows the mean and the standard deviation of the accuracies for 20 trials. N/A denotes that the method does not support the category. Values in bold indicate a significant improvement using the Student-t test, where $p < 0.05$.

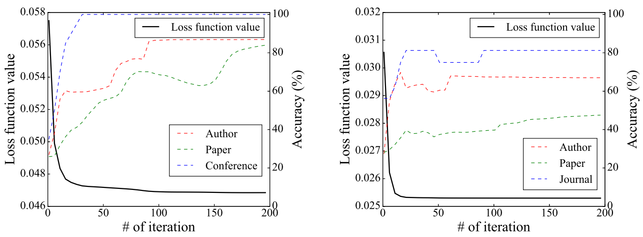
CARNMF achieved the best performance for community detection (author) and attribute-value clustering (paper and conference/journal) with significant gaps for DBLP dataset (respectively 11%, 22% and 7%) and for arXiv dataset (respectively 2%, 5%) relative to the comparative methods. In particular, CARNMF has an improved clustering quality compared to NMF by taking the relationships between communities and attribute-value clusters into account.

## 5.4 Insights on Parameters

This section discusses the effect of parameter $\lambda_t$ for each attribute. The larger the $\lambda_t$ value, the greater the influence of the attribute-value cluster for $t \in \mathbb{T}$ is on the community. Therefore, optimal parameter settings should result in better results. Figures 2 shows the behavior of the accuracy with different values with respect to different attributes. For each evaluation, $\lambda_s$ $(s \neq t)$ of the other attributes were fixed. In most cases, the accuracy shows a convex form and the peak is around $10^{-2}$. More importantly, the accuracy is insensitive to the setting, making tuning easier.

## 5.5 How to Determine Parameters

In this section, we discuss about how to determine the user defined parameters (i.e., $\ell$, $k_t$ and $\lambda_t$). As for the number of communities/clusters (i.e., $\ell$, $k_t$), the larger



(a) DBLP

(b) arXiv

**Fig. 3** Convergence analysis of the proposed algorithm to optimize a loss function and the corresponding accuracy curve.

community/cluster size brings the finer grained communities/clusters (e.g. laboratory, research topic) and the smaller community/cluster size brings the coarse grained communities/clusters (e.g. research society, research area). In the data analysis task, the required granularity of the communities/clusters varies depending on the purpose of the data analysis. Therefore, when applying CARNMF, the number of communities/clusters should be adjusted so as to obtain the target size by repeatedly applying our method. As for $\lambda_t$, as we discussed in the previos section, by setting the $\lambda_t$ to around 0.01, our method achieves highest accuracy. Thus, in the practical use of our method, it is better to set $\lambda_t$ to 0.01.

### 5.6 Convergence Analysis

In this section, we experimentally provide convergence analysis to optimize the proposed loss function in Eq. (4). Figures 3 (a) and (b) show the convergence curve of the loss function for DBLP and arXiv, respectively. In addition, the accuracy of each iteration is plotted. The black line shows the value of the loss function. The red, green, and blue lines show the accuracy of community detection and attribute-value clustering for author, paper, and conference/journal, respectively. As the number of iterations increases, the loss function decreases while the accuracy improves.

### 5.7 Efficiency Analysis

This section analyzes computational efficiency in terms of the numbers of communities and attribute clusters. When the numbers are fixed to four as experiments above, the running times of CARNMF on the DBLP (arXiv) dataset are 1.186 ± 0.253s (0.682 ± 0.138s). When changing the numbers of communities and term clusters to 50, while those of paper and conference remain four, the running times increases to 7.471 ± 0.563s (DBLP) and 6.526 ± 0.172s (arXiv). These values are still reasonable for various applications.

Moreover, we examine the running time of our method by changing the number of nodes in an input graph. Theoretically, as discussed in Sect. 4.4, the computational complexity is dependent on the number of vertices, that of edges, and that of distinct values of each attribute. As most of real-world graphs are modeled as scale-free networks, edges in a graph are very sparse, therefore, we examine the sensitivity of processing time on the proposed method in terms of the number of nodes. In this experiment, we selected all of the papers on DBLP, and construct the multi-attributed graph as same manner as described in Sect. 5.1. We set the number of communities and clusters are four. Figure 4 shows that the time complexity of our method is almost linear to the number of nodes. From the figure, we ensure that the time complexity of our method is linear to the numbers of nodes and edges (as shown on Eq. (14)). Therefore, when the input graph is sparse, our method is highly efficient.

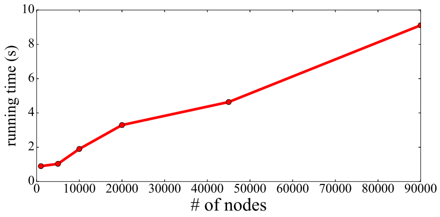


**Fig. 4** Time complexity of CARNMF w.r.t. the number of input nodes.

## 6. Conclusion

In this paper we have proposed CAR-clustering, which includes community detection, attribute-value clustering, and extraction of their relationships, for clustering over multi-attributed graphs. We have also proposed a novel algorithm CARNMF based on NMF. CARNMF employs a unified loss function to simultaneously solve different NMFs. This approach is better than the state-of-the-art methods in that it can exploit the correlation between communities and attribute-value clusters for enhancing the quality of the result. Our experiments have demonstrated that CARNMF successfully achieves CAR-clustering. CARNMF has detected reasonable communities with meaningful semantic descriptions via the relationship between communities and attribute-value clusters for real-world datasets. These results are useful for many applications such as node property estimations [3]–[5], community-wise information recommendations [6], and semantic reasoning for nodes/edges [7]. Additionally, CARNMF has achieved higher accuracy than comparative methods, including a baseline and the state-of-the-art methods. Our future work includes several directions. First, we will extend the proposed method for chronological analysis over temporal multi-attributed graphs. Second, we plan to automate the parameter tuning (e.g., the numbers of communities/clusters, $\lambda_t$, etc.).

### References

[1] P.V. Marsden, “Homogeneity in confiding relations,” Social Networks, vol.10, no.1, pp.57–76, 1988.

[2] D.B. Kandel, “Homophily, selection, and socialization in adolescent friendships,” American Journal of Sociology, vol.84, no.2, pp.427–436, 1978.

[3] M. Frank, A.P. Streich, D. Basin, and J.M. Buhmann, “Multi-assignment clustering for boolean data,” Journal of Machine Learning Research, vol.13, no.Feb, pp.459–489, 2012.

[4] M. Girvan and M.E.J. Newman, “Community structure in social and biological networks,” Proc. National Academy of Sciences of the United States of America, vol.99, no.12, pp.7821–7826, 2002.

[5] J. Yang and J. Leskovec, “Overlapping community detection at scale: A nonnegative matrix factorization approach,” Proc. Sixth ACM International Conference on Web Search and Data Mining, pp.587–596, ACM, 2013.

[6] J. Kamahara, T. Asakawa, S. Shimojo, and H. Miyahara, “A community-based recommendation system to reveal unexpected interests,” Proc. 11th International Multimedia Modelling Conference, MMM 2005, pp.433–438, IEEE, 2005.

[7] 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.Sep, pp.1981–2014, 2008.

[8] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, “Semantic community identification in large attribute networks,” AAAI, pp.265–271, 2016.

[9] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” Proc. 20th Conference on Uncertainty in Artificial Intelligence, pp.487–494, AUAI Press, 2004.

[10] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” 2013 IEEE 13th International Conference on Data Mining, pp.1151–1156, IEEE, 2013.

[11] Z. Yin, L. Cao, Q. Gu, and J. Han, “Latent community topic analysis: Integration of community discovery with topic modeling,” ACM Trans. Intelligent Systems and Technology (TIST), vol.3, no.4, Article No.63, 2012.

[12] H. Ito, T. Komamizu, T. Amagasa, and H. Kitagawa, “Community detection and correlated attribute cluster analysis on multi-attributed graphs,” Proc. Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienna, Austria, pp.2–9, 2018.

[13] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.8, pp.888–905, 2000.

[14] H. Zhang, B. Qiu, C.L. Giles, H.C. Foley, and J. Yen, “An LDA-based community structure discovery approach for large-scale social networks,” 2007 IEEE Intelligence and Security Informatics, pp.200–207, 2007.

[15] D. Kuang, C. Ding, and H. Park, “Symmetric nonnegative matrix factorization for graph clustering,” Proc. 2012 SIAM International Conference on Data Mining, pp.106–117, SIAM, 2012.

[16] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, “Overlapping community detection using Bayesian non-negative matrix factorization,” Physical Review E, vol.83, no.6, 066114, 2011.

[17] S. Fortunato, “Community detection in graphs,” Physics Reports, vol.486, no.3-5, pp.75–174, 2010.

[18] L. Tang and H. Liu, “Community detection and mining in social media,” Synthesis Lectures on Data Mining and Knowledge Discovery, vol.2, no.1, pp.1–137, 2010.

[19] M. Atzmueller, S. Doerfel, and F. Mitzlaff, “Description-oriented community detection using exhaustive subgroup discovery,” Information Sciences, vol.329, pp.965–984, 2016.

[20] S. Pool, F. Bonchi, and M. van Leeuwen, “Description-driven community detection,” ACM Trans. Intelligent Systems and Technology (TIST), vol.5, no.2, Article No.28, 2014.

[21] H. Cai, V.W. Zheng, F. Zhu, K.C.-C. Chang, and Z. Huang, “From community detection to community profiling,” Proc. VLDB Endowment, vol.10, no.7, pp.817–828, 2017.

[22] C. Bothorel, J.D. Cruz, M. Magnani, and B. Micenková, “Clustering attributed graphs: Models, measures and methods,” Network Science, vol.3, no.3, pp.408–444, 2015.

[23] M. Berlingerio, M. Coscia, and F. Giannotti, “Finding and characterizing communities in multidimensional networks,” 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp.490–494, IEEE, 2011.

[24] B. Boden, S. Günnemann, H. Hoffmann, and T. Seidl, “Mining coherent subgraphs in multi-layer graphs with edge labels,” Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1258–1266, ACM, 2012.

[25] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.797–806, ACM, 2009.

[26] J. Liu and J. Han, “HINMF: A matrix factorization method for clustering in heterogeneous information networks,” Proc. International Joint Conference on Artificial Intelligence Workshop, 2013.

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

[28] J. Gao, W. Fan, Y. Sun, and J. Han, “Heterogeneous source consensus learning via decision propagation and negotiation,” Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mmining, pp.339–348, ACM, 2009.

[29] D.D. Lee and H.S. Seung, “Algorithms for non-negative matrix factorization,” Advances in Neural Information Processing Systems, pp.556–562, 2001.

[30] H.W. Kuhn, “The Hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol.2, no.1-2, pp.83–97, 1955.

## Appendix A: Fixing $U^{(t)}, V^{(t)}, R^{(t)}$, Optimize $L$, over $U^*$

When update $U^*$ with $U^{(t)}$, $V^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned}\arg\min_{U^*\geq 0} L(U^*) &= \arg\min_{U^*\geq 0} \left\|A - U^*(U^*)^T\right\|_F^2 \\ &+ \sum_{t\in\mathbb{T}} \left\{\lambda_t \left\|U^{(t)} - U^*(R^{(t)})^T\right\|_F^2\right\}. \end{aligned} \qquad (A\cdot 1)$$

To this end, we introduce a Lagrange multiplier matrix $\Xi = (\Xi_{i,j})$ for the nonnegative constraint $U^* \geq 0$, and the Lagrange $\mathcal{L}(U^*) = L(U^*) + tr\left(\Xi^T U^*\right)$. We achieve the following equivalent objective function:

$$\begin{aligned}\mathcal{L}(U^*) &= tr\left(A^T A\right) - 2tr\left(A^T U^*(U^*)^T\right) \\ &+ tr\left((U^*)^T U^* U^* (U^*)^T\right) \\ &+ \sum_{t\in\mathbb{T}} \lambda_t \left(tr\left((U^{(t)})^T U^{(t)}\right) - 2tr\left((R^{(t)})^T (U^*)^T U^{(t)}\right)\right. \\ &+ \left. tr\left((R^{(t)})^T U^{*T} U^* R^{(t)}\right) + tr\left(\Xi^T U^*\right)\right). \end{aligned} \qquad (A\cdot 2)$$

Set derivative of $\mathcal{L}(U^*)$ with respect to $U^*$ to 0, we have:

$$\begin{aligned}\Xi = &-2A^T R U^* + 2U^*(U^*)^T U^* \\ &+ \sum_{t\in\mathbb{T}} \lambda_t (-U^{(t)}(R^{(t)})^T + U^* R^{(t)} (R^{(t)})^T). \end{aligned} \qquad (A\cdot 3)$$

Using Karush-Kuhn-Tucker (KKT) condition for the nonnegativity of $U^*$, we have the following equation:

$$\begin{aligned}U^*_{i,j}\Xi_{i,j} = U^*_{i,j} &\left(-2A^T R U^* + 2U^*(U^*)^T U^*\right. \\ &+ \left.\sum_{t\in\mathbb{T}} \lambda_t (-U^{(t)}(R^{(t)})^T + U^* R^{(t)} (R^{(t)})^T)\right)_{i,j} = 0. \end{aligned} \qquad (A\cdot 4)$$

This is the fixed point equation that the solution must satisfy at convergence. Given an initial value of $U^*$, the successive update of $U^*$ is:

$$U^* \leftarrow U^* \odot \frac{A^T U^* + \sum_{t\in\mathbb{T}} \lambda_t U^{(t)}(R^{(t)})^T}{2U^*(U^*)^T U^* + \sum_{t\in\mathbb{T}} \lambda_t U^* R^{(t)}(R^{(t)})^T}. \tag{A· 5}$$

## Appendix B: Fixing $U^*$, $V^{(t)}$, $R^{(t)}$, Optimize $L$, over $U^{(t)}$

When update $U^{(t)}$ with $U^*$, $V^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\begin{aligned}\mathop{\arg\min}_{U^{(t)}\geq 0} L(U^{(t)}) = \mathop{\arg\min}_{U^{(t)}\geq 0} &\left\|X^{(t)} - U^{(t)}(V^{(t)})^T\right\|_F^2 \\ &+ \lambda_t \left\|U^{(t)} - U^*(R^{(t)})^T\right\|_F^2.\end{aligned} \tag{A· 6}$$

To this end, we introduce a Lagrange multiplier matrix $\Psi = (\Psi_{i,j})$ for the nonnegative constraints $U^{(t)} \geq 0$ and the Lagrange $\mathcal{L}(U^{(t)}) = L(U^{(t)}) + tr\left(\Psi^T U^{(t)}\right)$. We achieve the following equivalent objective function:

$$\begin{aligned}\mathcal{L}(U^{(t)}) &= tr\left((X^{(t)})^T X^{(t)}\right) - 2tr\left((X^{(t)})^T V^{(t)}(U^{(t)})^T\right) \\ &+ tr\left(U^{(t)}(V^{(t)})^T U^{(t)}(V^{(t)})^T\right) \\ &+ \lambda_t tr\left((U^{(t)})^T U^{(t)}\right) - 2\lambda_t tr\left((R^{(t)})^T (U^*)^T U^{(t)}\right) \\ &+ \lambda_t tr\left((R^{(t)})^T U^{*T} U^* R^{(t)}\right) + tr\left(\Psi^T U^{(t)}\right).\end{aligned} \tag{A· 7}$$

Set derivative of $\mathcal{L}(R^{(t)})$ with respect to $R^{(t)}$ to 0, we have:

$$\begin{aligned}\Psi = &-X^{(t)}V^{(t)} + U^{(t)}(V^{(t)})^T V^{(t)} \\ &+ \lambda_t(U^{(t)} - U^* R^{(t)}).\end{aligned} \tag{A· 8}$$

Following KKT condition for the nonnegativity of $U^{(t)}$, we have the following equation:

$$\begin{aligned}U_{i,j}^{(t)}\Psi_{i,j} = U_{i,j}^{(t)}&\left(-X^{(t)}V^{(t)} + U^{(t)}(V^{(t)})^T V^{(t)}\right. \\ &\left.+\lambda_t(U^{(t)} - U^* R^{(t)})\right)_{i,j} = 0.\end{aligned} \tag{A· 9}$$

This is the fixed point equation that the solution must satisfy at convergence. Given an initial value of $U^{(t)}$, the successive update of $U^{(t)}$ is:

$$U^{(t)} \leftarrow U^{(t)} \odot \frac{X^{(t)}V^{(t)} + \lambda_t U^* R^{(t)}}{U^{(t)}(V^{(t)})^T V^{(t)} + \lambda_t U^{(t)}}. \tag{A· 10}$$

## Appendix C: Fixing $U^*$, $U^{(t)}$, $R^{(t)}$, Optimize $L$, over $V^{(t)}$

When update $V^{(t)}$ with $U^*$, $U^{(t)}$ and $R^{(t)}$ fixed, we need to solve the following problem:

$$\mathop{\arg\min}_{V^{(t)}\geq 0} L(V^{(t)}) = \mathop{\arg\min}_{V^{(t)}\geq 0} \left\|X^{(t)} - U^{(t)}(V^{(t)})^T\right\|_F^2. \tag{A· 11}$$

To this end, we introduce a Lagrange multiplier matrix $\Theta = (\Theta_{i,j})$ for the nonnegative constraints $V^{(t)} \geq 0$, and the Lagrange $\mathcal{L}(V^{(t)}) = L(V^{(t)}) + tr\left(\Theta^T V^{(t)}\right)$. We achieve the following equivalent objective function:

$$\begin{aligned}\mathcal{L}(V^{(t)}) &= tr\left((X^{(t)})^T X^{(t)}\right) - 2tr\left((X^{(t)})^T V^{(t)}(U^{(t)})^T\right) \\ &+ tr\left(U^{(t)}(V^{(t)})^T U^{(t)}(V^{(t)})^T\right) + tr\left(\Theta^T V^{(t)}\right).\end{aligned} \tag{A· 12}$$

Set derivative of $\mathcal{L}(V^{(t)})$ with respect to $V^{(t)}$ to 0, we have:

$$\Theta = -2(X^{(t)})^T U^{(t)} + 2(V^{(t)})^T (U^{(t)})^T U^{(t)}. \tag{A· 13}$$

Following KKT condition for the nonnegativity of $V^{(t)}$, we have the following equation:

$$\begin{aligned}V_{i,j}^{(t)}\Theta_{i,j} &= V_{i,j}^{(t)}\left(-2(X^{(t)})^T U^{(t)} + 2(V^{(t)})^T (U^{(t)})^T U^{(t)}\right)_{i,j} \\ &= 0.\end{aligned} \tag{A· 14}$$

This is the fixed point equation that the solution must satisfy at convergence. Given an initial value of $V^{(t)}$, the successive update of $V^{(t)}$ is:

$$V^{(t)} \leftarrow V^{(t)} \odot \frac{(X^{(t)})^T U^{(t)}}{(V^{(t)})^T (U^{(t)})^T U^{(t)}}. \tag{A· 15}$$

## Appendix D: Fixing $U^*$, $U^{(t)}$, $V^{(t)}$, Optimize $L$, over $R^{(t)}$

When update $R^{(t)}$ with $U^*$, $U^{(t)}$ and $V^{(t)}$ fixed, we need to solve the following problem:

$$\mathop{\arg\min}_{R^{(t)}\geq 0} L(R^{(t)}) = \mathop{\arg\min}_{R^{(t)}\geq 0} \left\|U^{(t)} - U^*(R^{(t)})^T\right\|_F^2. \tag{A· 16}$$

To this end, we introduce a Lagrange multiplier matrix $\Phi = (\Phi_{i,j})$ for the nonnegative constraints $R^{(t)} \geq 0$ and the Lagrange $\mathcal{L}(R^{(t)}) = L(R^{(t)}) + tr\left(\Phi^T R^{(t)}\right)$. We achieve the following equivalent objective function:

$$\begin{aligned}\mathcal{L}(R^{(t)}) &= tr\left((U^{(t)})^T U^{(t)}\right) - 2tr\left((R^{(t)})^T (U^*)^T U^{(t)}\right) \\ &+ tr\left((R^{(t)})^T U^{*T} U^* R^{(t)}\right) + tr\left(\Phi^T R^{(t)}\right).\end{aligned} \tag{A· 17}$$

Set derivative of $\mathcal{L}(R^{(t)})$ with respect to $R^{(t)}$ to 0, we have:

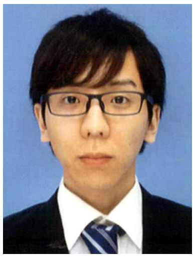
$$\Phi = -(U^*)^T U^{(t)} + (U^*)^T U^* R^{(t)}. \tag{A· 18}$$

Following the Karush-Kuhn-Tucker (KKT) condition for the nonnegativity of $R^{(t)}$, we have the following equation:

$$R_{i,j}^{(t)}\Phi = R_{i,j}^{(t)}\left(-(U^*)^T U^{(t)} + (U^*)^T U^* R^{(t)}\right)_{i,j} = 0. \tag{A· 19}$$

This is the fixed point equation that the solution must satisfy at convergence. Given an initial value of $V^{(t)}$, the successive update of $V^{(t)}$ is:

$$R^{(t)} \leftarrow R^{(t)} \odot \frac{(U^*)^T U^{(t)}}{(U^*)^T U^* R^{(t)}}. \tag{A· 20}$$



**Hiroyoshi Ito** received the B.Sc. and M.Eng. degrees from University of Tsukuba, Japan, in 2015 and 2017, respectively. He is currently a Ph.D. student at University of Tsukuba. His research interests include data mining and machine learning.

**Takahiro Komamizu** received the B.Eng. degree in computer science in 2009, the M.Eng. degree in 2011, and the Ph.D. degree in engineering from University of Tsukuba, Japan, in 2015. He is an assistant professor in the Information Technology Center, Nagoya University, Japan. His research interests include database, data analysis, and Linked Open Data. He is a member of IEICE, ACM, IEEE, DBSJ, IPSJ, NLP, and JSAI.

**Toshiyuki Amagasa** received B.E., M.E., and Ph.D. degrees from the Department of Computer Science, Gunma University in 1994, 1996, and 1999, respectively. Currently, he is a professor at the Center for Computational Sciences, University of Tsukuba. His research interests cover database engineering, data systems, semi-structured data management, and database application in scientific domains. He is a senior member of IEICE and IEEE, and a member of IPSJ, DBSJ, and ACM.

**Hiroyuki Kitagawa** received the B.Sc. degree in physics and the M.Sc. and Dr.Sc. degrees in computer science, all from the University of Tokyo. He is currently a full professor at Center for Computational Sciences and Center for Artificial Intelligence Research, University of Tsukuba. His research interests include databases, data integration, data mining, information retrieval, and data engineering applications. He served as President of the Database Society of Japan from 2014 to 2016. He is an IEICE Fellow, an IPSJ Fellow, an Associate Member of the Science Council of Japan, and a member of ACM, IEEE, JSST.