

MultiComm: Finding Community Structure in Multi-Dimensional Networks

Xutao Li¹, Michael K. Ng², Yunming Ye³

Abstract—The main aim of this paper is to develop a community discovery scheme in a multi-dimensional network for data mining applications. In online social media, networked data consists of multiple dimensions/entities such as users, tags, photos, comments and stories. For example, we are interested to find a group of users who interact significantly on these media entities. In a co-citation network, we are interested to find a group of authors who relate to other authors significantly on publication information in titles, abstracts, and keywords as multiple dimensions/entities in the network. The main contribution of this paper is to propose a framework (MultiComm) to identify a community in a multi-dimensional network by evaluating the affinity between two items in the same type of entity (same dimension) or different types of entities (different dimensions) from the network. Our idea is to calculate the probabilities of visiting each item in each dimension, and compare their values to generate communities. In order to evaluate a high quality of generated communities by the proposed algorithm, we develop and study a local modularity measure of a community in a multi-dimensional network. Experiments based on synthetic and real-world data sets suggest that the proposed framework is able to find a community effectively. Experimental results have also shown that the performance of the proposed algorithm is better in accuracy than the other testing algorithms in finding communities in multi-dimensional networks.

Index Terms—multi-dimensional networks, community, transition probability tensors; local modularity; affinity calculation

1 INTRODUCTION

Recently, there are growing interests in studying and analyzing large networks, such as social networks, genetic networks and co-citation networks [1], [2], [3], [4], [5]. In these networks, each node is an item corresponding to a dimension or an entity in a network. Each edge indicates a relationship between two nodes, for instance, a contact between two users in a social network, an interaction between two genes in a genetic network, and a citation between two papers or two authors in a co-citation network. Analyzing these networks enable us to understand their topological properties and structural organization.

One of such analysis is to detect communities or modules in large networks. One approach is to partition the network into sub-networks so that nodes in each sub-network are densely connected while nodes in different sub-networks are loosely connected. For example, the concepts of edge betweenness [1] and modularity [6], and the maximum-flow-minimum-cut theory [7] are used to divide the network. Spectral methods [8], [9], [10], [11] are also proposed to partition the entire network to detect communities.

In some applications, our primary concern may be to discover a particular community containing a set of nodes of interest rather than partition the whole structure of a network. For example, we are interested in a particular sharing group in a social network, or a

set of genes along a pathway in a genetic network. To find such community structure in a network, Bagrow and Bolt [12] developed an algorithm to search the structure of a network starting with a seed node. Clauset [13] proposed a local modularity measure and then developed a seed-based community discovery algorithm. In [14], [15], Luo et al. designed a new modularity measure and developed a KL-like algorithm to explore such community structure. However, these methods were developed for uni-dimensional networks.

In this paper, we are interested in developing such a community discovery scheme for a multi-dimensional network. There are many data mining and information retrieval applications in which there are several dimensions/entities in networks. In online social media, networked data consists of multiple dimensions/entities containing tags, photos, comments and stories [16]. We are interested to find a group of users who interact significantly on these media entities. In a co-citation network, we are interested to find a group of authors who cite/collaborate to each other (or a set of papers which are related to each other) significantly on publication information in titles, abstracts, and keywords as multiple dimensions/entities in the network [17].

Figure 1(a) shows an example of an academic publication network where some concepts are labeled to papers, and each paper is associated with several keywords and authors. In this network, there are four dimensions/entities. However, items in three dimensions (author, keyword and paper) are related and items in two dimensions (paper and concept) are

^{1,3}Department of Computer Science, Shenzhen Graduate School, Harbin Institute of Technology. Email:¹xutaoolee08@gmail.com; ³yeyunming@hit.edu.cn

²Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. Email: mng@math.hkbu.edu.hk

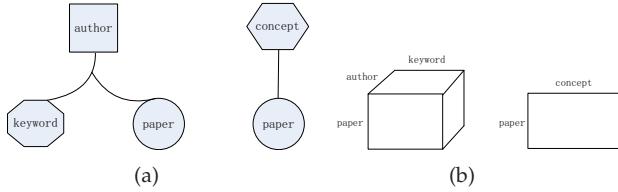


Fig. 1. (a) An example of an academic publication network. (b) The representation of the multi-dimensional network in (a): a tensor is used to represent the interactions among items in three dimensions/entities: author, keyword and paper, and a matrix is used to represent the interactions between items in two dimensions/entities: concept and paper.

related. We can make use of a tensor and a matrix to represent their interactions, see Figure 1(b), see the detailed representation in Section 3. Multiple interactions among entities/dimensions should be incorporated and studied in order to identify useful and important community structure in such a multi-dimensional network.

The main aim of this paper is to propose an algorithm, MultiComm, to identify community structure in a multi-dimensional network such that the involved items of the entities inside the community interact significantly, and meanwhile, they are not strongly influenced by the items outside the community. In our proposal, a community is constructed starting with a seed consisting of one or more items of the entities believed to be participated in a viable community. Given the seed item, we iteratively adjoin new items by evaluating the affinity between the items to build a community in the network. As there are multiple interactions among the items from different dimensions/entities in a multi-dimensional network, the main challenge is how to evaluate the affinity between the two items in the same type of entity (from the same dimension/entity) or in different types of entities (from different dimensions/entities). For example, in Figure 2, the affinity between a paper "A" and a paper "B" (the same type of entity), and the affinity between a paper "A" and a keyword "C" (different types of entities) are required in order to evaluate and decide the papers "A" and "B" or the paper "A" and the keyword "C" to put together in a community. On the other hand, we need a criterion in order to evaluate a high quality of generated communities by the proposed algorithm, we study a local modularity measure of a community in a multi-dimensional network. Experiments based on synthetic and real-world data suggest that the proposed framework is able to find a community effectively. Experimental results have also shown that the performance of the proposed algorithm is better in accuracy than the other testing algorithms in finding communities.

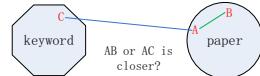


Fig. 2. The affinity between two items in the same/different type of entities.

The rest of the paper is organized as follows. In Section 2, we review the related work and give the motivations why the proposed algorithm is better than the other existing algorithms. In Section 3, we describe notations in this paper and present how to evaluate the affinity between items. In Section 4, we present the proposed framework and algorithm. In Section 5, we demonstrate the usefulness of the proposed algorithm by presenting the experimental results. In Section 6, we give some concluding remarks.

2 THE RELATED WORK

2.1 Community Discovery

In the literature, many methods have been proposed to extract community structures from uni-dimensional networks. In [13], Clauset proposed a local modularity measuring the sharpness of a subgraph boundary, and then developed a greedily-growing algorithm based on this modularity for exploring community structure. Later, Luo et al. designed a new local modularity based on the "indegree" and "outdegree" of a subgraph [14], and then proposed three algorithms for the identification of communities [15]. In 2006, Anderson and Lang investigated methods for growing communities by using random walk techniques [18]. Their basic idea is to generate communities by simulating a "truncated" random walk for a small number of steps starting from a distribution concentrated on the seed set. Recently, Mehler and Skiena [19] also proposed a general method for expanding network communities from input seed set. In this method, they studied several heuristic scoring criteria (neighbor count, juxtaposition count, neighbor ratio, etc.) to select the most promising next member for community expansion. During the expansion, part of seeds are kept aside as validation set to check when to stop. Based on the matrix blocking techniques, Chen and Saad proposed a method to extract dense subgraphs from sparse graphs for discovering dense communities [20]. Their basic idea is reordering the adjacency matrix to find dense diagonal blocks, each of which represents a dense community. However, all these methods were developed for uni-dimensional networks and thus may not be used to yield good performance for multi-dimensional networks.

As an example, we consider a network consisting of three entities/dimensions (A , B and C). Entity A includes five items $\{A_1, A_2, A_3, A_4, A_5\}$, entity B includes six items $\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and entity C

includes four items $\{C_1, C_2, C_3, C_4\}$. The interactions between them are represented as a tensor \mathcal{A} of size $5 \times 6 \times 4$. When items A_i , B_j and C_k interact, (i, j, k) position of \mathcal{A} is set to be 1, otherwise, it is a zero. \mathcal{A} is given as follows:

$$\mathcal{A}(:,:,1) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathcal{A}(:,:,2) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{A}(:,:,3) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{A}(:,:,4) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

In this example, we generate a community consisting of items $\{A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2\}$ in the network.

In order to employ uni-dimensional network community discovery methods, we must change the above tensor data into the adjacency matrix data as follows:

$$\left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

where the interaction between the items are constructed. For instance, when items A_i , B_j and C_k interact, the interactions between A_i and B_j , B_j and C_k , and A_i and C_k are shown in the matrix form. According to this example, we observe there are two disadvantages to handle matrix data. The first issue is that there is no direct interaction between the items of the same entity (all the diagonal blocks are zero). In order to group the items of same entity in the community, we must make use of the items in the other entities. The second issue is that some interactions are duplicated in the matrix form. For instance, the items A_1 , B_1 and C_1 interact and the items A_1 , B_1 and C_2 interact, thus A_1 and B_1 interact but we cannot differentiate the interaction, and capture the correlation between items in different dimensions/entities in the original tensor data. We report that the uni-dimensional based community discovery algorithms like Clauset cannot find the correct community in this example. The community discovered by Clauset is $\{A_1, A_2, A_3, B_1, B_2, B_3, C_2, C_4\}$. In this paper, we propose a scheme to extract communities from tensor data arising from multi-dimensional networks.

2.2 Tensor-based Community Detection

In multi-dimensional network analysis, networks have more than two types of entities. Most existing methods are based on tensor factorization [21], [16], [22]. Tensor factorization is a generalized approach for analyzing multi-way interactions among entities. In [16], Lin et al. proposed MetaFac (MetaGraph Factorization), a framework that extracts community structures from various social contexts and interactions. In this method, we need to select the number of decompositions (low-rank approximation) in the tensor factorization. The number of decompositions may not be known in advance. On the other hand, the computation of such tensor factorization may not be unique as there are several numerical methods (e.g., the alternating least squares procedure) used to compute such factorization and the factorization results depend on the initial guess. Different from these methods, we compute visiting probabilities of items in different dimensions in a multi-dimensional network to calculate the affinity between the items, and to find a community. The analysis of such probability calculation will be studied and analyzed in the next section. By using the example in Section 2.1, we find that MetaFac detects the community $\{A_1, A_2, A_3, A_5, B_1, B_2, B_3, B_4, C_1, C_2, C_4\}$ which is not the correct one.

2.3 Random Walk With Restart

The proposed affinity calculation in Section 3 is based on the idea of random walk with restart. There are many applications using random walk and related methods such as image annotation [23], connection subgraph identification [24], cluster discovery [25], and bi-relational network analysis [26]. The idea of random walk with restart is to consider a random particle that starts from node i , and the particle iteratively transmits to its neighborhood with the probability that is proportional to their edge weights. Also at each step, it has some probability α ($0 < \alpha < 1$) to return to the node i . The relevance score of node j with respect to node i is defined as the steady-state probability p_j that the particle will finally stay at node j [27]:

$$\mathbf{p} = (1 - \alpha)W\mathbf{p} + \alpha\mathbf{e}_i$$

where $\mathbf{p} = (p_j)$ is the steady-state probability referring to relevance scores of different nodes, W is the normalized weighted matrix associated with the graph, and \mathbf{e}_i is the starting vector with the i -th element 1 and 0 for others. Here the relevance score can capture the global structure of the graph and multi-facet relationship between two nodes. However, random walk and related methods only deal with simple networks so far. The main contribution of this paper to develop random walk and related methods based on tensors for finding communities in a multi-dimensional network. By using the proposed

algorithm in Section 3, we can find the ground truth community $\{A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2\}$ exactly in the network example in Section 2.1.

3 THE MATHEMATICAL FORMULATION

3.1 Preliminary

In this subsection, we describe notations and present some preliminary knowledge on tensors for representing multi-dimensional networks.

Let \mathbb{R} be the real field. We call $\mathcal{A} = (a_{i_1, \dots, i_m})$ where $a_{i_1, \dots, i_m} \in \mathbb{R}$, for $i_k = 1, \dots, n_k$, $k = 1, 2, \dots, m$, a real $(n_1 \times \dots \times n_m)$ -dimensional tensor \mathcal{A} . \mathcal{A} is called non-negative if $a_{i_1, \dots, i_m} \geq 0$. A positive (or nonnegative) vector means all its entries are positive (or non-negative). It is denoted by $\mathbf{x} > 0$ (or $\mathbf{x} \geq 0$).

Definition 1: A m -dimensional network with the size of the k -th dimension being n_k ($k = 1, 2, \dots, m$), is represented by a set of S tensors $\{\mathcal{A}^{(s)}\}_{s=1}^S$ where $\mathcal{A}^{(s)} = (a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s)})$ is a real $(n_{j_1^{(s)}} \times n_{j_2^{(s)}} \times \dots \times n_{j_{l_s}^{(s)}})$ -dimensional tensor to describe the interactions of items among l_s dimensions of indices $j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}$ in dimensions between 1 and m (they are not necessarily distinct): $a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s)} = 1$ if the $i_{j_1^{(s)}}$ -th item in the $j_1^{(s)}$ -th dimension, the $i_{j_2^{(s)}}$ -th item in the $j_2^{(s)}$ -th dimension, \dots , the $i_{j_{l_s}^{(s)}}$ -th item in the $j_{l_s}^{(s)}$ -th dimension are interacted, otherwise, $a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s)} = 0$. Each $\mathcal{A}^{(s)}$ is a non-negative tensor.

For example, there are four dimensions/entities (papers, authors, keywords and concepts), and there are n_1 papers, n_2 authors, n_3 keywords and n_4 concepts in the network shown in Figure 1. The network is represented by two tensors, i.e., $S = 2$. $\mathcal{A}^{(1)}$ is an $(n_1 \times n_2 \times n_3)$ -dimensional tensor representing the interactions among the papers, authors and keywords, i.e., $l_1 = 3$, $j_1^{(1)} = 1$, $j_2^{(1)} = 2$ and $j_3^{(1)} = 3$. $\mathcal{A}^{(2)}$ is an $(n_1 \times n_4)$ -dimensional tensor (matrix) representing the interactions between the papers and concepts, i.e., $l_2 = 2$, $j_1^{(2)} = 1$ and $j_2^{(2)} = 4$. In this example, when the $i_{j_1^{(1)}}$ -th paper is related to the $i_{j_2^{(1)}}$ -th author and it contains the $i_{j_3^{(1)}}$ -th keyword, we set $a_{i_{j_1^{(1)}, j_2^{(1)}, j_3^{(1)}}^{(1)}} = 1$, otherwise we set it to be 0, when the $i_{j_2^{(2)}}$ -th paper is related to the $i_{j_1^{(2)}}$ -th concept, we set $a_{i_{j_1^{(2)}, j_2^{(2)}}^{(2)}} = 1$, otherwise we set it to be 0.

As we consider the calculation of probabilities of the items of each dimension in a non-negative tensor arising from a multi-dimensional network, and study the likelihood that we will arrive at any particular item in a network, we construct an $(n_{j_1^{(s)}} \times n_{j_2^{(s)}} \times \dots \times n_{j_{l_s}^{(s)}})$ -dimensional transition probability tensor $\mathcal{P}^{(s,t)} = (p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s,t)})$ by normalizing the entries

of $\mathcal{A}^{(s)}$ with respect to the index $i_{j_t^{(s)}}$ in between 1 and $n_{j_t^{(s)}}$ ($t = 1, 2, \dots, l_s$) as follows:

$$p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s,t)}}^{(s,t)} = \frac{a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s)}}}{\sum_{i_{j_t^{(s)}}=1}^{n_{j_t^{(s)}}} a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}^{(s)}}}$$

These numbers give the estimates of the following conditional probabilities:

$$p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s,t)} = \text{Prob} \left[X_k^{(j_t^{(s)})} = i_{j_t^{(s)}} \mid \underbrace{X_{k-1}^{(j_u^{(s)})}}_{u=1, \dots, l_s, u \neq t} = i_{j_u^{(s)}} \right] \quad (1)$$

where $X_k^{(j_u^{(s)})}$ is a random variable referring to visit at any particular item of the $j_u^{(s)}$ -th dimension/entity at the time k . Here $p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s,t)}$ can be interpreted as the probability of visiting the $i_{j_t^{(s)}}$ -th item of the $j_t^{(s)}$ -th dimension in a network by given that the $i_{j_u^{(s)}}$ -th item of the $j_u^{(s)}$ -th dimension is currently visited where for $u = 1, \dots, l_s$ except $u \neq t$.

We note that if $a_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s)}$ is equal to 0 for all $1 \leq i_{j_t^{(s)}} \leq n_{j_t^{(s)}}$, this is called the dangling node [29], and the values of $p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s,t)}$ can be set to $1/n_{j_t^{(s)}}$ (an equal chance to visit any item in the $j_t^{(s)}$ -th dimension). With the above construction, we have

$$0 \leq p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s,t)} \leq 1, \text{ and } \sum_{i_{j_t^{(s)}}=1}^{n_{j_t^{(s)}}} p_{i_{j_1^{(s)}, j_2^{(s)}, \dots, j_{l_s}^{(s)}}}^{(s,t)} = 1. \quad (2)$$

We call $\mathcal{P}^{(s,t)}$ ($t = 1, 2, \dots, l_s$) transition probability tensors derived from $\mathcal{A}^{(s)}$. $\mathcal{P}^{(s,t)}$ can be viewed as a high-dimensional analog of transition probability matrices in Markov chains [28]. It is necessary to know the connectivity among the items of the entities within a tensor. We remark that the concept of irreducibility has been used in the PageRank matrix in order to compute the PageRank vector [29].

Definition 2: A $(n_1 \times \dots \times n_m)$ -dimensional tensor T is called irreducible if for any j and j' (the other indices are fixed) the $n_j \times n_{j'}$ matrices ($t_{\dots, i_j, \dots, i_{j'}, \dots}$) are irreducible. If T is not irreducible, then we call T reducible.

When $\mathcal{A}^{(s)}$ is irreducible, any two items in the same dimension or in different dimensions in a network can be connected via the other items. As we would like to make use of probability distributions to define an affinity between two items of the entities, irreducibility is a reasonable assumption that we will use in the following analysis and discussion. It is clear that when $\mathcal{A}^{(s)}$ is irreducible, the corresponding tensors

$\mathcal{P}^{(s,t)}$ ($s = 1, 2, \dots, S$ and $t = 1, 2, \dots, l_s$) are also irreducible.

According to (1), we calculate the probability of visiting an item in the $j_t^{(s)}$ -th dimension by multiplying the probabilities of visiting the other items in the other dimensions with $p_{i_1^{(s)}, i_2^{(s)}, \dots, i_{l_s^{(s)}}}^{(s,t)}$. Here we deal with a multiplication operation of a tensor with several vectors. Let $\mathbf{x}_{j_u^{(s)}}$ be vectors of length $n_{j_u^{(s)}}$ and $[\mathbf{x}_{j_u^{(s)}}]_{i_{j_u^{(s)}}}$ be the value of the $i_{j_u^{(s)}}$ -th item in the $j_u^{(s)}$ -th dimension ($u = 1, 2, \dots, l_s$). Let $\mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} \mathbf{x}_{j_u^{(s)}}$ be a vector in $\mathbb{R}^{n_{j_t^{(s)}}}$ such that

$$\begin{aligned} & \left[\mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} \mathbf{x}_{j_u^{(s)}} \right]_{i_{j_t^{(s)}}} \\ &= \underbrace{\sum_{i_{j_1^{(s)}}=1}^{n_{j_1^{(s)}}} \sum_{i_{j_2^{(s)}}=1}^{n_{j_2^{(s)}}} \cdots \sum_{i_{j_{l_s^{(s)}}}=1}^{n_{j_{l_s^{(s)}}}} p_{i_1^{(s)}, i_2^{(s)}, \dots, i_{l_s^{(s)}}}^{(s,t)}}_{\text{except the sum for } i_{j_t^{(s)}}} \\ & \quad [\mathbf{x}_{j_1^{(s)}}]_{i_{j_1^{(s)}}} [\mathbf{x}_{j_2^{(s)}}]_{i_{j_2^{(s)}}} \cdots [\mathbf{x}_{j_{l_s^{(s)}}}]_{i_{j_{l_s^{(s)}}}}, \\ & \quad \text{except the term } [\mathbf{x}_{j_t^{(s)}}]_{i_{j_t^{(s)}}}, \end{aligned}$$

for $i_{j_t^{(s)}} = 1, 2, \dots, n_{j_t^{(s)}}$. It is easy to check that when $\mathbf{x}_{j_u^{(s)}}$ are probability distribution vectors, i.e., $\sum_{i_{j_u^{(s)}}=1}^{n_{j_u^{(s)}}} [\mathbf{x}_{j_u^{(s)}}]_{i_{j_u^{(s)}}} = 1$, then the output vector $\mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} \mathbf{x}_{j_u^{(s)}}$ is also a probability distribution vector.

By using the publication network in Figure 1 as an example, we can calculate

- the probabilities of visiting the items in the “paper” dimension by given the probabilities that the items in the “author” and “keyword” dimensions are visited in $\mathcal{P}^{(1,1)}$ derived from $\mathcal{A}^{(1)}$:

$$\mathbf{x}_1 = \mathcal{P}^{(1,1)} \mathbf{x}_2 \mathbf{x}_3 \quad \text{or}$$

$$[\mathbf{x}_1]_{i_1} = \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} p_{i_1, i_2, i_3}^{(1,1)} [\mathbf{x}_2]_{i_2} [\mathbf{x}_3]_{i_3};$$

- the probabilities of visiting the items in the “author” dimension by given the probabilities that the items in the “paper” and “keyword” dimensions are visited in $\mathcal{P}^{(1,2)}$ derived from $\mathcal{A}^{(1)}$:

$$\mathbf{x}_2 = \mathcal{P}^{(1,2)} \mathbf{x}_1 \mathbf{x}_3 \quad \text{or}$$

$$[\mathbf{x}_2]_{i_2} = \sum_{i_1=1}^{n_1} \sum_{i_3=1}^{n_3} p_{i_1, i_2, i_3}^{(1,2)} [\mathbf{x}_1]_{i_1} [\mathbf{x}_3]_{i_3};$$

- the probabilities of visiting the items in the “keyword” dimension by given the probabilities that the items in the “author” and “paper” dimensions are visited in $\mathcal{P}^{(1,3)}$ derived from $\mathcal{A}^{(1)}$:

$$\mathbf{x}_3 = \mathcal{P}^{(1,3)} \mathbf{x}_1 \mathbf{x}_2 \quad \text{or}$$

$$[\mathbf{x}_3]_{i_3} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} p_{i_1, i_2, i_3}^{(1,3)} [\mathbf{x}_1]_{i_1} [\mathbf{x}_2]_{i_2}.$$

Similarly, we can calculate the probabilities of visiting the items in the “paper” dimension by given the probabilities that the items in the “concept” dimension are visited in $\mathcal{P}^{(2,1)}$ derived from $\mathcal{A}^{(2)}$:

$$\mathbf{x}_1 = \mathcal{P}^{(2,1)} \mathbf{x}_4 \quad \text{or} \quad [\mathbf{x}_1]_{i_1} = \sum_{i_4=1}^{n_4} p_{i_1, i_4}^{(2,1)} [\mathbf{x}_4]_{i_4}.$$

In the above example, there is no overlapping dimension in $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ which refer to paper-author-keyword interactions and paper-concept interactions respectively. In our model, we can handle the case where overlapping dimensions appear in a tensor, see the two examples in Experiment 3.

3.2 The Affinity Between Two Items

A community is constructed starting with a seed consisting of one or more items of the entities believed to be participated in a viable community. Given the seed item, we iteratively adjoin new items by evaluating the affinity between the items to build a community in the network. In this subsection, we present how to evaluate the affinity between the two items in the same type of entity (from the same dimension/entity) or in different types of entities (from different dimensions/entities).

Our idea is to calculate the affinity based on calculation of probabilities of visiting other items in a network from a given set of items. Motivated by the idea of topic-sensitive PageRank [30] and random walk with restart [27], we consider a random walker chooses randomly among the available interactions among the items in different dimensions, and makes a choice with probability α going back to a set of items in the current community. Based on this concept, we set the following tensor equations to calculate the required probabilities of visiting items in the v -th dimension in the whole network ($v = 1, 2, \dots, m$):

$$\mathbf{x}_v = (1-\alpha) \left[\frac{1}{|N_v|} \sum_{s=1, j_t^{(s)}=v}^S \mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} \mathbf{x}_{j_u^{(s)}} \right] + \alpha \mathbf{z}_v, \quad (3)$$

where \mathbf{z}_v is a probability vector that is constructed by setting the entries that correspond to the seed items or the current items in the v -th dimension in the community to be ones, and then normalizing it; the parameter $0 < \alpha < 1$ is to control the probability of a random jump to the items in the current community; and N_v is the number of tensors involving the v -th dimension in the network. As there is no prior knowledge in using a particular tensor in the network, we assume that we have equal chance to consider these N_v tensors involving the v -th dimension in the random walk, and the factor $1/N_v$ is used in the calculation of probabilities.

By using the publication network in Figure 1 as an example, we calculate the following probabilities of visiting the items in the “paper”, “author”, “key-word” and “concept” dimensions with given initial probability vectors in these dimensions $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ and \mathbf{z}_4 by solving the following set of tensor equations with $N_1 = 2$ and $N_2 = N_3 = N_4 = 1$:

$$\begin{aligned}\mathbf{x}_1 &= (1 - \alpha) \left[\frac{1}{2} \mathcal{P}^{(1,1)} \mathbf{x}_2 \mathbf{x}_3 + \frac{1}{2} \mathcal{P}^{(2,1)} \mathbf{x}_4 \right] + \alpha \mathbf{z}_1, \\ \mathbf{x}_2 &= (1 - \alpha) \mathcal{P}^{(1,2)} \mathbf{x}_1 \mathbf{x}_3 + \alpha \mathbf{z}_2, \\ \mathbf{x}_3 &= (1 - \alpha) \mathcal{P}^{(1,3)} \mathbf{x}_1 \mathbf{x}_2 + \alpha \mathbf{z}_3, \\ \mathbf{x}_4 &= (1 - \alpha) \mathcal{P}^{(2,2)} \mathbf{x}_1 + \alpha \mathbf{z}_4.\end{aligned}$$

We note that the only “paper” dimension involves in the two tensors $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$.

In the theorem below, we state that (3) is solvable, and the proof is given in the appendix.

Theorem 1: Suppose $0 < \alpha < 1$ and $\mathcal{A}^{(s)}$ ($1 \leq s \leq S$) are irreducible. Then (3) is solvable and its solution \mathbf{x}_v is non-negative and the summation of all the entries of \mathbf{x}_v is equal to 1 (i.e., a probability distribution vector) for $v = 1, 2, \dots, m$. Moreover, \mathbf{x}_v for $v = 1, 2, \dots, m$ are even positive.

Under the certain assumption on $\{\mathcal{P}^{(s,t)}\}_{s=1}^S$, the solution \mathbf{x}_v for $v = 1, 2, \dots, m$ are unique, see the appendix.

Next we present an efficient iterative algorithm to solve the tensor equations in (3) to obtain \mathbf{x}_v ($v = 1, 2, \dots, m$) for the probabilities of the items in the multi-dimensional network.

Input: Transition probability tensors $\{\mathcal{P}^{(s,t)}\}$ ($s = 1, 2, \dots, S, t = 1, 2, \dots, l_s$) and initial probability distributions $\{(\mathbf{x}_v)_0\}_{v=1}^m$ (here $(\cdot)_k$ denotes the iterate at the k -th iteration of the algorithm), prior probability distributions $\{\mathbf{z}_v\}_{v=1}^m$, prior probability parameter α , and the tolerance ϵ

Output: $\{\mathbf{x}_v\}_{v=1}^m$

Procedure:

- 1: Set $k = 1$
- 2: For $v = 1$ to m
- 3: Compute

$$(\mathbf{x}_v)_k = (1 - \alpha) \left[\frac{1}{|N_v|} \sum_{s=1, j_t^{(s)}=v}^S \mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} (\mathbf{x}_{j_u^{(s)}})_{k-1} \right] + \alpha \mathbf{z}_v$$
- 4: End For
- 5: If $\sum_{v=1}^m \|(\mathbf{x}_v)_k - (\mathbf{x}_v)_{k-1}\| < \epsilon$, then stop, set $\mathbf{x}_v = (\mathbf{x}_v)_k$; otherwise set $k = k + 1$ and goto Step 2.

In the algorithm, the computations require several iterations, through the collection to adjust approximate probability values of items of the entities in the multi-dimensional network to more closely reflect their theoretical true values (underlying probability distributions $\{\mathbf{x}_v\}_{v=1}^m$). The iterative method is similar to the power method for computing the eigenvector

corresponding to the largest eigenvalue value of a matrix [29]. The main computational cost of the above algorithm depends on the cost of performing tensor operations in Step 3. Assume that there are $O(N)$ nonzero entries (sparse data) in $\mathcal{P}^{(s,t)}$, the cost of these tensor calculations are of $O(N)$ arithmetic operations.

4 THE PROPOSED ALGORITHM

After solving a set of tensor equations in (3), we obtain the probability distributions of visiting each item in each dimension in the multi-dimensional network. These probability distributions $\{\mathbf{x}_v\}_{v=1}^m$ can be viewed as an affinity vector because it indicates the affinity of the items in each dimension to the items in the current community. Based on their probability values, we can determine the candidate items in different dimensions that are closely related to the current items in the community. In the next subsection, we will define the goodness criterion in order to determine the “best” community.

4.1 Local Modularity in a Multi-dimensional Network

In [13], Clauset defined a measure of community structure for a graph. The idea is that a good community with a sharp boundary will have few connections from its boundary to the other portion of the network, while having a greater proportion of connections from the boundary back into the community. Here we extend this idea to define a local modularity of a community in a multi-dimensional network.

Let B denote its boundary set composed of items in an m -dimensional network, where the item has connections to the items outside of the community explored. Let $\mathcal{B}^{(s)} = (b_{i_1, i_2, \dots, i_{l_s}}^{(s)})$ indicate its boundary-adjacency tensor for $\mathcal{A}^{(s)}$ ($s = 1, 2, \dots, S$), which can be computed as

$$b_{i_1, i_2, \dots, i_{l_s}}^{(s)} = \begin{cases} 1, & \text{if } a_{i_1, i_2, \dots, i_{l_s}}^{(s)} = 1 \text{ and the } i_t\text{-th} \\ & \text{item of the } t\text{-th dimension is in } B, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, we define the local modularity r_s for the tensor $\mathcal{A}^{(s)}$ to be

$$r_s = \frac{\sum_{i_1, i_2, \dots, i_{l_s}} b_{i_1, i_2, \dots, i_{l_s}}^{(s)} \delta(i_1, i_2, \dots, i_{l_s})}{\sum_{i_1, i_2, \dots, i_{l_s}} b_{i_1, i_2, \dots, i_{l_s}}^{(s)}} \quad (4)$$

where $\delta(i_1, i_2, \dots, i_{l_s})$ is one when some i_v -th items of the v -th dimension are in B and the other $i_{v'}$ -th items of the v' -th dimension are in the community, otherwise it is zero. Here the denominator in (4) is the number of connections with one or more items in B , while the numerator in (4) is the number of connections with neither items outside the community. By considering the fraction of boundary connections

which are internal to the community, we ensure that the measure of local modularity lies on the interval $0 \leq r_s < 1$, where its value is directly proportional to sharpness of the boundary given by B .

Algorithm 1 The MultiComm Algorithm

Input: Transition probability tensors $\mathcal{P}^{(s,t)}$ ($s = 1, 2, \dots, S$) and ($t = 1, 2, \dots, l_s$), the prior probability parameter $0 < \alpha < 1$, and the set L of seed items and its related dimension.

Output: a set of items for a community

Procedure:

- 1: Set $\{\mathbf{z}_v\}_{v=1}^m$ as follows: $[\mathbf{z}_v]_{i_v} = 1/c_v$ if $(i_v, v) \in L$, otherwise $[\mathbf{z}_v]_{i_v} = 0$, where c_v is the number of items in the v -th dimension from L ;
 - 2: Solve (3) to obtain probability distribution vectors $\{\mathbf{x}_v\}_{v=1}^m$;
 - 3: For each dimension, find the candidate item which is not in L and has the highest probabilities in \mathbf{x}_v ($v = 1, 2, \dots, m$), and generate the set of candidate items from all the dimensions to be L_c ;
 - 4: For each candidate item in L_c , determine the change of local modularity r_s of its corresponding tensor $\mathcal{A}^{(s)}$ (i.e., the corresponding dimension belongs to such tensor) after this candidate item is added;
 - 5: If the increases of local modularity for all the candidate items in L_c are not significant, then stop; otherwise, select the item i_v in the v th dimension which has the largest increase of local modularity r_s for $\mathcal{A}^{(s)}$, set $L = L \cup \{(i_v, v)\}$ and goto Step 1.
-

When communities vary in different subsets of dimensions, we can make use of r_s to identify which dimension of its corresponding item with the highest probability joins the community. In Algorithm 1, we summarize the MultiComm algorithm for finding a community in a multi-dimensional network.

We remark that after one community is determined, we can apply the MultiComm algorithm again to find another community. The proposed algorithm allows an item belonging to different communities.

5 EXPERIMENTAL RESULTS

In this section, we present experiments on both synthetic data sets and real data sets to demonstrate the performance of the proposed algorithm.

5.1 Data Sets and Evaluation Metrics

In the comparison, we study the results by using the MetaFac algorithm [16], the Clauset algorithm [13] and the LWP algorithm [15]. For the Clauset algorithm and the LWP algorithm, we put all the items as nodes in a single graph and two nodes are connected if their corresponding items are interacted in the original multi-dimensional network. This setting cannot

distinguish the interactions among different dimensions. MetaFac is to extract communities from a multi-dimensional network based on tensor factorization. The number of decompositions in the factorization is required to be specified.

We use four data sets to test the performance of MultiComm and the other three comparison algorithms. In the first two experiments, we test the performance based on two synthetic data sets. In the third experiment, we use two journal publication data sets to construct multi-dimensional networks to identify communities. In the last experiment, the MovieLens data set¹ are employed to discovery genre related communities.

In this paper, two metrics are employed to evaluate the accuracy for the algorithms, i.e., F-measure and NMI (*Normalized Mutual Information*)[31]. Given a community identified, we compute the F-measure as follows:

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where the precision and recall are calculated in terms of the ground-truth community. For NMI, we consider the community discovery problem as a binary class problem of whole items, and compute the normalized mutual information score between the identified partition π^a and the ground-truth partition π^b as follows:

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^2 \sum_{l=1}^2 n_{h,l} \log(\frac{n \cdot n_{h,l}}{n_h^{(a)} n_l^{(b)}})}{\sqrt{(\sum_{h=1}^2 n_h^{(a)} \log \frac{n_h^{(a)}}{n})(\sum_{l=1}^2 n_l^{(b)} \log \frac{n_l^{(b)}}{n})}}$$

where n is the total number of items, $n_h^{(a)}$, $n_l^{(b)}$ and $n_{h,l}$ represents the number of items in the h -th class in the partition π^a , the number of items in the l -th class in the partition π^b , and the number of items both in the h -th class and the l -th class respectively. Both F-measure and NMI are measured in between 0 and 1. When their values are higher, the accuracy of the algorithm is better.

5.2 Experiment 1

In this experiment, we generate a m -dimensional network represented by a tensor. We construct one “ground-truth” community and add noisy interactions in a tensor, and then check how different algorithms can recover this community. There are two parameters β and γ to control the data generation. The parameter β is used to control how strong the interactions among items in the community. The parameter γ is used to control how many noisy interactions in the network.

1. The data set is downloaded from <http://www.grouplens.org/node/462>

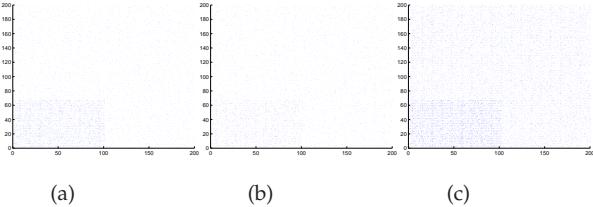


Fig. 3. The generated tensor with one community. (a) $m = 2$, $\beta = 5.0 \times 10^{-1}$, $\gamma = 1.0 \times 10^{-1}$; (b) $m = 2$, $\beta = 2.5 \times 10^{-1}$, $\gamma = 1.0 \times 10^{-1}$; (c) $m = 2$, $\beta = 6.0 \times 10^{-1}$, $\gamma = 4.0 \times 10^{-1}$.

For example, we show in Figure 3 three generated 2-dimensional networks, i.e., $m = 2$. The two axes represent the two different dimensions/entities. Each value on the axis refer to an item in the corresponding dimension/entity. Here the number of items in each dimension is 200, and thus \mathcal{A} is a (200×200) 2-dimensional tensor, i.e., $S = 1$. A point in the figure represents an interaction between two items. We see from Figure 3(a) that the region $[1, 100] \times [1, 66]$ in the generated network has more points concentrated together, and this region corresponds to a community in the network. The points sparsely distributed in the other parts refer to the noisy interactions among items of the two dimensions. When the value of β is large, the items in the community are strongly interacted, see Figure 3(a) with $\beta = 5.0 \times 10^{-1}$ and Figure 3(b) with $\beta = 2.5 \times 10^{-1}$. Also when the value of γ is large, there are more noisy interactions among items, see Figure 3(c) with $\gamma = 4.0 \times 10^{-1}$ and Figure 3(b) with $\gamma = 1.0 \times 10^{-1}$. We note that in these 2-dimensional networks, the following tensor equations are required to solve in order to compute the required probabilities:

$$\mathbf{x}_1 = (1-\alpha)\mathcal{P}^{(1,1)}\mathbf{x}_2 + \alpha\mathbf{z}_1 \text{ and } \mathbf{x}_2 = (1-\alpha)\mathcal{P}^{(1,2)}\mathbf{x}_1 + \alpha\mathbf{z}_2$$

where $\mathcal{P}^{(1,1)}$ and $\mathcal{P}^{(1,2)}$ are two transition probability matrices of sizes 200-by-200 from derived from \mathcal{A} .

We also construct two “ground-truth” communities and add noisy interactions in the generated networks. The two communities can be overlapped, i.e., an item can belong two communities, see Figure 4. In the figure, we assume that 25% of items of each dimension in a community is overlapped. For the other generated multi-dimensional networks, their corresponding tensor equations can be set up similarly in Tables 1 and 2.

In Figure 5, we show the changes of F-measure and NMI with respect to the values of α to the discovery of the community. We see that the values of F-measure and NMI increase when α increases up to around 0.4. For $\alpha > 0.4$ the values of F-measure and NMI are about the same. Therefore, we set $\alpha = 0.85$ to be the default value in MultiComm in the following experiments.

Tables 1 and 2 show the performance of four algorithms for several values of m , β and γ . The accuracy

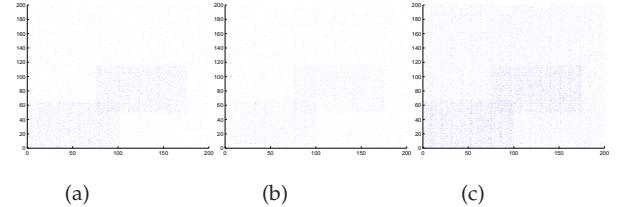


Fig. 4. The generated tensor with two overlapping communities. (a) $m = 2$, $\beta = 5.0 \times 10^{-1}$, $\gamma = 1.0 \times 10^{-1}$; (b) $m = 2$, $\beta = 2.5 \times 10^{-1}$, $\gamma = 1.0 \times 10^{-1}$; (c) $m = 2$, $\beta = 6.0 \times 10^{-1}$, $\gamma = 4.0 \times 10^{-1}$.

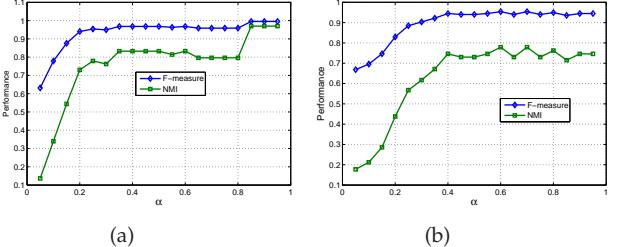


Fig. 5. The changes of F-measure and NMI with respect to the values of α . (a) Two community data generated with $m = 3$, $\beta = 5.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$; (b) Two community data generated with $m = 3$, $\beta = 3.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$.

results of Clauset, LWP and MultiComm are shown based on the averaged value of the ten runs of the corresponding algorithm with randomly selected seed items in the community. Since Metafac computes the tensor factorization, it is independent of seed items. In Metafac, we set the number of decompositions to be 2 for the tensors in Table 1 and select the community with the largest F-measure value from one of the decompositions. For the tensors in Table 2, we set the number of decompositions to be 3, and select the two communities with the largest and the second largest F-measure values from the tensor decomposition. According to Tables 1 and 2, we find that the performance of MultiComm is better than the other three algorithms. We note in Table 2 that 25% of items of each dimension in a community is overlapped in the generated multi-dimensional networks. We have also tested the performance of the four methods for different percentages of items to be overlapped in a community. The results show that MultiComm can give higher values of F-measure and NMI than those by the other three methods.

Moreover, Figure 6 shows how the local modularity changes against the number of items joined in the community on two generated 3-dimensional networks. We see from the figure that the increase of local modularity is not significantly changed when more than 200 items (the size of the ground-truth community) are joined in the community. This phenomena indicates the local modularity can be used as

TABLE 1
The performance of four different methods (one community).

m	β	γ	F-measure				NMI			
			MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
2	5.0×10^{-1}	1.0×10^{-1}	0.9171	0.4663	0.4187	0.9940	0.6838	0.4376	0.5449	0.9589
2	2.5×10^{-1}	1.0×10^{-1}	0.8114	0.7608	0.6517	0.9772	0.3761	0.3043	0.2406	0.8663
2	6.0×10^{-1}	4.0×10^{-1}	0.9477	0.7813	0.7616	0.9575	0.7473	0.3639	0.3387	0.7773
3	5.0×10^{-3}	5.0×10^{-4}	0.9307	0.4843	0.6813	0.9691	0.7576	0.3295	0.5723	0.8491
3	3.0×10^{-3}	5.0×10^{-4}	0.8055	0.7042	0.6143	0.9212	0.5191	0.2947	0.2240	0.6823
3	1.0×10^{-2}	2.5×10^{-3}	0.4404	0.8491	0.8406	0.8806	0.2470	0.5006	0.4815	0.5759
5	5.0×10^{-6}	2.0×10^{-8}	0.7780	0.3035	0.4214	0.9259	0.5376	0.2511	0.4001	0.8093
5	4.0×10^{-6}	2.0×10^{-8}	0.6773	0.6439	0.6097	0.9300	0.4362	0.5490	0.5720	0.7777
5	1.0×10^{-5}	8.0×10^{-8}	0.3974	0.8073	0.7969	0.9493	0.1665	0.4996	0.4712	0.8431
average F-measure/NMI			0.7451	0.6445	0.6440	0.9450	0.4968	0.3923	0.4273	0.7933

TABLE 2
The performance of four different methods (two communities).

m	β	γ	F-measure				NMI			
			MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
2	5.0×10^{-1}	1.0×10^{-1}	0.8504	0.8904	0.8817	0.9922	0.4453	0.7592	0.7429	0.9487
2	2.5×10^{-1}	1.0×10^{-1}	0.8174	0.7572	0.7321	0.9551	0.3693	0.4328	0.3589	0.7665
2	6.0×10^{-1}	4.0×10^{-1}	0.8402	0.7663	0.8228	0.8988	0.4226	0.4473	0.4701	0.7143
3	5.0×10^{-3}	5.0×10^{-4}	0.8957	0.9134	0.8938	0.9700	0.6711	0.6660	0.6265	0.8498
3	3.0×10^{-3}	5.0×10^{-4}	0.6895	0.7093	0.7441	0.9088	0.1859	0.3634	0.3578	0.6558
3	1.0×10^{-2}	2.5×10^{-3}	0.4571	0.8167	0.8337	0.9327	0.2563	0.5054	0.5418	0.7222
5	5.0×10^{-6}	2.0×10^{-8}	0.7352	0.8561	0.9081	0.9303	0.4911	0.6711	0.7576	0.7892
5	4.0×10^{-6}	2.0×10^{-8}	0.6303	0.6647	0.6843	0.9034	0.3968	0.4088	0.4618	0.7986
5	1.0×10^{-5}	8.0×10^{-8}	0.3990	0.8284	0.8187	0.9524	0.0431	0.5985	0.5751	0.8276
average F-measure/NMI			0.7016	0.8003	0.8133	0.9382	0.3646	0.5392	0.5436	0.7859

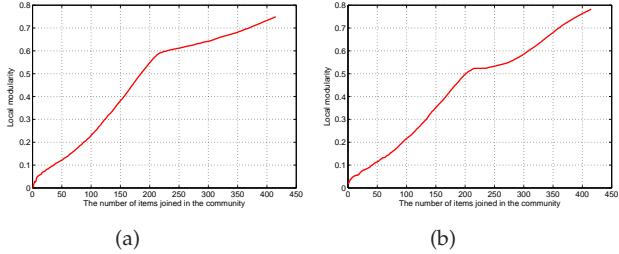


Fig. 6. The local modularity changes against the number of items joined in the community. (a) One community data generated with $m = 3$, $\beta = 5.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$; (b) Two community data generated with $m = 3$, $\beta = 5.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$.

a stopping criterion in the discovery of a community.

5.3 Experiment 2

In this experiment, we generate several tensors to represent a multi-dimensional network in order to test the performance of the proposed method. We consider a m -dimensional network represented by S tensors. Similar to Experiment 1, we generate a “ground-truth” community in the network, and add noisy interactions randomly in the network. For example, we show in Figure 7 a three-dimensional network generated by two 2-dimensional tensors $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ with $m = 3$, $S = 2$, $l_1 = 2$ ($j_1^{(1)} = 1$ and $j_2^{(1)} = 2$) and $l_2 = 2$ ($j_1^{(2)} = 1$ and $j_2^{(2)} = 3$). In this example, the community is composed of items (from 1 to 100) in the

first dimension, (from 1 to 66) in the second dimension and (from 101 to 166) in the third dimension. We see from the figure that the items (from 1 to 66) in the first dimension and the items (from 1 to 66) in the second dimension are strongly linked together in the first tensor, and the items (from 34 to 100) in the first dimension and the items (from 101 to 166) in the third dimension are strongly linked together in the second tensor. Here the following tensor equations are required to solve in MultiComm in order to compute the required probabilities:

$$\mathbf{x}_1 = (1 - \alpha) \left[\frac{1}{2} \mathcal{P}^{(1,1)} \mathbf{x}_2 + \frac{1}{2} \mathcal{P}^{(2,1)} \mathbf{x}_3 \right] + \alpha \mathbf{z}_1,$$

$$\mathbf{x}_2 = (1 - \alpha) \mathcal{P}^{(1,2)} \mathbf{x}_1 + \alpha \mathbf{z}_2, \quad \mathbf{x}_3 = (1 - \alpha) \mathcal{P}^{(2,2)} \mathbf{x}_1 + \alpha \mathbf{z}_3,$$

where $\mathcal{P}^{(1,1)}$ and $\mathcal{P}^{(1,2)}$ are two transition probability matrices of sizes 200-by-200 derived from $\mathcal{A}^{(1)}$, and $\mathcal{P}^{(2,1)}$ and $\mathcal{P}^{(2,2)}$ are two transition probability matrices of sizes 200-by-200 derived from $\mathcal{A}^{(2)}$. For the other generated multi-dimensional networks, their corresponding tensor equations can be set up similarly.

Table 3 ($S = 2$) and Table 4 ($S = 3$) show the performance of the four methods for several values of m , β and γ . For each m -dimensional network, we generate one community and set $l_1 = l_2 = \dots = l_S = l = \lceil \frac{2m-S+1}{S+1} \rceil$. For example, when $m = 5$ and $S = 2$, l is equal to 3, i.e., for a five-dimensional network, there are two 3-dimensional tensors to represent the network. For Metafac, we set the number of decompositions to be 2 and select the community

TABLE 3
The performance of four different methods on multi-dimensional networks generated with $S = 2$.

m	l	β	γ	F-measure				NMI			
				MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
3	2	5.0×10^{-1}	1.0×10^{-1}	0.9261	0.1453	0.1118	0.9957	0.7199	0.3300	0.4095	0.9710
	2	2.5×10^{-1}	1.0×10^{-1}	0.8696	0.6065	0.4138	0.9828	0.5631	0.1860	0.1667	0.8951
	2	6.0×10^{-1}	4.0×10^{-1}	0.9620	0.8211	0.7419	0.9957	0.8073	0.4203	0.3247	0.9701
5	3	5.0×10^{-3}	5.0×10^{-4}	0.9738	0.6370	0.4527	0.9742	0.8776	0.2876	0.0857	0.8784
	3	3.0×10^{-3}	5.0×10^{-4}	0.9048	0.5699	0.5604	0.9216	0.6475	0.0981	0.0900	0.6875
	3	1.0×10^{-2}	2.5×10^{-3}	0.4228	0.5681	0.5522	0.9272	0.2454	0.0963	0.0835	0.7094
8	5	5.0×10^{-6}	2.0×10^{-8}	0.8378	0.9678	0.9723	0.9805	0.6257	0.8921	0.9015	0.9211
	5	4.0×10^{-6}	2.0×10^{-8}	0.7458	0.9066	0.9578	0.9661	0.5145	0.7929	0.8762	0.8903
	5	1.0×10^{-5}	8.0×10^{-8}	0.3488	0.6223	0.6002	0.9899	0.2022	0.1954	0.1722	0.9490
average F-measure/NMI				0.7768	0.6494	0.5959	0.9704	0.5781	0.3665	0.3456	0.8747

TABLE 4
The performance of four different methods on multi-dimensional networks generated with $S = 3$.

m	l	β	γ	F-measure				NMI			
				MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
4	2	5.0×10^{-1}	1.0×10^{-1}	0.9141	0.1973	0.1025	0.9967	0.6937	0.3917	0.3890	0.9763
	2	2.5×10^{-1}	1.0×10^{-1}	0.8596	0.2862	0.3072	0.9920	0.5594	0.1593	0.1772	0.9425
	2	6.0×10^{-1}	4.0×10^{-1}	0.9643	0.7919	0.7286	0.9967	0.8326	0.3810	0.2705	0.9763
7	3	5.0×10^{-3}	5.0×10^{-4}	0.9307	0.4152	0.3917	0.9568	0.7959	0.0153	0.0088	0.8046
	3	3.0×10^{-3}	5.0×10^{-4}	0.9023	0.3937	0.3710	0.9419	0.6808	0.0093	0.0041	0.7552
	3	1.0×10^{-2}	2.5×10^{-3}	0.4890	0.3748	0.3565	0.9328	0.2876	0.0052	0.0023	0.7275
11	5	5.0×10^{-6}	2.0×10^{-8}	0.8772	0.9899	0.9861	0.9907	0.6903	0.9503	0.9499	0.9547
	5	4.0×10^{-6}	2.0×10^{-8}	0.7896	0.9856	0.9593	0.9906	0.5691	0.9496	0.8404	0.9537
	5	1.0×10^{-5}	8.0×10^{-8}	0.3589	0.3159	0.2656	0.9907	0.2358	0.0081	0.0010	0.9547
average F-measure/NMI				0.7873	0.5278	0.4965	0.9765	0.5939	0.3189	0.2937	0.8939

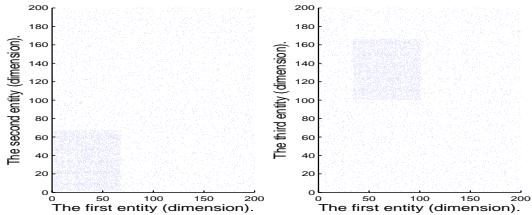


Fig. 7. The generated tensor with one community with $m = 3$, $\beta = 5 \times 10^{-1}$ and $\gamma = 1 \times 10^{-1}$.

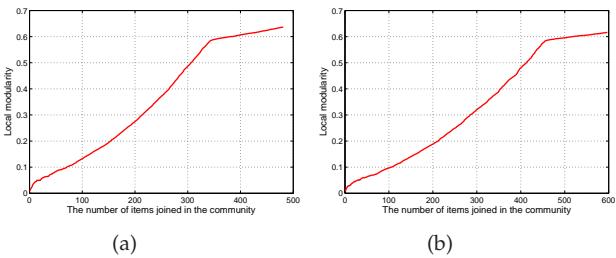


Fig. 8. The local modularity changes against the number of items joined in the community. (a) The multi-dimensional network generated with $S = 2$, $m = 5$, $l = 3$, $\beta = 5.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$; (b) The multi-dimensional network generated with $S = 3$, $m = 7$, $l = 3$, $\beta = 5.0 \times 10^{-3}$ and $\gamma = 5.0 \times 10^{-4}$.

with the largest F-measure value from one of the decompositions. The accuracy results of Clauset, LWP and MultiComm are shown based on the averaged value of the ten runs of the corresponding algorithm

with randomly selected seed items in the community. We see from the two tables that the performance of MultiComm is better than those of the other three methods for different settings.

In addition, Figure 8 shows how the local modularity changes with respect to the number of items joined in the community on two generated multi-dimensional networks. As each of these two multi-dimensional networks is represented by multiple tensors, here the local modularity refers to the average value of local modularities corresponding to these tensors. For instance, when $S = 3$ we have three tensors $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$, and the local modularity is $(r_1 + r_2 + r_3)/3$ where r_s is defined as in (4). We see from the figures that the increase of local modularity is not significantly changed when more than 350 items in Figure 8(a), and when more than 450 items in Figure 8(b). These item numbers are indeed the size of the ground-truth community in the generated 5-dimensional network and 7-dimensional network in Figure 8(a) and Figure 8(b) respectively. These results demonstrate the usefulness of the local modularity designed in Section 4.1.

5.4 Experiment 3

In this experiment, we use the SIAM journal data and DBLP conference data to construct multi-dimension networks and test the performance of MultiComm. For SIAM journal data, we consider the papers published in SJMAEL (SIAM Journal on Matrix Analysis and Applications) from volume 18 to volume 32,

in SJNAAM (SIAM Journal on Numerical Analysis) from volume 34 to volume 49, and in SJOCE3 (SIAM Journal on Scientific Computing) from volume 18 to volume 33. For DBLP data, we consider the papers published in KDD conference from 1999 to 2010 and in SIGIR from 2000 to 2010.

Journal/Conference-related Communities: We construct multi-dimensional networks as follows. The first step is to preprocess the data by keeping the authors that have at least two papers and their papers in the collection. We construct four tensors to represent multi-dimensional networks, namely, the paper-author-keyword tensor, the paper-paper citation tensor, the author-author collaboration tensor and the paper-category concept tensor. For the SIAM data set, the category concepts refer to the AMS codes in each paper. For the DBLP data set, the category concepts are provided in each paper. The description of these tensors in the multi-dimensional SIAM and DBLP networks are shown in Table 5. There are four dimensions (papers, authors, keywords, category concepts) in each network, i.e., $m = 4$. Each network is represented as four tensors, i.e., $S = 4$. $\mathcal{A}^{(1)}$ is a tensor representing the interactions among papers, authors and keywords with $l_1 = 3$, $j_1^{(1)} = 1$, $j_2^{(1)} = 2$ and $j_3^{(1)} = 3$. $\mathcal{A}^{(2)}$ is a tensor representing the citation interactions among papers with $l_2 = 2$, $j_1^{(2)} = 1$ and $j_2^{(2)} = 1$. (Here $l_2=2$ because the citation interactions are directed.) $\mathcal{A}^{(3)}$ is a tensor representing the collaboration interactions among authors with $l_3 = 1$ and $j_1^{(3)} = 2$. $\mathcal{A}^{(4)}$ is a tensor representing the interactions among papers and concepts with $l_4 = 2$, $j_1^{(4)} = 1$ and $j_2^{(4)} = 4$.

In MultiComm, the following tensor equations are built to calculate the required probabilities:

$$\begin{aligned}\mathbf{x}_1 &= (1-\alpha) \left[\frac{1}{4} \mathcal{P}^{(1,1)} \mathbf{x}_2 \mathbf{x}_3 + \frac{1}{4} \mathcal{P}^{(2,1)} \mathbf{x}_1 + \frac{1}{4} \mathcal{P}^{(2,2)} \mathbf{x}_1 \right. \\ &\quad \left. + \frac{1}{4} \mathcal{P}^{(4,1)} \mathbf{x}_4 \right] + \alpha \mathbf{z}_1, \\ \mathbf{x}_2 &= (1-\alpha) \left[\frac{1}{2} \mathcal{P}^{(1,2)} \mathbf{x}_1 \mathbf{x}_3 + \frac{1}{2} \mathcal{P}^{(3,1)} \mathbf{x}_2 \right] + \alpha \mathbf{z}_2,\end{aligned}$$

$$\mathbf{x}_3 = (1-\alpha) \mathcal{P}^{(1,3)} \mathbf{x}_1 \mathbf{x}_2 + \alpha \mathbf{z}_3, \quad \mathbf{x}_4 = (1-\alpha) \mathcal{P}^{(4,2)} \mathbf{x}_1 + \alpha \mathbf{z}_4,$$

where $\mathcal{P}^{(1,1)}$, $\mathcal{P}^{(1,2)}$ and $\mathcal{P}^{(1,3)}$ are three transition probability tensors derived from tensor $\mathcal{A}^{(1)}$, and $\mathcal{P}^{(2,1)}$ and $\mathcal{P}^{(2,2)}$ are two transition probability tensors derived from $\mathcal{A}^{(2)}$, and $\mathcal{P}^{(3,1)}$ is a transition probability tensor derived from $\mathcal{A}^{(3)}$, and $\mathcal{P}^{(4,1)}$ and $\mathcal{P}^{(4,2)}$ are two transition probability tensors derived from $\mathcal{A}^{(4)}$.

We test the usefulness of MultiComm by evaluating the three journal communities (SJMAEL, SJNAAM and SJOCE3) discovered in the SIAM network and the two conference communities (KDD and SIGIR) in the DBLP network. We report the values of F-measure and NMI by computing the discovered communities and the ground-truth journal or conference

labels. The four methods are performed on the SIAM network and the DBLP network. For MetaFac, we set the number of decompositions to be 3 for the SIAM network and 2 for the DBLP network. Table 6 shows the results of four methods. The accuracy results of Clauset, LWP and MultiComm are shown based on the averaged value of the ten runs of the corresponding algorithm with randomly selected seed items in the corresponding community. We see from the table that the performance of MultiComm is better than those of the other three methods on both networks.

Category Concept-related Communities: In this setting, we construct the multi-dimensional networks by using the tensors $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ because we want to discover category concept-related communities. The corresponding tensor equations are required to solve in MultiComm:

$$\begin{aligned}\mathbf{x}_1 &= (1-\alpha) \left[\frac{1}{3} \mathcal{P}^{(1,1)} \mathbf{x}_2 \mathbf{x}_3 + \frac{1}{3} \mathcal{P}^{(2,1)} \mathbf{x}_1 + \frac{1}{3} \mathcal{P}^{(2,2)} \mathbf{x}_1 \right] + \alpha \mathbf{z}_1, \\ \mathbf{x}_2 &= (1-\alpha) \left[\frac{1}{2} \mathcal{P}^{(1,2)} \mathbf{x}_1 \mathbf{x}_3 + \frac{1}{2} \mathcal{P}^{(3,1)} \mathbf{x}_2 \right] + \alpha \mathbf{z}_2, \\ \mathbf{x}_3 &= (1-\alpha) \mathcal{P}^{(1,3)} \mathbf{x}_1 \mathbf{x}_2 + \alpha \mathbf{z}_3.\end{aligned}$$

In MetaFac, we set the number of decompositions to be 10, and select the decomposition that corresponds to the largest F-measure for each testing category concept. Table 7 shows the results of four methods. Again, we see from the table that the performance of MultiComm method is better than those of the other three methods on both networks in discovery of category-concept related communities.

5.5 Experiment 4

In this experiment, we consider the MovieLens data set, and construct a five-dimensional network. The five dimensions/entities involve 10197 movies, 2113 users, 13222 tags, 95321 actors and 4060 directors. Here we study the three tensors (i.e., $S = 3$): movie-user-tag tensor, movie-actor tensor, and movie-director tensor, see Table 8. $\mathcal{A}^{(1)}$ is a tensor representing the interactions among the movies, users and tags with $l_1 = 3$, $j_1^{(1)} = 1$, $j_2^{(1)} = 2$ and $j_3^{(1)} = 3$. $\mathcal{A}^{(2)}$ is a tensor representing the interactions between the movies and actors with $l_2 = 2$, $j_1^{(2)} = 1$ and $j_2^{(2)} = 4$. $\mathcal{A}^{(3)}$ is a tensor representing the interactions between the movies and directors with $l_3 = 2$, $j_1^{(3)} = 1$ and $j_2^{(3)} = 5$. The following tensor equations are required to solve in order to compute the required probabilities in MultiComm:

$$\begin{aligned}\mathbf{x}_1 &= (1-\alpha) \left[\frac{1}{3} \mathcal{P}^{(1,1)} \mathbf{x}_2 \mathbf{x}_3 + \frac{1}{3} \mathcal{P}^{(2,1)} \mathbf{x}_4 + \frac{1}{3} \mathcal{P}^{(3,1)} \mathbf{x}_5 \right] + \alpha \mathbf{z}_1, \\ \mathbf{x}_2 &= (1-\alpha) \mathcal{P}^{(1,2)} \mathbf{x}_1 \mathbf{x}_3 + \alpha \mathbf{z}_2, \quad \mathbf{x}_3 = (1-\alpha) \mathcal{P}^{(1,3)} \mathbf{x}_1 \mathbf{x}_2 + \alpha \mathbf{z}_3, \\ \mathbf{x}_4 &= (1-\alpha) \mathcal{P}^{(2,2)} \mathbf{x}_1 + \alpha \mathbf{z}_4, \quad \mathbf{x}_5 = (1-\alpha) \mathcal{P}^{(3,2)} \mathbf{x}_1 + \alpha \mathbf{z}_5.\end{aligned}$$

TABLE 5

The descriptions of SIAM multi-dimensional network and DBLP multi-dimensional network. The notation nz represents the number of nonzero entries in the corresponding tensor.

SIAM	paper-author-keyword	paper citation	author collaboration	paper-category concept
	$3736 \times 1807 \times 7630$ nz=31257	3736×3736 nz=4523	1807×1807 nz=4410	3736×1202 nz=11897
DBLP	$2362 \times 1380 \times 3751$ nz=17317	2362×2362 nz=3819	1380×1380 nz=7396	2362×316 nz=3727

TABLE 6

The results of four methods on SIAM and DBLP data sets for journal/conference communities. For SIAM data, there are 899 papers, 1416 papers and 1421 papers from SJAMEL, SJNAAM and SJOCE3 respectively. For DBLP data, there are 992 papers and 1370 papers from KDD and SIGIR respectively.

Results on SIMA data								
Community	F-measure				NMI			
	MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
SJAMEL	0.3544	0.0612	0.0022	0.6396	0.0079	0.0619	0.0103	0.2261
SJNAAM	0.4591	0.4321	0.0001	0.5113	0.0164	0.0052	0.0032	0.0340
SJOCE3	0.3868	0.3047	0.0056	0.4321	0.0001	0.0014	0.0013	0.0052
average F-measure/NMI	0.4001	0.2660	0.0026	0.5277	0.0081	0.0228	0.0049	0.0884
Results on DBLP data								
Community	F-measure				NMI			
	MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
KDD	0.5741	0.7662	0.0020	0.7722	0.0384	0.2740	0.0073	0.2869
SIGIR	0.6497	0.6664	0.0015	0.7350	0.0384	0.0521	0.0001	0.1016
average F-measure/NMI	0.6119	0.7163	0.0018	0.7536	0.0384	0.1631	0.0037	0.1942

TABLE 7

The results of four methods on SIAM and DBLP data sets for category concept communities. For SIAM data, there are 665 papers, 355 papers, 670 papers, 262 papers and 274 papers labeled as 65F10, 65F15, 65N30, 65N15 and 65N55 respectively. For DBLP data, there are 137 papers, 77 papers, 105 papers and 243 papers labeled as clustering, classifier design and evaluation, information filtering and retrieval models respectively.

Results on SIMA data								
Community(AMS code)	F-measure				NMI			
	MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
65F10	0.2123	0.1143	0.0090	0.4602	0.0107	0.0070	0.0134	0.1047
65F15	0.1641	0.2197	0.0222	0.4282	0.0077	0.0237	0.0333	0.1391
65N30	0.2133	0.0701	0.0059	0.3672	0.0116	0.0224	0.0026	0.0483
65N15	0.1341	0.0802	0.0152	0.3244	0.0047	0.0002	0.0418	0.0890
65N55	0.1743	0.2555	0.0001	0.3321	0.0135	0.0494	0.0008	0.0911
average F-measure/NMI	0.1796	0.1480	0.0105	0.3824	0.0096	0.0205	0.0184	0.0944
Results on DBLP data								
Community(Category Concept)	F-measure				NMI			
	MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
clustering	0.1424	0.0949	0.0284	0.2336	0.0103	0.0301	0.0246	0.0487
classifier design and evaluation	0.0938	0.0667	0.0189	0.1688	0.0058	0.0013	0.0058	0.0367
information filtering	0.1125	0.1048	0.0360	0.1905	0.0063	0.0082	0.0194	0.0379
retrieval models	0.1663	0.2305	0.0082	0.3086	0.0063	0.0243	0.0275	0.0582
average F-measure/NMI	0.1288	0.1242	0.0229	0.2254	0.0072	0.0160	0.0193	0.0454

TABLE 8

The description of MovieLens multi-dimensional network.

MovieLens	movie-user-tag	movie-actor	movie-director
	$10197 \times 2113 \times 13222$ nz=47957	10197×95321 nz=231742	10197×4060 nz=10155

TABLE 9

The results of four comparison methods on MovieLens data sets. There are 5076 movies, 1664 movies, 3566 movies, 1644 movies and 279 movies labeled as drama, thriller, comedy, romance and animation respectively.

Results on MovieLens								
Community (Genre)	F-measure				NMI			
	MetaFac	Clauset	LWP	MultiComm	MetaFac	Clauset	LWP	MultiComm
Drama	0.1460	0.5028	0.0012	0.5577	0.0045	0.0007	0.0048	0.0103
Thriller	0.1104	0.1851	0.0012	0.2476	0.0035	0.0007	0.0084	0.0104
Comedy	0.1074	0.3870	0.0006	0.3828	0.0014	0.0025	0.0003	0.0020
Romance	0.1057	0.1928	0.0012	0.2257	0.0019	0.0015	0.0021	0.0062
Animation	0.0758	0.0573	0.0212	0.2832	0.0047	0.0029	0.0404	0.1049
average F-measure/NMI	0.1091	0.2650	0.0051	0.3394	0.0032	0.0017	0.0112	0.0268

For each movie, several genres are given to indicate its styles. Our aim is to identify genre related communities in this multi-dimensional network. We report the values of F-measure and NMI computed based on evaluation of genres associated with the set of movies in the community. In MetaFac, we set the number of decompositions to be 10, and select the decomposition that corresponds to the largest F-measure for each testing genre. Table 9 shows the results of four methods on five genres. We see from the table that the performance of MultiComm is better than those of the other three methods except for the genre ‘‘Comedy’’. However, the values of F-measure and NMI for the ‘‘Comedy’’ community by MultiComm and by Clauset are quite close. The average values of F-measure and NMI by MultiComm are still higher than those by the other three methods.

6 CONCLUDING REMARKS

In this paper, we have proposed a framework (MultiComm) to determine communities in a multi-dimensional network based on probability distribution of each dimension/entity computed from the network. Both theoretical and experimental results have demonstrated that the proposed algorithm is efficient and effective.

On the other hand, in social networks, user actions are constantly changing and co-evolving. In the future work, it is required to adapt the proposed model to be time-varying. As probability distributions are non-stationary in this situation, we must consider and study statistically dependence in time-varying Markov chains [32] for items of different dimensions to obtain the affinities among them in order to find an evolution of communities across different time stamps.

APPENDIX A PROOF OF THEOREM 1

We let $n = n_1 + n_2 + \dots + n_m$, $\Omega_s = \{\mathbf{x}_s = (x_1, x_2, \dots, x_{n_s}) \in \mathbb{R}^{n_s} | x_i \geq 0, 1 \leq i \leq n_s, \sum_{i=1}^{n_s} x_i = 1\}$ for $s = 1, 2, \dots, m$, and also set $\Omega = \{[\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^n | \mathbf{x}_s \in \Omega_s, s = 1, 2, \dots, m\}$. We note that Ω_k and Ω are closed convex sets.

The problem can be reduced to a fixed point problem as follows. We define the following mapping $\Phi : \Omega \rightarrow \Omega$ as follows

$$\begin{aligned} & \Phi([\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]) \\ &= (1-\alpha) \left[\frac{1}{|N_1|} \sum_{s=1, j_t^{(s)}=1}^S \mathcal{P}^{(s,t)} \otimes_{u=1, u \neq 1}^{l_s} \mathbf{x}_{j_u^{(s)}}, \dots, \right. \\ & \quad \left. \frac{1}{|N_m|} \sum_{s=1, j_t^{(s)}=m}^S \mathcal{P}^{(s,t)} \otimes_{u=1, u \neq m}^{l_s} \mathbf{x}_{j_u^{(s)}} \right] \\ & \quad + \alpha [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m] \end{aligned} \quad (5)$$

It is clear that when $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \Omega$, $\Phi([\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]) \in \Omega$, and Φ is continuous. According to the Brouwer Fixed Point Theorem, there exists a non-negative vector $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \Omega$ such that $\Phi([\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]) = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$.

Next we show that $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ is a positive vector. Suppose it is not positive, i.e., there exist some entries of $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ are zero. Let $J_v = \{i_v | [\mathbf{x}_v]_{i_v} = 0\}$ for $v = 1, 2, \dots, m$. It is obvious that there exists v such that J_v is a proper subset of $\{1, 2, \dots, n_v\}$. Let $\delta = \min_v \{\min_{i_v} \{[\mathbf{x}]_{i_v} | i_v \notin J_v\}\}$. We must have $\delta > 0$. As

$$\mathbf{x}_v = (1-\alpha) \left[\frac{1}{|N_v|} \sum_{s=1, j_t^{(s)}=v}^S \mathcal{P}^{(s,t)} \otimes_{u=1, u \neq t}^{l_s} \mathbf{x}_{j_u^{(s)}} \right] + \alpha \mathbf{z}_v,$$

without loss of generality, we assume that $[\mathbf{x}_v]_{i_v} = 0$. It is clear that when $[\mathbf{z}_v]_{i_v}$ is positive, then $[\mathbf{x}_v]_{i_v}$ must be positive. Now we establish the following inequality:

$$\begin{aligned} & \delta^{l_s-1} \underbrace{\sum_{i_{j_1^{(s)}} \notin J_{j_1^{(s)}}} \sum_{i_{j_2^{(s)}} \notin J_{j_2^{(s)}}} \dots \sum_{i_{j_{l_s}^{(s)}} \notin J_{j_{l_s}^{(s)}}} p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)}}_{\text{except the sum for } i_{j_t^{(s)}}} \\ & \leq \underbrace{\sum_{i_{j_1^{(s)}} \notin J_{j_1^{(s)}}} \sum_{i_{j_2^{(s)}} \notin J_{j_2^{(s)}}} \dots \sum_{i_{j_{l_s}^{(s)}} \notin J_{j_{l_s}^{(s)}}} p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)}}_{\text{except the sum for } i_{j_t^{(s)}}} \\ & \quad [\mathbf{x}_{j_1^{(s)}}]_{i_{j_1^{(s)}}} [\mathbf{x}_{j_2^{(s)}}]_{i_{j_2^{(s)}}} \dots [\mathbf{x}_{j_{l_s}^{(s)}}]_{i_{j_{l_s}^{(s)}}} \\ & \leq \underbrace{\sum_{i_{j_1^{(s)}}=1}^{n_{j_1^{(s)}}} \sum_{i_{j_2^{(s)}}=1}^{n_{j_2^{(s)}}} \dots \sum_{i_{j_{l_s}^{(s)}}=1}^{n_{j_{l_s}^{(s)}}} p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)}}_{\text{except the sum for } i_{j_t^{(s)}}} \\ & \quad [\mathbf{x}_{j_1^{(s)}}]_{i_{j_1^{(s)}}} [\mathbf{x}_{j_2^{(s)}}]_{i_{j_2^{(s)}}} \dots [\mathbf{x}_{j_{l_s}^{(s)}}]_{i_{j_{l_s}^{(s)}}} \\ & \quad \text{except the term } [\mathbf{x}_{j_t^{(s)}}]_{i_{j_t^{(s)}}} \end{aligned} \quad (6)$$

By (6), we have

$$\begin{aligned} & \frac{1}{|N_v|} \sum_{s=1, j_t^{(s)}=v}^S \delta^{l_s-1} \underbrace{\sum_{i_{j_1^{(s)}} \notin J_{j_1^{(s)}}} \sum_{i_{j_2^{(s)}} \notin J_{j_2^{(s)}}} \dots \sum_{i_{j_{l_s}^{(s)}} \notin J_{j_{l_s}^{(s)}}}}_{\text{except the sum for } i_{j_t^{(s)}}} \\ & \quad p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)} \end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{|N_v|} \sum_{s=1}^S \underbrace{\sum_{j_t^{(s)}=v}^{n_{j_1^{(s)}}} \sum_{j_1^{(s)}=1}^{n_{j_2^{(s)}}} \cdots \sum_{j_{l_s}^{(s)}=1}^{n_{j_{l_s}^{(s)}}} p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)}}_{\text{except the sum for } i_{j_t^{(s)}}} \\
&\quad [\mathbf{x}_{j_1^{(s)}}]_{i_{j_1^{(s)}}} [\mathbf{x}_{j_2^{(s)}}]_{i_{j_2^{(s)}}} \cdots [\mathbf{x}_{j_{l_s}^{(s)}}]_{i_{j_{l_s}^{(s)}}} \\
&\quad \text{except the term } [\mathbf{x}_{j_t^{(s)}}]_{i_{j_t^{(s)}}} \\
&= 0,
\end{aligned}$$

for all $i_{j_t^{(s)}} \in J_{j_t^{(s)}}$ (with $j_t^{(s)} = v$). Hence we have

$$p_{i_{j_1^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)} = 0$$

for all $i_{j_t^{(s)}} \in J_{j_t^{(s)}}$ and for all $i_{j_u^{(s)}} \notin J_{j_u^{(s)}}$ with $u \neq t$.

Thus the matrices $(p_{i_{j_t^{(s)}}, i_{j_2^{(s)}}, \dots, i_{j_{l_s}^{(s)}}}^{(s,t)}, \dots)$ are reducible. It implies that $\mathcal{P}^{(s,t)}$ is reducible. This is a contradiction. By the construction of $P^{(s,t)}$, it must be irreducible. Thus \mathbf{x}_v must be positive. The theorem follows. \square

In [33], it has been given a general condition which guarantees the uniqueness of the fixed point in the Brouwer Fixed Point Theorem, namely, (i) 1 is not an eigenvalue of the Jacobian matrix of the mapping, and (ii) for each point in the boundary of the domain of the mapping, it is not a fixed point. In our case, we have shown in Theorem 1 that all the fixed points of Φ are positive when $\mathcal{P}^{(s,t)}$ are irreducible, i.e., they do not lie on the boundary $\partial\Omega$ of Ω .

Suppose $0 < \alpha < 1$. If 1 is not the eigenvalue of the Jacobian of the mapping Φ , then \mathbf{x}_v for $v = 1, 2, \dots, m$ given in Theorem 1 are unique.

REFERENCES

- [1] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821, 2002.
- [2] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824, 2002.
- [3] M. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
- [4] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [5] S. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.
- [6] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [7] G. Flake, S. Lawrence, C. Giles, and F. Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002.
- [8] B. Yang, J. Liu and J. Feng. On the spectral characterization and scalable mining of network communities. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):326–337, 2012.
- [9] J. Ruan and W. Zhang. An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07)*, pages 643–648, IEEE Computer Society, 2007.
- [10] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, pages 647–656, ACM, 2007.
- [11] S. Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 5th SIAM International Conference on Data Mining (SDM'05)*, 2005.
- [12] J. Bagrow and E. Bollt. Local method for detecting communities. *Physical Review E*, 72(4):046108, 2005.
- [13] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(2):026132, 2005.
- [14] F. Luo, Y. Yang, C. Chen, R. Chang, J. Zhou, and R. Scheuermann. Modular organization of protein interaction networks. *Bioinformatics*, 23(2):207, 2007.
- [15] F. Luo, J.Z. Wang, E. Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6(4):387–400. IOS Press, 2008.
- [16] Y. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 527–536, ACM, 2009.
- [17] L. Tang, H. Liu, and J. Zhang. Identifying evolving groups in dynamic multimode networks. *IEEE Transaction on Knowledge and Data Engineering*, 24(1):72–85, 2012.
- [18] R. Andersen and K.J. Lang. Communities from seed sets. In *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, 223–232, 2006.
- [19] A. Mehler and S. Skiena. Expanding network communities from representative examples. *ACM Transactions on Knowledge Discovery from Data*, 3(2), 2009.
- [20] J. Chen and Y. Saad. Dense Subgraph Extraction with Application to Community Detection. *IEEE Transactions on Knowledge and Data Engineering*, DOI:10.1109/TKDE.2010.271, 2010.
- [21] Y. Chi, S. Zhu, Y. Gong, and Y. Zhang. Probabilistic polyadic factorization and its application to personalized recommendation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*, pages 941–950, ACM, 2008.
- [22] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1):6–20, 2009.
- [23] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 653–658, ACM, 2004.
- [24] H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, pages 404–413, ACM, 2006.
- [25] K. Macropol, T. Can, and A. Singh. RRW: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC bioinformatics*, 10(1):283, 2009.
- [26] J. Xia, D. Caragea, and W. Hsu. Bi-relational network analysis using a fast random walk with restart. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'09)*, pages 1052–1057, IEEE, 2009.
- [27] H. Tong, C. Faloutsos, and J. Pan. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346, 2008.
- [28] S. Ross. *Introduction to probability models*. Academic Press, 2007.
- [29] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1998.
- [30] T. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*, ACM, 2002.
- [31] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.
- [32] W. Ching and M. Ng. Markov chains: models, algorithms and applications. *International Series on Operations Research and Management Science*, 2006.
- [33] R. Kellogg. Uniqueness in the schauder fixed point theorem. *Proceedings of the American Mathematical Society*, 60:207–210, 1976.