

Nonnegative Multi-level Network Factorization for Latent Factor Analysis

Junyu Xuan, Jie Lu, *Senior Member, IEEE*, Xiangfeng Luo, *Member, IEEE*
and Guangquan Zhang

Abstract—Nonnegative Matrix Factorization (NMF) aims to factorize a matrix into two optimized nonnegative matrices and has been widely used for unsupervised learning tasks such as product recommendation based on a rating matrix. However, although networks between nodes with the same nature exist, standard NMF overlooks them, e.g., the social network between users. This problem leads to comparatively low recommendation accuracy because these networks are also reflections of the nature of the nodes, such as the preferences of users in a social network. Also, social networks, as complex networks, have many different structures. Each structure is a composition of links between nodes and reflects the nature of nodes, so retaining the different network structures will lead to differences in recommendation performance. To investigate the impact of these network structures on the factorization, this paper proposes four multi-level network factorization algorithms based on the standard NMF, which integrates the vertical network (e.g., rating matrix) with the structures of horizontal network (e.g., user social network). These algorithms are carefully designed with corresponding convergence proofs to retain four desired network structures. Experiments on synthetic data show that the proposed algorithms are able to preserve the desired network structures as designed. Experiments on real-world data show that considering the horizontal networks improves the accuracy of document clustering and recommendation with standard NMF, and various structures show their differences in performance on these two tasks. These results can be directly used in document clustering and recommendation systems.

Index Terms—Multi-level Network, Nonnegative Matrix Factorization, Complex Network

I. INTRODUCTION

MULTI-LEVEL network is a structure that is composed of vertically connected nodes with different characters and horizontally connected nodes with the same characters. It is an abstract structure, as shown in Fig. 1, that can be used to model the data from many areas. One

J. Xuan is with the Centre for Quantum Computation and Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia and the School of Computer Engineering and Science, Shanghai University, China (e-mail: Junyu.Xuan@student.uts.edu.au).

J. Lu is with the Centre for Quantum Computation and Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia (e-mail: Jie.Lu@uts.edu.au).

X. Luo is with the School of Computer Engineering and Science, Shanghai University, China. (e-mail: luoxf@shu.edu.cn).

G. Zhang is with the Centre for Quantum Computation and Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia (e-mail: Guangquan.Zhang@uts.edu.au).

example, in the text mining area, is author-paper-keyword mapping relations (vertical network) with authors' social network, paper citation network and keyword co-occurrence relation network (horizontal networks), as shown in Fig. 2. In recommender systems, a multi-level network is composed of tag-user-movie mapping relations (vertical network) with tag similarity network, user social network and movie similarity network (horizontal networks), as shown in Fig. 3. The multi-level network structure is very common, especially in a big data environment. Due to the Variety property of big data [1], multiple sources and multiple attributes appear to describe the same data. To comprehensively model this kind of data, all the sources and attributes need to be considered simultaneously and so a multi-level network is a good choice. However, it is commonly believed that multi-level networks are sparse and redundant, because the scale of the nodes is normally much larger than the scale of the links.

Considering the sparsity and redundancy of the multi-level network, it would be helpful if an *optimized* latent space for nodes could be found according to its structure. This task is called *factorization*. Take text mining as an example. Suppose we have a multi-level network structure with 1000 authors, 10000 papers, and 8000 keywords. If we find a 100-dimensional space and map all the nodes into this space, we can represent each node, i.e., an author, only by a 100-dimensional vector. The term *optimized* means that the multi-level network structure can be reconstructed by the new representations of nodes with minimum error, such as reconstructing a link between an author and a paper by the cosine similarity between two corresponding 100-dimensional vectors (new representations) for the author and paper. Under this constraint, it is believed that this new representation is not only more concise but also more intrinsic, because the original multi-level network structure as unchanged as possible (some redundant information is removed but the significant information is kept) [2], [3]. Many real-world applications can benefit from this new concise and intrinsic representation; for example,

- **Document Clustering:** after the factorization of the author-paper-keyword structure, we can cluster papers with similar content together, and then help to organize and retrieve the papers or webpages;
- **Recommendation:** after the factorization of the tag-user-movie structure, we can recommend a movie to a user by the similarity between this movie and user

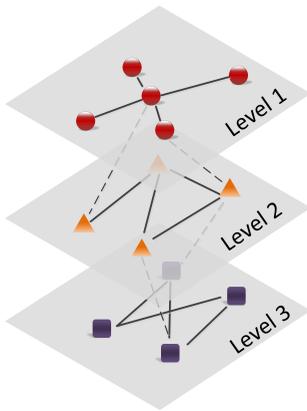


Fig. 1: Multi-level network

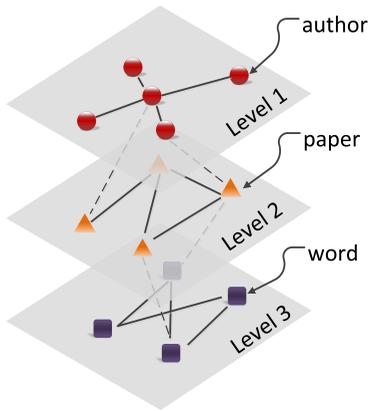


Fig. 2: An instance in Text mining

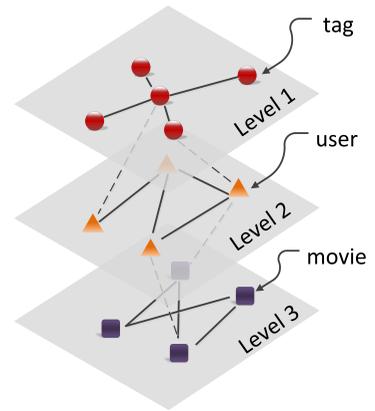


Fig. 3: An instance in Recommender Systems

using their concise representations, or recommend a tag to a user in the same fashion.

Motivated by these real-world applications, we want to design algorithms in this study that will define and implement multi-level network factorization.

Existing factorization algorithms cannot complete this task, because they overlook the horizontal network structures. The most classical algorithm is Nonnegative Matrix Factorization (NMF) [2]. NMF is an elegant tool for factorizing a matrix into two nonnegative matrices, which has been widely used in visual tracking [4], maximum margin classification [5], face recognition [6] and probabilistic clustering [7]. Two notable inference algorithms to resolve NMF are Alternative Least Square [8] and Multiplicative Iterative algorithms [9]. Here, we want to interpret NMF on a matrix M as a *bipartite network factorization* so that we can conduct our multi-level network factorization based on it. A matrix M can be seen as a bipartite network (vertical network) with rows as one kind of nodes and columns as another kind of nodes, while each element of matrix $m_{i,j}$ denotes a link between two nodes with different characters. The aim of NMF on a matrix is to find an optimized and relatively small latent space according to the bipartite network structure hidden in this matrix. However, the standard NMF cannot be directly used for the multi-level network because it only considers the vertical network. In fact, the difficulty of multi-level network factorization mainly lies in how to jointly conduct the vertical network factorization and horizontal network factorization.

In this paper, we propose four multi-level network factorization algorithms for preserving four different network structures of the horizontal network, including the whole network structure, community structure, degree distribution structure and max spanning tree structure. These structures are expressed by different link compositions which are sourced from the different natures of nodes. For example, if a user has a range of friends in a social network, it shows that he/she tends to have a range of preferences in movies. In this example, the friend relations of a user express the user's nature (preferences in movies). A natural problem is what is the difference between these network structures in the impact of the recommendation or cluster-

ing performance. To evaluate these differences, we conduct experiments to compare the performance of each on two real-world tasks. As the experimental results show, we can achieve better performance by preserving the structures instead of disregarding the horizontal network. We also compare the different performance of retaining different network structures on these tasks. Four cost functions are carefully designed to preserve the desired network structures. To optimize each cost function, the corresponding update equations are introduced with convergence analysis. Experiments on synthetic data show the designed algorithms have the ability to preserve the desired network structures. To show the usefulness of these algorithms, two real-world tasks, document clustering and recommendation, have been carried out. The results show that the proposed algorithms perform better than traditional NMF in achieving the clustering and recommendation accuracy.

The contributions of this paper are:

- 1) A general multi-level network which can jointly express the relations between nodes that have the same or different natures by the vertical network and horizontal networks is proposed to model different kinds of data;
- 2) Based on the nonnegative matrix factorization, four multi-level network factorization algorithms with their convergence proofs are carefully designed to discover the latent space, with different horizontal network structures as constraints.

The rest of this paper is organized as follows. Section II reviews related work. The multi-level network and its factorization are formally defined in Section III. Our algorithms for multi-level network factorization and their convergence analysis are proposed in Section IV. Experiments on synthetic data and real-world data are conducted in Section V. Lastly, Section VI concludes the study and discusses future work.

II. RELATED WORK

Since our motivation is to use complex network structures as constraints for nonnegative matrix factorization, this section is composed of two parts: 1) we will discuss elementary introductions to, and research on, the structures

of a complex network, and 2) we will discuss recent works on factorizations with networks.

A. One-level and Multi-level Complex Network

Complex network is an interdisciplinary research, attracting researchers from computer science, physics, sociology, biology, and so on. Due to the pervasiveness of the network phenomenon, complex network has been adopted to model many things, such as the users' friend network and the cell network in the brain. Comparing to the graph, the complex network area focuses more on the non-trivial structures. The two outstanding structures are small-world network published in Nature [10] and power-law degree distribution [11] published in Science. In fact, many other different network structures have also been discovered and defined in this area [12], [13]. However, it is commonly accepted that the following network structures are the most fundamental and significant for describing the structure of a complex network: community structure [14], [15], degree distribution [16]–[18], and max spanning tree [19], [20]. To the best of our knowledge, little work has been done to consider the influences of complex network structures on factorization.

Recently, the multi-layer/multi-level network has attracted the attention of researchers. Its mathematical formulation is given in [21]. Similar to the one-layer complex network, its structures are defined and discussed in [22]. Apart from formalization and structure definition, the multi-layer network has been used for modeling the influence propagation over microblogs [23] and the analysis and management of change propagation [24]. However, most state-of-the-art research of the researches on multi-layer/multi-level networks focuses on basic structure analysis. There is no work on factorization and its applications for document clustering or recommender systems.

Since the traditional network structures (i.e., community, degree distribution and max spanning tree) do not consider the directions of the edges in the network and our aim is to preserve these network structures after the factorization, we assume in this paper that the network is undirected.

B. Network-related factorization models/algorithms

The existing network-related models/algorithms in recommender systems and document clustering are mainly dominated by two renowned techniques: nonnegative matrix factorization and topic model.

1) *Network-related Nonnegative Matrix Factorization:* First, we give a brief introduction to nonnegative matrix factorization (NMF). Given a nonnegative matrix $Y_{a \times x}$ (extended to semi-nonnegative by [25]), the NMF aims to find two matrices $A_{a \times k}$ and $X_{k \times x}$ to minimize the following cost function

$$J(A, X) = \frac{1}{2} \|Y - AX\|_F^2, \quad (1)$$

where $\|\cdot\|_F$ is the Frobenius norm and the elements of A and X are also nonnegative. In the literature, constraints are added to the A or X in the cost function

in Eq. (1) for purposes such as sparseness constraint [3], smooth constraint [26], orthogonal constraint [27], and label information [28]. All these constraints aim to make the discovered (k -dimensional) latent space preserve more properties.

The networks/relations between data have been considered in a number of ways in NMF. One way is to let users define the must-link and cannot-link relations between data [29], and then use these relations as constraints for the NMF. Another way is to define a graph between data as the constraint of NMF (also called graph-embedding [30], [31] or graph-regularization [32], [33]). As we prove later, this constraint only preserves the community structure of network. Some works have jointly considered two-side information during factorization. For example, the constraint in [34] considers user similarity network and post content; the constraint in [35] considers graphs from multiple domains. There are also works on using NMF to find the community structures of a network [36] or two networks [37] but not as constraints, as in this paper. These works are similar to this paper, but they do not explore the structures of graphs. As we introduced in Section II.A, there are many important structures for a given network. However, these network structures are disregarded during factorization. Our contribution is that the different network structures are considered during factorization.

2) *Network-related Topic Models:* Topic models [38] were originally developed to discover the hidden topics in documents, which can also be seen as a kind of factorization. Recently, some extensions of the topic models have attempted to adapt the data to network structures. Since the social network and citation network are two explicit and commonly used networks, most topic models try to adapt to these two types of network. For the social network, the Author-Recipient-Topic (ART) model [39] has been proposed to analyse the categories of roles in the social network, based on the relations of people in the network. A similar task is investigated in [40]. The social network is inferred from informal chat-room conversations utilizing the topic model [41]. The 'noisy links' and 'popularity bias' of the social network are addressed by properly designed topic models in [42] and [43]. As an important issue of social network analysis, communities [44] are extracted by Social Topic Model (STM) [45]. Mixed Membership Stochastic Block model is another way to learn the mixed membership vector (i.e., topic distribution) for each node of the community structure [46]. For the citation network, Relational Topic Model (RTM) [47] is proposed to infer the topics and hierarchical topics from document networks by introducing a link variable between two documents. Unlike RTM, a block is adopted to model the link between two documents [48]. To retain the document network, Markov Random Field (MRF) is combined with topic model [49]. The communities in the citation network are also investigated [50].

Some models have also considered the hierarchical structure. Original topic model only consider two levels: document level and keyword level. Since there are many

features associated with documents, each feature can be seen as the third level in the hierarchical structure, including conference [51], time [52], author [53], entity [54], emotion [55] and other labels [56]. Through these specified models, not only can the topic distribution of each document be discovered, but the topic distribution of these features can also be learned. However, all these works disregard the networks. At each level, the items are considered as independent with each other.

To summarize, many researchers have noticed the importance of network structure, and it has been considered in a variety of models. From their work, we can see that the network structure indeed can help to unveil the nature of data. However, other complex network structures (i.e., degree distribution and max spanning tree) are overlooked.

III. PROBLEM DEFINITION

In this section, we will formally define and explain multi-level network and its factorization, as well as the problem encountered in multi-level network factorization. The designed algorithms to resolve this problem will be given in the subsequent section.

A. Multi-level Network

Definition 1 (Multi-level Network, Ω): Multi-level Network is composed of nodes with different characters. It includes horizontal networks between nodes with the same character and vertical networks between nodes with a different character, as shown in Fig. 1.

This is an abstract and very common model that can be used to model data from different areas. For example,

- in recommender systems, there is a multi-level network structure: tag-user-movie, in which users may have trust relations with each other, tags may have correlation relations with each other and movies may have similarity relations due to their genre information;
- in the text mining area, there is a multi-level network structure: author-paper-keyword, in which authors may have cooperation relations with each other, papers may have citation relations with each other and keywords may have semantic relations with each other.

B. Horizontal-network Factorization

Given a horizontal network $H_{n \times n}$ in Ω , and the element $h_{n_i, n_j} = w$ denotes that there is an edge between node n_i and node n_j with weight w . Against different backgrounds, the meanings of these edges will be different. For example, in recommender systems, there could be a user social network which expresses the trust relations between users; in the text mining area, there could be a paper citation network which expresses the citation relations between papers.

Definition 2 (Horizontal Network Factorization):

Horizontal network factorization projects the nodes to a lower dimension space and, at the same time, keeps

the original horizontal network structure as unchanged as possible. Based on the NMF, it can be expressed as

$$J(A) = \frac{1}{2} \|H - AA^T\|_F^2, \quad (2)$$

where $A_{n \times k}$ is a nonnegative matrix with each row corresponding to a node in the network $H_{n \times n}$. We can see that the nodes in $H_{n \times n}$ are all projected to a k -dimensional space as A . The cost function in Eq. (2) tries as far as possible to retain the network structure of H .

C. Vertical-network Factorization

Given a vertical network $V_{n \times p}$, the element v_{n_i, p_i} denotes that there is an edge between node n_i and node p_i . Against different backgrounds, the meanings of these edges will also be different. For example, in recommender systems, there could be user-movie relations due to user ratings on movies; in the text mining area, there could be author-paper relations as a result of authors writing papers.

Definition 3 (Vertical Network Factorization): The vertical network factorization also projects the nodes to a lower dimension space and, at the same time, keeps the original vertical network structure as unchanged as possible. Based on the NMF, it can be expressed as

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2, \quad (3)$$

where $A_{n \times k}$ is a nonnegative matrix with rows corresponding to one kind of node in the network $V_{n \times p}$ and $X_{k \times p}$ corresponding to the other kind of node in $V_{n \times p}$. We can see that nodes with different characters in $V_{n \times p}$ are all projected to a k -dimensional space. The cost function in Eq. (3) tries as far as possible to retain the network structure of V .

Note that the nonnegativity condition of A and X is necessary, because the underlying components A and X have their own physical interpretations. In the recommender system, A denotes the users' interests and X denotes the movies' properties.

D. Multi-level Network Factorization

The friend relations between users will apparently impact on users' ratings on movies, and the citation relations between papers will impact on the keyword usage of each paper. Thus, we need a way to combine them.

Definition 4 (Multi-level Network Factorization):

Multi-level network factorization is the combination of Horizontal-network factorization and Vertical-network factorization, and the final latent k -dimensional space is shared by two factorizations. Based on the NMF, it can be expressed as

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 + \alpha \cdot \frac{1}{2} \|H - AA^T\|_F^2, \quad (4)$$

where α is the parameter used to adjust the weights of two parts in the cost function.

It should be noted that H only represents one horizontal network. However, multi-level factorization can easily be

achieved by adding more items on the right side of Eq. (4). Here, we only discuss one level horizontal network for the sake of simplicity, but a comprehensive analysis of the situations with a different number of levels will be tested in the experiment section.

One problem when conducting multi-level network factorization is how to preserve the network structures, such as community structure, degree distribution and max spanning tree, after the projection to the new and relatively small k -dimensional latent space. Note that the minimization of Eq. (4) cannot ensure the different structures will definitely be preserved, as will be demonstrated in the experiments in Section V.

IV. PRESERVING COMPLEX NETWORK STRUCTURES DURING MULTI-LEVEL NETWORK FACTORIZATION

In this section, we will introduce four algorithms for preserving four different network structures during multi-level network factorization. There are two levels of nodes with vertical network, $V_{n \times p}$, but only one level horizontal network, $H_{n \times n}$, will be considered for brevity.

A. NNMF: Preserve the Whole Network Structure

In this situation, all edges in the network have the same status. The cost function is the same as Eq. (4). Since Eq. (4) contains second-order of matrix A , an approximation with lower computation complexity can be made by

$$P = \arg \min_{P \geq 0} \left\{ \frac{1}{2} \|H - PP^T\|_F^2 \right\} \quad (5)$$

and

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 + \alpha \frac{1}{2} \|P - A\|_F^2. \quad (6)$$

The first equation is a symmetric NMF for network H . We can then obtain an optimized latent space which preserves the whole network structure and the new representation, P , of nodes by this latent space.

The derivatives of Eq. (6) with respect to A is

$$\begin{aligned} \frac{\partial J}{\partial A} &= (-VX^T + AXX^T) + \alpha(A - P) \\ &= A(XX^T + \alpha \cdot I) - (VX^T + \alpha P) \end{aligned} \quad (7)$$

and according to KKT condition

$$\left[(-VX^T + AXX^T) + \alpha(A - P) \right]_{ij} A_{ij} = 0, \quad (8)$$

the update equation is set as

$$A_{ij}^{t+1} \leftarrow A_{ij}^t \sqrt{\frac{[VX^T + \alpha P^+]_{ij}}{[AXX^T + \alpha A + \alpha P^-]_{ij}}} \quad (9)$$

where $P^+ = (|P_{ij}| + P_{ij})/2$, $P^- = (|P_{ij}| - P_{ij})/2$ and $P = P^+ - P^-$. For X , the derivative is

$$\frac{\partial J}{\partial X} = -A^T V + A^T A X. \quad (10)$$

According to KKT condition

$$\left[-A^T V + A^T A X \right]_{ij} X_{ij} = 0, \quad (11)$$

the update equation for X is

$$X_{ij} \leftarrow X_{ij} \left[\frac{[A^T V]_{ij}}{[A^T A X]_{ij}} \right]^{1/2}. \quad (12)$$

Since the update equation of X is the same as traditional NMF, we only give the convergence proof of A .

Proof: According to Eq. (9), give the objective function with respect to A as

$$\begin{aligned} F(A) &= \text{tr} \left(-VX^T A^T + \frac{1}{2} A X X^T A^T + \frac{1}{2} \alpha \cdot A A^T \right. \\ &\quad \left. - \alpha \cdot P^+ A^T + \alpha \cdot P^- A^T \right) \end{aligned} \quad (13)$$

and define

$$\begin{aligned} G(A, A^t) &= \sum_{ij} -(VX^T)_{ij} A_{ij}^t (1 + \log \frac{A_{ij}}{A_{ij}^t}) \\ &\quad + \sum_{ij} \frac{(A^t X X^T)_{ij} A_{ij}^2}{2 A_{ij}^t} + \sum_{ij} \alpha \frac{(A_{ij}^t) A_{ij}^2}{2 A_{ij}^t} \\ &\quad - \sum_{ij} \alpha (P^+)_{ij} A_{ij}^t (1 + \log \frac{A_{ij}}{A_{ij}^t}) \\ &\quad + \sum_{ij} \alpha (P^-)_{ij} \frac{A_{ij}^2 + (A_{ij}^t)^2}{A_{ij}^t}. \end{aligned} \quad (14)$$

Then, $G(A, A^t)$ is the auxiliary function of $F(A)$, because the conditions

$$\begin{aligned} F(A^t) &= G(A^t, A^t) \\ G(A^t, A^t) &\geq G(A^{t+1}, A^t) \\ G(A^{t+1}, A^t) &\geq F(A^{t+1}) \end{aligned} \quad (15)$$

are satisfied when A^{t+1} takes the minimum value of $G(A, A^t)$ with respect to A_{ij} . The derivative of $G(A, A^t)$ is

$$\begin{aligned} \frac{\partial G(A, A^t)}{\partial A_{ij}} &= -\frac{(VX^T)_{ij} A_{ij}^t}{A_{ij}} + \frac{(A^t X X^T)_{ij} A_{ij}}{A_{ij}^t} \\ &\quad + \alpha \frac{A_{ij}^t A_{ij}}{A_{ij}^t} + \alpha \frac{(P^-)_{ij} A_{ij}}{A_{ij}^t} - \alpha \frac{P_{ij}^+ A_{ij}^t}{A_{ij}} \end{aligned} \quad (16)$$

and set A^{t+1} as the minimal value of $G(A, A^t)$ through $\frac{\partial G(A, A^t)}{\partial A_{ij}} = 0$. We have

$$A_{ij}^{t+1} = A_{ij}^t \sqrt{\frac{[VX^T + \alpha P^+]_{ij}}{[AXX^T + \alpha A + \alpha P^-]_{ij}}} \quad (17)$$

Therefore, we know that the update of A according to Eq. (9) will lead to the non-increasing of $J(A, X)$. \blacksquare

We can see from the update equations for A and X that, since we only consider one level network, the A is influenced by the horizontal network structure in Eq. (9). The update equation of X is the same as Eq. (12), since the new term does not impact on the derivative of cost function with respect to X .

Algorithm 1: NNMF**Input:** H and V , Maximum Iteration number: I_{\max} **Output:** A and X

```

1:  $P = \arg \min_{P \geq 0} \left\{ \frac{1}{2} \|H - PP^T\|_F^2 \right\};$ 
2: while  $i < I_{\max}$  do
3:   if  $\frac{\partial J}{\partial A_{ij}} < 0$  in Eq. (7) then
4:      $\bar{A}_{ij} = \max(A_{ij}, \sigma);$ 
5:   end if
6:   compute  $\bar{\eta}_A$  by Eq. (19);
7:    $A_{ij} \leftarrow A_{ij} - \bar{\eta}_A \frac{\partial J}{\partial A_{ij}};$ 
8:   if  $\frac{\partial J}{\partial X_{ij}} < 0$  in Eq. (10) then
9:      $\bar{X}_{ij} = \max(X_{ij}, \sigma);$ 
10:  end if
11:  compute  $\bar{\eta}_X$  by Eq. (21);
12:   $X_{ij} \leftarrow X_{ij} - \bar{\eta}_X \frac{\partial J}{\partial X_{ij}};$ 
13:   $i = i + 1;$ 
14: end while

```

Eq. (17) is equal to update A through the gradient descent method with a special step size [9], [57]. The corresponding step size of Eq. (17) is

$$\eta_A = A_{ij} \cdot \left[\left((AXX^T + \alpha A + \alpha P^-)_{ij}^{1/2} + (VX^T + \alpha P^+)_{ij}^{1/2} \right) \cdot (AXX^T + \alpha A + \alpha P^-)_{ij}^{1/2} \right]^{-1} \quad (18)$$

As pointed out in [57], [58], Eq. (17) might not converge to a stationary point due to improper step size. To ensure the convergence of the update, we revise this step size [57] as,

$$\bar{\eta}_A = \bar{A}_{ij} \cdot \left[\left((\bar{A}XX^T + \alpha \bar{A} + \alpha P^-)_{ij}^{1/2} + (VX^T + \alpha P^+)_{ij}^{1/2} \right) \cdot (\bar{A}XX^T + \alpha \bar{A} + \alpha P^-)_{ij}^{1/2} + \delta \right]^{-1} \quad (19)$$

where

$$\bar{A}_{ij} = \begin{cases} A_{ij}, & \text{if } \frac{\partial J}{\partial A_{ij}} \geq 0 \\ \max(A_{ij}, \sigma), & \text{if } \frac{\partial J}{\partial A_{ij}} < 0 \end{cases} \quad (20)$$

and δ and σ are two small positive numbers.

For X , the corresponding step size of Eq. (12) is

$$\bar{\eta}_X = \bar{X}_{ij} \cdot \left[\left((A^T A \bar{X})_{ij}^{1/2} + (A^T V)_{ij}^{1/2} \right) \cdot (A^T A \bar{X})_{ij}^{1/2} + \delta \right]^{-1} \quad (21)$$

The whole procedure is summarized in Algorithm 1.

B. CNMF: Preserve Community Structure

Community structure [13], [37], [59] is an important structural property of a complex network. Like the whole network structure, our idea is to project the nodes into a latent space which can preserve community structure rather than the whole network structure. Then, the problem becomes how to find a space to preserve the community structure. The Laplacian matrix, which is broadly used in spectral analysis [60], of the original network matrix, H , is considered here. One of its definition is,

$$L = D - H \quad (22)$$

$$(P, \lambda) = \text{svd}(L)$$

where D is the degree matrix defined as $d_{i,i} = \sum_j h_{i,j}$ and $d_{i,j} = 0 (i \neq j)$, $\text{svd}(\cdot)$ is the singular value decomposition operation, P is the eigenvectors and λ is eigenvalues.

With this latent space, P , the cost function can be easily designed as

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 + \alpha \cdot \frac{1}{2} \|P_k - A\|_F^2, \quad (23)$$

where P_k is the first k eigenvectors in P . Then, the update equations are the same as Eq. (9) and Eq. (12). Next, we try to prove the ability to preserve the community structure of the designed cost function.

Proof: P 's ability to preserve the community structure originates from graph cut theory. In this theory, separating a group of nodes into k subgroups is equal to optimizing the following cost function

$$\begin{aligned} & \min_G \text{RatioCut}(G_1, G_2, \dots, G_k) \\ &= \min_G \left\{ \frac{1}{2} \sum_{i=1}^k \frac{W(G_i, \bar{G}_i)}{|G_i|} \right\} \\ &= \min_Q \text{Tr}(Q^T L Q), \quad \text{s.t. } Q^T Q = I \end{aligned} \quad (24)$$

where $G = G_1, G_2, \dots, G_k$ is the partition of nodes, $|G_i|$ is the number of nodes in G_i , and W is a designed indicator matrix (more details can be found in [60]). With a small relaxation, the solution Q in Eq. (24) is just the matrix that contains the first k eigenvectors of L . That means our P_k can optimize the cost function in Eq. (24), and then can give the best partition of the network. Therefore, the cost function in Eq. (23) is able to preserve the community structure of the original network H . ■

It should be noted that the cost function in Eq. (23) has the same trend with

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 + \alpha \cdot \text{tr}(A^T L A). \quad (25)$$

This equation directly combines the NMF and Graph-Cut, which are commonly adopted by other researches [31], [32] for graph-embedding or graph-regularization. The difference between Eq. (23) and Eq. (25) is the same as the difference between Eq. (6) and Eq. (4).

The whole procedure is summarized in Algorithm 2.

Algorithm 2: CNMF**Input:** H and V , Maximum Iteration number: I_{\max} **Output:** A and X

```

1:  $L = D - H$ ;
2:  $P = \text{svd}(L)$ ;
3: while  $i < I_{\max}$  do
4:   if  $\frac{\partial J}{\partial A_{ij}} < 0$  in Eq. (7) then
5:      $\bar{A}_{ij} = \max(A_{ij}, \sigma)$ ;
6:   end if
7:   compute  $\bar{\eta}_A$  by Eq. (19);
8:    $A_{ij} \leftarrow A_{ij} - \bar{\eta}_A \frac{\partial J}{\partial A_{ij}}$ ;
9:   if  $\frac{\partial J}{\partial X_{ij}} < 0$  in Eq. (10) then
10:     $\bar{X}_{ij} = \max(X_{ij}, \sigma)$ ;
11:   end if
12:   compute  $\bar{\eta}_X$  by Eq. (21);
13:    $X_{ij} \leftarrow X_{ij} - \bar{\eta}_X \frac{\partial J}{\partial X_{ij}}$ ;
14:    $i = i + 1$ ;
15: end while

```

The update equation of A is set

$$A_{ij} \leftarrow A_{ij} \cdot \left[\left((AXX^T)_{ij}^2 + 8\alpha(AA^T \mathbf{1}A)_{ij}(VX^T + \alpha H \mathbf{1}A)_{ij} \right)^{1/2} - (AXX^T)_{ij} \right]^{\frac{1}{2}} / \left[(AA^T \mathbf{1}^{n \times n} A)_{ij} \right]^{\frac{1}{2}} \quad (31)$$

Proof: According to Eq. (29), give the objective function with respect to A as

$$F(A) = \text{tr} \left(-YX^T A^T + \frac{1}{2} AXX^T A^T - \alpha \cdot H \mathbf{1}^{n \times n} AA^T + \frac{1}{2} \alpha \cdot AA^T \mathbf{1}^{n \times n} AA^T \right) \quad (32)$$

C. DNMF: Preserve Degree Distribution Structure

The degree distribution [11], [13], [18] is another very important structure. A node, i , in a network will have a number of neighbors as, d_i . The degree distribution can be described by a function of $n_d \sim \pi(d)$, where n_d denotes the number of nodes with the same degree d and $\pi(\cdot)$ is the distribution of the number of degrees. Our idea to preserve the degree distribution is to maximize the correlation between the node degree sequences of the original matrix and the new generated matrix (formed by k -dimensional latent space).

Suppose the network matrix H is a binary matrix,

$$H_{n \times n} \mathbf{1}^{n \times 1} \quad (26)$$

will be a vector containing degrees of all nodes. Similarly,

$$AA^T \mathbf{1}^{n \times 1} \quad (27)$$

is also a vector containing degrees of all nodes in the k -dimensional latent space. The distance between two degree vectors can be evaluated by,

$$\|H_{n \times n} \mathbf{1}^{n \times 1} - AA^T \mathbf{1}^{n \times 1}\|_F^2 \quad (28)$$

If we can minimize this distance, the degree distributions in original space and new latent space will be similar. If the H is not a binary matrix but a real-valued matrix, the degree sequence can be seen as a *weighted degree distribution*. Then, a new cost function with this term is

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 + \frac{1}{2} \alpha \cdot \|H_{n \times n} \mathbf{1}^{n \times 1} - AA^T \mathbf{1}^{n \times 1}\|_F^2. \quad (29)$$

The derivative of A is

$$\frac{\partial J(A, X)}{\partial A} = -VX^T + AXX^T - \alpha \cdot H \mathbf{1}^{n \times 1} \mathbf{1}^{1 \times n} A + 2\alpha \cdot AA^T \mathbf{1}^{n \times 1} \mathbf{1}^{1 \times n} A \quad (30)$$

and define

$$G(A, A^t) = - \sum_{ij} \left((VX^T)_{ij} A_{ij}^t (1 + \log \frac{A_{ij}}{A_{ij}^t}) \right) + \frac{1}{2} \sum_{ij} \left(\frac{(A^t X X^T)_{ij} A_{ij}^2}{A_{ij}^t} \right) - \sum_{ij} \left(\alpha \cdot (H \mathbf{1}^{n \times n} A^t)_{ij} A_{ij}^t (1 + \log \frac{A_{ij}}{A_{ij}^t}) \right) + \frac{1}{2} \sum_{ij} \left(\alpha \cdot \frac{(A^t (A^t)^T \mathbf{1}^{n \times n} A^t)_{ij} A_{ij}^4}{(A_{ij}^t)^3} \right) \quad (33)$$

Then, $G(A, A^t)$ is the auxiliary function of $F(A)$, because Eq. (15) is satisfied. The derivative of $G(A, A^t)$ with respect to A is

$$\frac{\partial G(A, A^t)}{\partial A_{ij}} = - \frac{(VX^T)_{ij} A_{ij}^t}{A_{ij}} + \frac{(A^t X X^T)_{ij} A_{ij}}{A_{ij}^t} - \alpha \cdot \frac{((H \mathbf{1}^{n \times n}) A^t)_{ij} A_{ij}^t}{A_{ij}} + 2\alpha \cdot \frac{(AA^T \mathbf{1}^{n \times 1} \mathbf{1}^{1 \times n} A)_{ij} A_{ij}^3}{(A_{ij}^t)^3} \quad (34)$$

and set A^{t+1} through $\frac{\partial G(A, A^t)}{\partial A_{ij}} = 0$, and then the Eq. (31) is derived. ■

Algorithm 3: DNMF**Input:** H and V , Maximum Iteration number: I_{\max} **Output:** A and X

```

1: while  $i < I_{\max}$  do
2:   if  $\frac{\partial J}{\partial A_{ij}} < 0$  in Eq. (30) then
3:      $\bar{A}_{ij} = \max(A_{ij}, \sigma)$ ;
4:   end if
5:   compute  $\bar{\eta}_A$  by Eq. (35);
6:    $A_{ij} \leftarrow A_{ij} - \bar{\eta}_A \frac{\partial J}{\partial A_{ij}}$ ;
7:   if  $\frac{\partial J}{\partial X_{ij}} < 0$  in Eq. (10) then
8:      $\bar{X}_{ij} = \max(X_{ij}, \sigma)$ ;
9:   end if
10:  compute  $\bar{\eta}_X$  by Eq. (21);
11:   $X_{ij} \leftarrow X_{ij} - \bar{\eta}_X \frac{\partial J}{\partial X_{ij}}$ ;
12:   $i = i + 1$ ;
13: end while

```

The revised step size for the update in Eq. (31) is,

$$\bar{\eta}_A = \bar{A}_{ij} \cdot \left[(4\alpha \cdot (\overline{AA^T \mathbf{1}\bar{A}})_{ij})^{1/2} - \left((\overline{AXX^T})_{ij}^2 + 8\alpha (\overline{AA^T \mathbf{1}\bar{A}})_{ij} \cdot (VX^T + \alpha \cdot H\mathbf{1}\bar{A})_{ij} \right)^{1/2} - (\overline{AXX^T})_{ij} \right] / \left[(4\alpha \cdot (\overline{AA^T \mathbf{1}\bar{A}})_{ij})^{1/2} \cdot (-VX^T + \overline{AXX^T} - \alpha \cdot H\mathbf{1}\bar{A} + 2\alpha \cdot \overline{AA^T \mathbf{1}\bar{A}}) + \delta \right] \quad (35)$$

where \bar{A} satisfies Eq. (20) with $\frac{\partial J}{\partial A}$ replaced by Eq. (30).

The whole procedure is summarized in Algorithm 3.

D. TNMF: Preserve Max Spanning Tree Structure

The max spanning tree [13], [59] of network H is first mined,

$$T \circledast H = T^H \quad (36)$$

where \circledast denotes the Hadamard (element-wise) product and T^H is the mined max spanning tree of network H and T is called the *tree-mask matrix* which is a binary matrix with $t_{i,j} = 1$ if it is in T^H otherwise $t_{i,j} = 0$, and \bar{T} is the complement of T with $t_{i,j} = 1$ if it is not in T^H otherwise $t_{i,j} = 0$.

The cost function is

$$J(A, X) = \frac{1}{2} \|V - AX\|_F^2 - \alpha \cdot \frac{1}{4} \|T \circledast (AA^T) - \bar{T} \circledast (AA^T)\|_F^2. \quad (37)$$

The derivative with respect to A is

$$\frac{\partial J(A, X)}{\partial A} = (-VX^T + AXX^T) + \alpha \cdot (\bar{T} - T) \circledast (AA^T)A \quad (38)$$

The update equation for A is

$$A_{ij} \leftarrow A_{ij} \cdot \left[\left((AXX^T)_{ij}^2 + 4\alpha (\bar{T} \circledast (AA^T)A)_{ij} (VX^T + \alpha T \circledast (AA^T)A)_{ij} \right)^{1/2} - (AXX^T)_{ij} \right]^{1/2} / \left[2\alpha (\bar{T} \circledast AA^T A) \right]^{1/2} \quad (39)$$

The convergence proof of this update equation is given as follows.

Proof: According to Eq. (37), give the objective function with respect to A

$$F(A) = \text{tr} \left(-VX^T A^T + \frac{1}{2} AXX^T A^T + \frac{1}{4} \alpha \cdot ((\bar{T} - T) \circledast AA^T AA^T) \right) \quad (40)$$

and define

$$G(A, A^t) = - \sum_{ij} \left((VX^T)_{ij} A_{ij}^t (1 + \log \frac{A_{ij}}{A_{ij}^t}) \right) + \frac{1}{2} \sum_{ij} \left(\frac{(A^t X X^T)_{ij} A_{ij}^2}{(A_{ij}^t)} \right) - \frac{1}{4} \alpha \sum_{ijkl} \left(T_{il} A_{il}^t A_{kl}^t A_{ik}^t A_{ij}^t \left(1 + \log \frac{A_{il} A_{kl} A_{ik} A_{ij}}{A_{il}^t A_{kl}^t A_{ik}^t A_{ij}^t} \right) \right) + \frac{1}{4} \alpha \sum_{ij} \left(\frac{(\bar{T} \circledast (A^t (A^t)^T) A^t)_{ij} (A_{ij}^t)^4}{(A_{ij}^t)^3} \right). \quad (41)$$

Then, $G(A, A^t)$ is the auxiliary function of $F(A)$, because Eq. (15) is satisfied. The derivative of $G(A, A^t)$ with respect to A is

$$\frac{\partial G(A, A^t)}{\partial A_{ij}} = - \frac{(VX^T)_{ij} A_{ij}^t}{A_{ij}} + \frac{(A^t X X^T)_{ij} A_{ij}}{(A_{ij}^t)} - \alpha \frac{(T \circledast (A^t (A^t)^T) A^t)_{ij} A_{ij}^t}{A_{ij}} + \alpha \frac{(\bar{T} \circledast (A^t (A^t)^T) A^t)_{ij} A_{ij}^3}{(A_{ij}^t)^3} \quad (42)$$

and set A_{ij}^{t+1} as the minimal value of $G(A, A^t)$ through $\frac{\partial G(A, A^t)}{\partial A_{ij}} = 0$. Then, we get Eq. (39). Therefore, we know that the update of A according to Eq. (39) will lead to the non-increasing of $J(A, X)$. \blacksquare

Algorithm 4: TNMF**Input:** H and V , Maximum Iteration number: I_{\max} **Output:** A and X

- 1: Obtain Max spanning tree T of H ;
- 2: **while** $i < I_{\max}$ **do**
- 3: **if** $\frac{\partial J}{\partial A_{ij}} < 0$ in Eq. (38) **then**
- 4: $\bar{A}_{ij} = \max(A_{ij}, \sigma)$;
- 5: **end if**
- 6: compute $\bar{\eta}_A$ by Eq. (43);
- 7: $A_{ij} \leftarrow A_{ij} - \bar{\eta}_A \frac{\partial J}{\partial A_{ij}}$;
- 8: **if** $\frac{\partial J}{\partial X_{ij}} < 0$ in Eq. (10) **then**
- 9: $\bar{X}_{ij} = \max(X_{ij}, \sigma)$;
- 10: **end if**
- 11: compute $\bar{\eta}_X$ by Eq. (21);
- 12: $X_{ij} \leftarrow X_{ij} - \bar{\eta}_X \frac{\partial J}{\partial X_{ij}}$;
- 13: $i = i + 1$;
- 14: **end while**

The revised step size for the update in Eq. (31) is,

$$\begin{aligned} \bar{\eta}_A = \bar{A}_{ij} \cdot & \left[\left(2\alpha \cdot (\bar{T} \otimes (\overline{AA^T})\bar{A})_{ij}^{1/2} - \left((\overline{AXX^T})_{ij}^2 \right. \right. \right. \\ & \left. \left. \left. + 4\alpha (\bar{T} \otimes (\overline{AA^T})\bar{A})_{ij} (VX^T + \alpha T \otimes (\overline{AA^T})\bar{A})_{ij} \right)^{1/2} \right. \right. \\ & \left. \left. - (\overline{AXX^T})_{ij} \right] / \left[\left(2\alpha \cdot (\bar{T} \otimes (\overline{AA^T})\bar{A})_{ij}^{1/2} \right. \right. \right. \\ & \left. \left. \left. \cdot \left(-VX^T + \overline{AXX^T} + \alpha \cdot (\bar{T} - T) \otimes (\overline{AA^T})\bar{A} \right)_{ij} + \delta \right] \right. \end{aligned} \quad (43)$$

where \bar{A} satisfies Eq. (20) with $\frac{\partial J}{\partial A}$ replaced by Eq. (38).

The whole procedure is summarized in Algorithm 4.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section is composed of two parts: experiments on synthetic data and experiments on real-world data. The first part is used to verify the correctness of our proposed algorithms. The second part is used to show the usefulness of our work. For the sake of brevity, abbreviations are used as follows: NNMF for the algorithm proposed in Section IV(A) to preserve the whole network structure; CNMF for the algorithm proposed in Section IV(B) to preserve the community structure; DNMF for the algorithm proposed in Section IV(C) to preserve the degree distribution structure; TNMF for the algorithm proposed in Section IV(D) to preserve the max spanning tree structure.

A. Experiments on Synthetic data

In this section, we randomly construct two matrices, V and H , to verify the proposed algorithms, including the convergence of algorithms and the ability to preserve the desired network structures.

1) *Convergence analysis:* Although we have given the proofs for the convergence of the proposed algorithms in Section IV, the real values of cost functions are given in this section to show the convergence of the algorithms under the designed update equations. First, the matrices, V and H , are randomly generated. Four algorithms are used to factorize them. The value after each iteration is recorded. The number of iterations is set as 10,000. It should be noted that the iterations may stop before this number is reached due to the stationary (smaller than $1e - 10$) between the values of the cost function before and after one iteration. We can see from Fig. 4 that the values of the cost functions will decrease under the designed update equations of the four algorithms in line with our proofs.

2) *Test for community structure preservation:* For a given pair of matrices $\langle V, H \rangle$, H denotes a network. Suppose we want to factorize $\langle V, H \rangle$ and preserve the community structure of network H at the same time. To show the ability of CNMF to keep the community structure of the original network H , we compare the communities of the re-constructed networks (H_{NNMF} and H_{DNMF}) with the communities of the original network H . For quantification purposes, we use clustering evaluation metrics. The evaluation metrics of document clustering are Jaccard Coefficient (JC), Folkes&Mallows (FM) and F1 measure (F1). Given a clustering result,

- a is the number of two points that are in the same cluster of both benchmark and clustering results;
- b is the number of two points that are in the same cluster of benchmark but in different clusters of clustering results;
- c is the number of two points that are not in the same cluster of both benchmark but in the same cluster of clustering results.

and three metrics are computed by equations in Table I (bigger means better).

We randomly generate 1000 pairs of matrices $\langle V, H \rangle$ for each size (10 and 100). The comparison between the communities of re-constructed network with the communities of the original network are shown in Table II. From these results, we can draw the conclusion that CNMF preserves the community structure better than NNMF.

TABLE I: Evaluation metrics of document clustering/communities

Jaccard Coefficient	$JC = \frac{a}{a+b+c}$
Folkes & Mallows	$FM = \left(\frac{a}{a+b} \cdot \frac{a}{a+c} \right)^{1/2}$
F1 measure	$F1 = \frac{2a^2}{2a^2+ac+ab}$

3) *Test for degree distribution preservation:* A pair of matrices are randomly generated $\langle V, H \rangle$. H is the original network whose degree distribution we want to keep. After running both NNMF and DNMF, we obtain A_{NNMF} and A_{DNMF} which can re-construct the network by $H_{NNMF} = A_{NNMF}A_{NNMF}^T$ and $H_{DNMF} =$

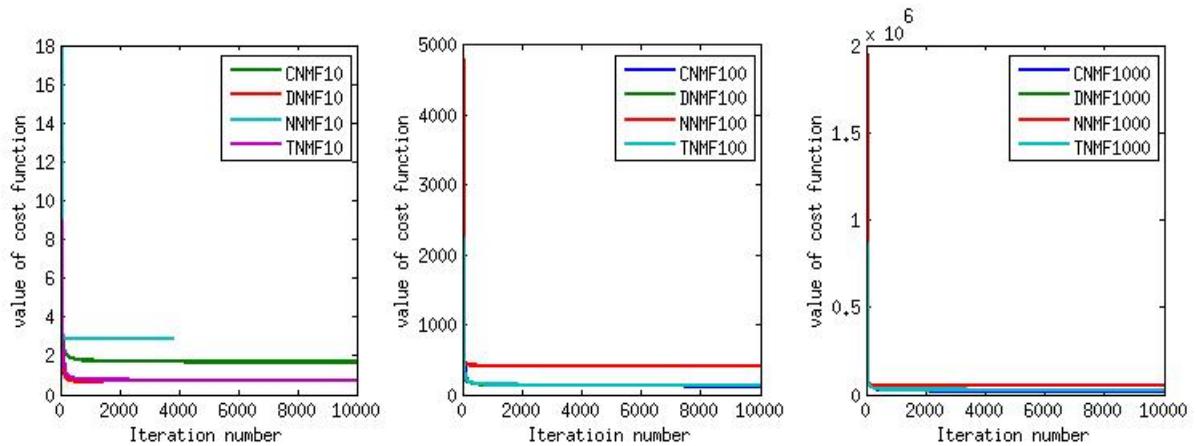


Fig. 4: The values of cost functions under the designed update equations. Three figures denote the different values of the dimension numbers of latent space, 10, 100 and 1000, respectively.

TABLE II: Comparison between NNMF and CNMF on preserving community structure

Algorithm	n=10 (K=3)			n=100 (K=10)		
	JC	FM	F1	JC	FM	F1
NNMF	0.1914 ± 0.0642	0.3240 ± 0.0930	0.3165 ± 0.0879	0.1894 ± 0.0200	0.3701 ± 0.0487	0.3180 ± 0.0282
CNMF	0.2542 ± 0.0667	0.5172 ± 0.0885	0.4615 ± 0.0870	0.3437 ± 0.0952	0.5209 ± 0.0983	0.5044 ± 0.1022

$A_{DNMF}A_{DNMF}^T$. To show the ability of DNMF to retain the degree distribution of the original network H , we compute the correlation between the degree distributions of the re-constructed networks (H_{NNMF} and H_{DNMF}) and the original network H . If the degree distribution of H_{DNMF} has larger correlation coefficient with the degree distribution of H than the degree distribution of H_{NNMF} , we can draw the conclusion that DNMF preserves degree distribution better than H_{NNMF} . Here, we consider two matrix sizes: 10 and 100. For each size, we randomly generate 1000 pairs of matrices. The number of hidden factors is set as 3 and 10. The average and standard deviation of results are given in Table III. These numbers show that the DNMF preserves more of the network degree distribution structure than NNMF.

TABLE III: Comparison between NNMF and DNMF on preserving of degree distribution structure

Algorithm	n = 10 (K=3)	n = 100 (K=10)
NNMF	0.0658 ± 0.3314	0.0045 ± 0.0980
DNMF	0.5548 ± 0.2531	0.9079 ± 0.0190

4) *Test for max spanning tree preservation:* For the max spanning tree, we first randomly generate a pair of $V_{100 \times 100}$ and $H_{100 \times 100}$. Using NNMF and TNMF to factorize V and H obtain two A_{NNMF} and A_{TNMF} which can re-construct the network H by $H_{NNMF} = A_{NNMF}A_{NNMF}^T$ and $H_{TNMF} = A_{TNMF}A_{TNMF}^T$. After extracting max spanning trees of three networks H , H_{NNMF} and H_{TNMF} , we can compare the similarity between the re-constructed trees (T_{NNMF} and T_{TNMF}) and the benchmark T_H . An illustrative example is shown in Fig. 5, in which each point in the matrix denotes an edge between two nodes. To quantify the ability of TNMF

to retain the tree structure, we randomly generated 1000 pairs of V and H for each size: 10 and 100. For each pair $\langle V, H \rangle$, we run both NNMF and TNMF, and compute the similarity between the re-constructed trees with the benchmark tree by the number of overlap edges as

$$s(T_H, T_{NNMF}) = \text{sum}(T_H \otimes T_{NNMF}) \quad (44)$$

where $\text{sum}(M)$ is to count the number of 1 in matrix M . The number of hidden factors are set as 3 and 10. If the re-constructed max spanning tree from TNMF has more overlap edges with the original max spanning tree than the tree from NNMF, we draw the conclusion that TNMF is able to preserve the max spanning tree better than NNMF. The results are shown in Table IV. The max in the table denotes the maximum number of edges in the max spanning tree. For a network with n nodes, the maximum number of edges in the max spanning tree is $n - 1$.

TABLE IV: Comparison between NNMF and TNMF on preserving max spanning tree structure

Algorithm	n = 10 (max = 9)	n = 100 (max = 99)
NNMF	2.5 ± 1.3	21.4 ± 6.3
TNMF	6.2 ± 0.7	56.1 ± 7.2

B. Experiments on Real-world Data

The above section shows the convergence and abilities to preserve the desired network structures. In this section, we will compare the efficiency of the proposed algorithms for real-world applications, including document clustering and recommendation.

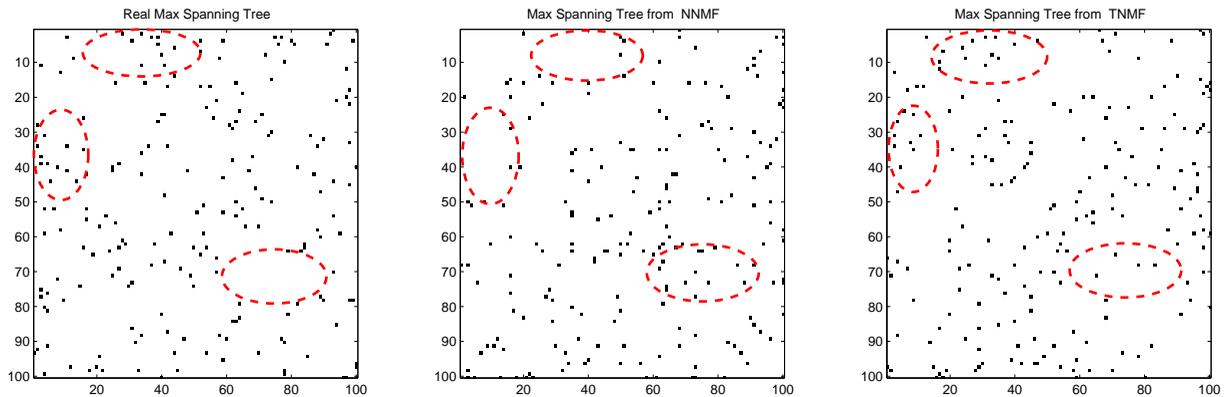


Fig. 5: Comparison between NNMF and TNMF on the ability to preserve the max spanning tree. Each point in the figure denotes a link in the network. The left figure denotes the real max spanning tree of the original network; the centre figure denotes the max spanning tree from NNMF; the right figure denotes the max spanning tree from TNMF. Parts of them are highlighted as an ellipse (red). We can make an approximate comparison through these three ellipses.

1) *Document Clustering with One-level Network*: Our document data is *Cora*¹, which is a public dataset and consists of 2708 scientific publications classified into one of seven classes. One-level network is composed by a horizontal network and a vertical network. The horizontal network, $H_{2708 \times 2708}$, is the citation relations between publications and it consists of 5429 links. This vertical network, $V_{2708 \times 1433}$, is constructed by the mapping relation between documents and words. The detailed statistics are shown in Table V.

TABLE V: Statistics of *Cora* dataset

Number of documents	2,708
Number of keywords	1,433
Number of links	5,429
Number of classes	7

After the factorization of $V_{2708 \times 1433}$ and $H_{2708 \times 2708}$ through our proposed algorithms, NNMF, CNMF, DNMF and TNMF, the documents are projected into a latent space and then given new representations. It is believed that the different classes are formed as a result of the intrinsic of the documents, so if the discovered latent space is good enough, it will cause the documents to cluster into these seven classes. According to this idea, we conduct the document clustering through the learned matrix A (the new representations of documents) by the k-means clustering algorithm. To compare the efficiency, we also implement standard NMF which does not consider the horizontal network $H_{2708 \times 2708}$ and Relational Topic Model (RTM) [47] which is a successful probabilistic Bayesian model for the $H_{2708 \times 2708}$ and $V_{2708 \times 1433}$. RTM can also be seen as a method for the factorization but from a probabilistic view, which also does not consider the network structure.

The final results are shown in Fig. 6. Three subfigures denote three clustering result comparisons by the metric in Table I. We have tested four numbers of factors: $K = 100$,

$K = 300$, $K = 500$ and $K = 1000$. In each subfigure in Fig. 6, we have compared the results from NNMF, CNMF, DNMF, TNMF, NMF and RTM on the clustering evaluation metric. Except for NMF, the algorithms all consider the effects from the horizontal network $H_{2708 \times 2708}$. From this result, we can see that NMF achieves the worst performance compared to others. Thus, we can draw the conclusion that incorporating the citation network is helpful for the clustering of the publications. Except for DNMF, which has similar performance to RTM, NNMF, CNMF and TNMF are better than RTM on this document clustering task. Notably, CNMF and TNMF achieve the best performance of all the algorithms. The reason is that the community structure of CNMF is beneficial for the clustering because it encourages ‘similar’ nodes to cluster together. For example, two documents d_i and d_j are in the same community in network H due to their ‘similarity’. Retaining the community structure will make it more possible for d_i and d_j to remain within the same cluster under the new factor representations. In the TNMF, preserving max spanning tree encourages the most important relations of all the nodes/documents. These relations in the tree can be seen as the ‘bones’ of a network, which determine the weighted distances between the nodes (documents). Therefore, the TNMF can benefit for the document clustering. It should be noted that although the network structure will influence the final clustering results, this influence will be under constraint from the document-word mapping network V . Each document will exhibit two natures from two networks: H and V . If the two natures are consistent, the constraint from V will help to enhance the learned network structure from H ; If the two natures are not consistent, there will be a contradiction between H and V , which will prevent the network structure learning from H .

2) *Recommendation with One-level Network*: A public dataset is adopted, *Lastfm*², which is a commonly used dataset for evaluating algorithms for recommender systems.

¹<http://linqs.cs.umd.edu/projects/projects/lbc/>

²<http://labrosa.ee.columbia.edu/millionsong/lastfm>

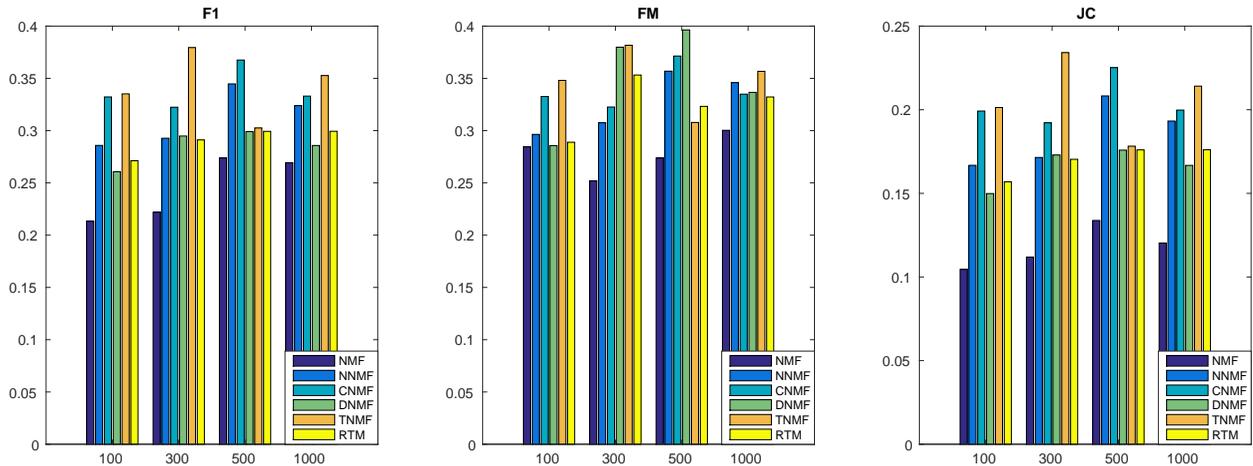


Fig. 6: Comparison of document clustering on different values of K . The influence of K values with $\alpha = 0.1$. ($K = 100$, $K = 300$, $K = 500$, and $K = 1000$)

In this dataset, a vertical network $V_{1892 \times 17632}$ is formed by users and artists, and V_{ij} represents the count of user i listening artist j . The horizontal network $H_{1892 \times 1892}$ is the user friend network. The statistics are shown in Table VI. A certain number of user-artist pairs are retained as the test data. The task is to predict the counts of users listening to these artists.

TABLE VI: Statistics of *Lastfm* dataset

number of users	1892
number of artists	17632
number of friend relations	12717
test user-artist pairs	5000

The evaluation metric is Mean Absolute Error (MAE) which is the simplest and the most intuitive. The definition is

$$MAE = \frac{1}{N} \sum_{u,i} |\hat{r}_{u,i} - r_{u,i}|, \quad (45)$$

where N is the number of test user-artist parts, $r_{u,i}$ is the real count and $\hat{r}_{u,i}$ is the predicted count. The correlation coefficient is computed by

$$\rho = \frac{\sum_i (r_i - \bar{r})(\hat{r}_i - \bar{\hat{r}})}{\sqrt{\sum_i (r_i - \bar{r})^2 \sum_i (\hat{r}_i - \bar{\hat{r}})^2}}. \quad (46)$$

To show the performance of the proposed algorithms on the recommendation, we compared them with a state-of-the-art method: Graph regularized NMF (GNMF) [31], [32]. The results are shown in Fig. 7. The standard NMF (without the horizontal network $H_{1892 \times 1892}$) has the worst performance of all the algorithms, which is similar to the document clustering in Section V.B(1). We can therefore draw the conclusion that incorporating user friend network $H_{1892 \times 1892}$ improves the performance of the recommendation. GNMF has the same property as CNMF for preserving the community structure of $H_{1892 \times 1892}$, as discussed in Section IV.B. The results in Fig. 7 also show

that GNMF has similar performance to CNMF. Although DNMF has good performance on MAE, the correlation with DNMF is the worst of all the proposed algorithms and the state-of-the-art GNMF. This reflects that DNMF predicts accurate values for some test data but not all data. The reason is that DNMF preserves the network degrees of all the nodes/users. The nodes with relatively large degrees will tend to have large weights in the weighted degree distribution, and DNMF will have a bias toward these nodes during factorization. Therefore, DNMF tends to achieve good results for the nodes/users with big degrees. In all the algorithms, TNMF achieves the best performance both on MAE and Correlation. As discussed in Section V.B(1), preserving max spanning tree encourages the most important relations (the ‘bones’ of a network) of all the nodes/users. These relations reflect not only the ‘distances’ between nodes but also the degrees of nodes.

3) *Document Clustering with Two-level Network*: First, we introduce the dataset we use. Documents are a collection of papers from *CiteSeer* [61]. There are 3312 papers in the whole corpus. Each paper is represented by a binary vector using words. The labels of these papers are set as their research areas, such as AI (Artificial Intelligence), ML (Machine Learning), Agents, DB (Database), IR (Information Retrieval) and HCI (Human-Computer Interaction). The statistics are shown in Table VII.

TABLE VII: Statistics of *CiteSeer* dataset

Number of documents	3,312
Number of keywords	3,703
Number of classes	6

A two-level network is composed by two horizontal networks $H_{3312 \times 3312}$ and $H_{3703 \times 3703}$ and one vertical network $V_{3312 \times 3703}$, and they are constructed as following. The first level horizontal network $H_{3312 \times 3312}$ is a paper citation network, which is formed by the citation relations between papers. The second level horizontal network

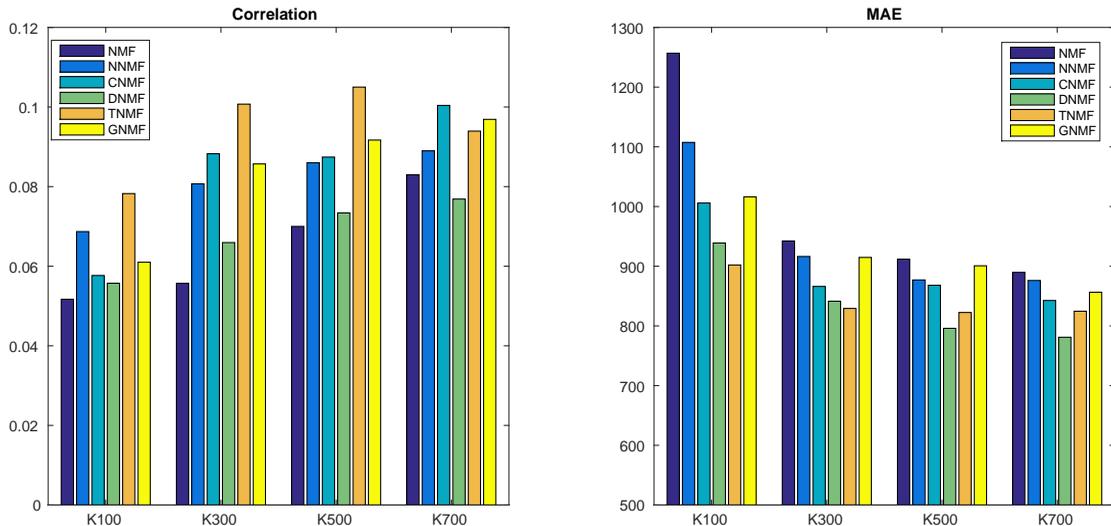


Fig. 7: Comparison of recommendation on different values of K ($K = 100$, $K = 300$, $K = 500$, and $K = 700$) with $\alpha = 10$.

$H_{3703 \times 3703}$ is the keyword concurrence network, which is formed by the concurrence relations between keywords. The vertical network $V_{3312 \times 3703}$ is the mapping relation between documents and keywords (there will be a link between a keyword and a document if this keyword shows in this document).

It should be noted that we keep only one type of structure for two horizontal networks. For example, CNMF only keeps the community structures of both networks, hence Eq. (23) only adds $\alpha \cdot \frac{1}{2} \|P_X - X\|_F^2$. For brevity, the same coefficient is used for both networks.

Here, we compare the performances between the one-level network $H_{3312 \times 3312}$ and the two-level network $H_{3312 \times 3312}$ and $H_{3703 \times 3703}$. The comparisons on three evaluation metrics are shown in Fig. 8. Except CNMF with $K = 300$, two-level network outweighs the one-level network. It means that incorporating the keyword co-occurrence network can improve the document clustering task with only the document citation network.

4) *Recommendation with Two-level Network*: The dataset used for recommendation with two-level network is *Delicious*. This dataset is collected from the Delicious website³, which records the bookmark options of users on webpages/URLs in this website according to users' interests [62]. We filter this dataset by keeping the URLs that are marked by at least three users, and the number of URL is 5633. The links between URLs are generated by their tags. In the Delicious website, each URL will get tags from users, and these tags will give an hint for the content/semantics of this URL. We use these tags to compute the content similarity between URLs with tag-vector representations through cosine similarity metric. The statistics of the dataset are shown in Table. VIII. In this dataset, vertical network $V_{1867 \times 5633}$ is formed by users and URLs. The horizontal networks are: the user friend network $H_{1867 \times 1867}$ and the

content similarity network $H_{5633 \times 5633}$. A certain number of user-URL pairs are kept to be the test data. The evaluation metrics are in Eq. (45) and Eq. (46). As shown in Fig. 9, the two-level network still outweighs the one-level network.

TABLE VIII: Statistics of *Delicious* dataset

number of users	1867
number of URLs	5633
test user-URL pairs	2000

Despite the good performances on both document clustering and recommendation tasks, we still want to give an advice that it should be very careful to combine the different networks: vertical network and different horizontal networks. For example, suppose there are a user-movie vertical network and a movie network for the recommendation task, and the movie network is formed by the distance between the releasing dates of the movies. We know that the releasing date of a movie will not highly impact on the rating from users. In this situation, the combination of two networks may not improve the rating prediction due to the 'noise' from the movie network. Since the information in the movie network will not help the prediction of the ratings, the constraint from movie network will decrease the learned information from the user-movie vertical network. However, if the formation of the movie network is replaced by another strategy: the movies with similar director and actors tend to have links, this new movie network will help to predict the user ratings on movies different from the old movie network. In this new situation, the combination of the movie network with user-movie network will help to improve the rating prediction. Therefore, the consistence between the different networks is important and should be considered before using the multilevel network model.

³<http://www.delicious.com>

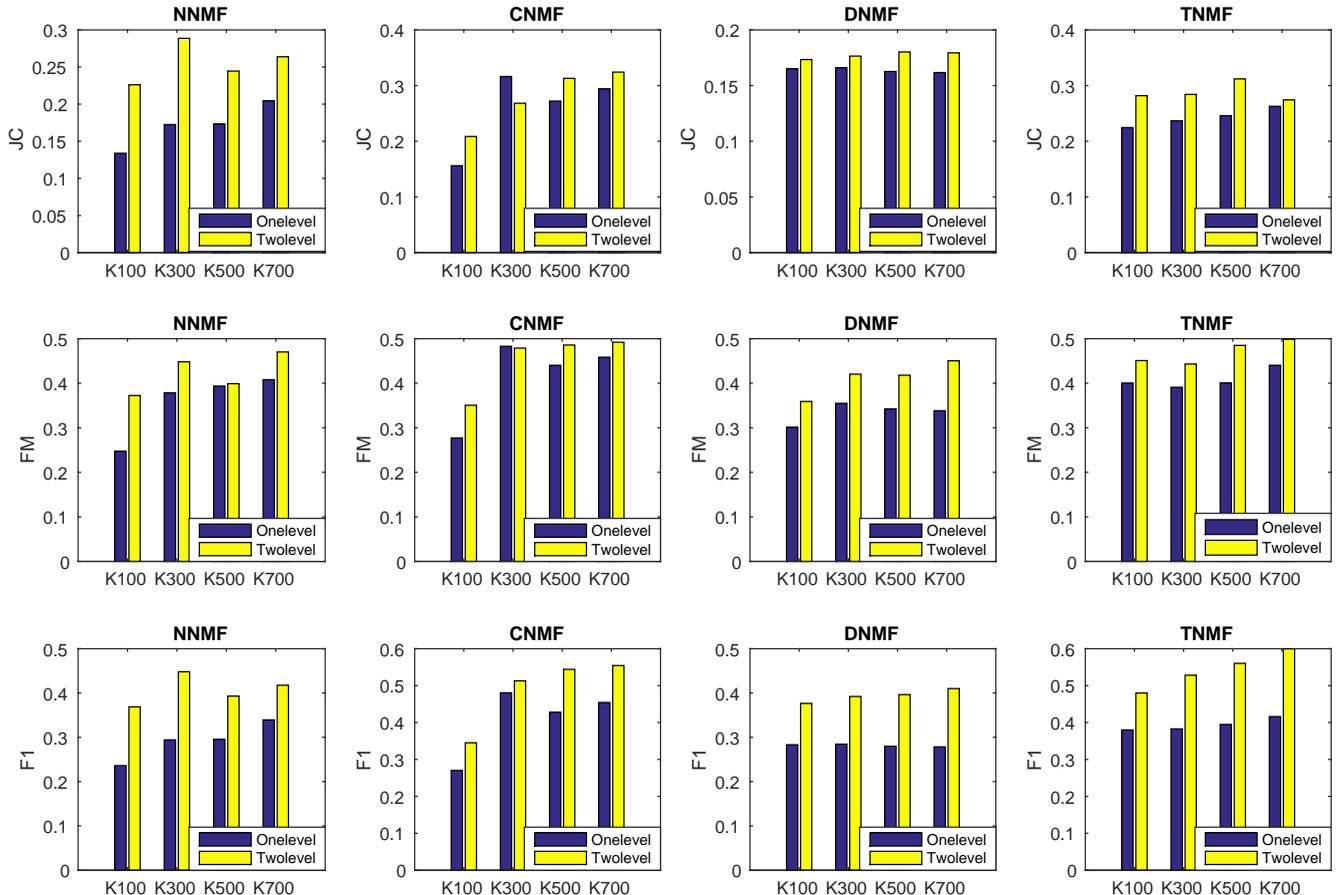


Fig. 8: Comparison of the influence from one-level network (with one horizontal network and one vertical network) and two-level network (with two horizontal networks and one vertical network) on document clustering with $\alpha = 0.1$ and $K = 100$, $K = 300$, $K = 500$ and $K = 700$.

VI. CONCLUSIONS AND FUTURE STUDY

In this paper, we have introduced a general multi-level network model, and proposed four algorithms for multi-level network factorization with four different network structure constraints. The network structural constraints have been incorporated into the cost function of traditional nonnegative matrix factorization. The optimization of the new cost function has constructed a new latent space and projected all nodes in different levels into this new latent space. At the same time, the projected nodes will, as far as possible, retain the original network structure. Four algorithms with their convergence proofs have been carefully designed to find the optimal latent spaces. Finally, experiments on synthetic and real-world data show that our algorithms are able to preserve the network structures, and can be used for recommender systems and document clustering.

There are still some interesting further study points. For example, the dimension number of the discovered latent space needs to be predefined in the present algorithms.

The ability to automatically find an optimized number for the shared latent space will make multi-level network factorization more practical for real-world tasks.

ACKNOWLEDGMENTS

This work was supported in part by the Australian Research Council (ARC) under discovery grant DP140101366 and the China Scholarship Council, and by the National Science Foundation of China under grant nos. 91024012, 91324005.

REFERENCES

- [1] P. Hitzler and K. Janowicz, "Linked data, big data, and the 4th paradigm," *Semantic Web*, vol. 4, no. 3, pp. 233–235, 2013.
- [2] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: a comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, June 2013.
- [3] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [4] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, March 2014.

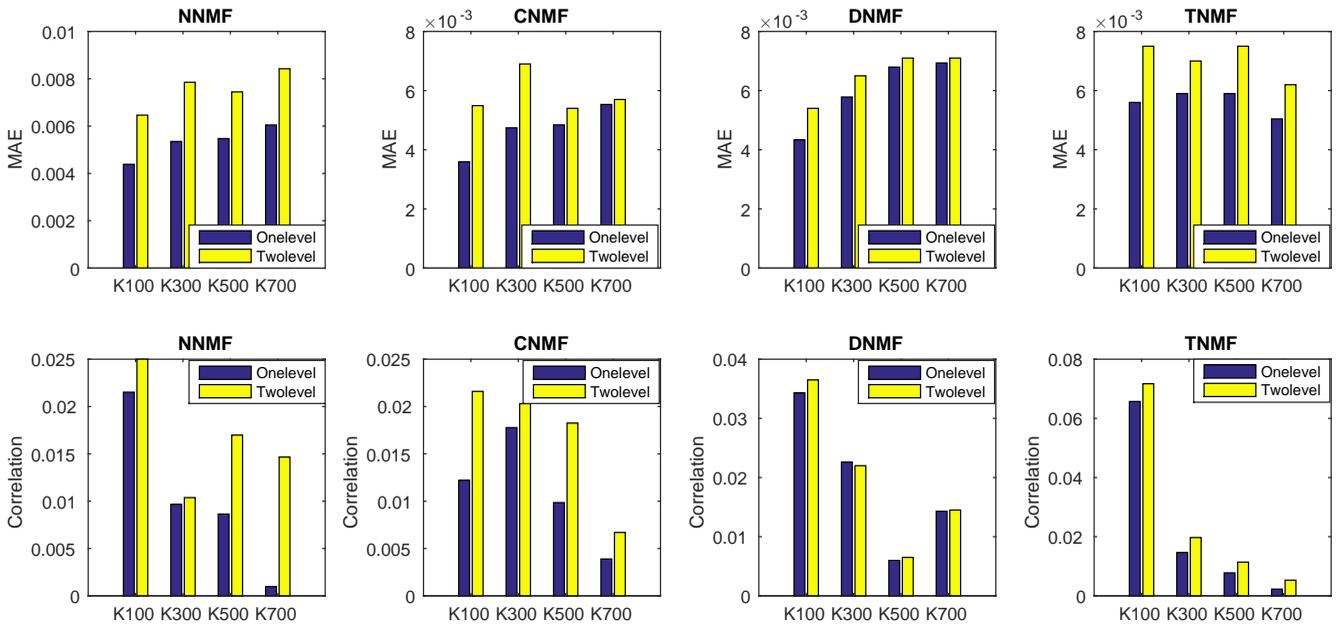


Fig. 9: Comparison of the influence from one-level network (with one horizontal network and one vertical network) and two-level network (with two horizontal networks and one vertical network) on recommendation with $\alpha = 10$ and $K = 100$, $K = 300$, $K = 500$ and $K = 700$.

- [5] O. Zoidi, A. Tefas, and I. Pitas, "Multiplicative update rules for concurrent nonnegative matrix factorization and maximum margin classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 3, pp. 422–434, March 2013.
- [6] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1087–1099, July 2012.
- [7] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki, "Symmetric nonnegative matrix factorization: algorithms and applications to probabilistic clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2117–2131, Dec 2011.
- [8] H. Liu, X. Li, and X. Zheng, "Solving non-negative matrix factorization by alternating least squares with a modified strategy," *Data Mining and Knowledge Discovery*, vol. 26, no. 3, pp. 435–451, 2013.
- [9] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 556–562.
- [10] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [11] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [12] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [13] E. Estrada, *The structure of complex networks: theory and applications*. Oxford University Press, 2011.
- [14] T. Narayanan and S. Subramaniam, "A newtonian framework for community detection in undirected biological networks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 1, pp. 65–73, Feb 2014.
- [15] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1216–1230, July 2012.
- [16] M. W. Hadley, M. F. McGranaghan, A. Willey, C. W. Liew, and E. R. Reynolds, "A new measure based on degree distribution that links information theory and network graph analysis," *Neural Systems & Circuits*, vol. 2, no. 1, 2012.
- [17] D. Centola, "The spread of behavior in an online social network experiment," *Science*, vol. 329, no. 5996, pp. 1194–1197, 2010.
- [18] L. A. Adamic and B. A. Huberman, "Power-law distribution of the world wide web," *Science*, vol. 287, no. 5461, p. 2115, 2000.
- [19] F. Meyer and L. Najman, *Segmentation, minimum spanning tree and hierarchies*. John Wiley & Sons, Inc., 2013, pp. 229–261.
- [20] W. B. March, P. Ram, and A. G. Gray, "Fast euclidean minimum spanning tree: algorithm, analysis, and applications," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD10)*. Washington, DC, USA: ACM, 2010, pp. 603–612.
- [21] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, p. 041022, Dec 2013.
- [22] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gomez-Gardees, M. Romance, I. Sendia-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1 – 122, 2014, the structure and dynamics of multilayer networks.
- [23] C. Li, J. Luo, J. Huang, and J. Fan, "Multi-layer network for influence propagation over microblog," in *Intelligence and Security Informatics*, ser. Lecture Notes in Computer Science, M. Chau, G. Wang, W. Yue, and H. Chen, Eds. Springer Berlin Heidelberg, 2012, vol. 7299, pp. 60–72.
- [24] M. Pasqual and O. de Weck, "Multilayer network model for analysis and management of change propagation," *Research in Engineering Design*, vol. 23, no. 4, pp. 305–328, 2012.
- [25] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, Jan 2010.
- [26] W. Ren, G. Li, D. Tu, and L. Jia, "Nonnegative matrix factorization with regularizations," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 153–164, March 2014.
- [27] Z. Li, X. Wu, and H. Peng, "Nonnegative matrix factorization on orthogonal subspace," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 905 – 911, 2010.
- [28] H. Liu, Z. Wu, X. Li, D. Cai, and T. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1299–1311, July 2012.
- [29] Y. Chen, M. Rege, M. Dong, and J. Hua, "Non-negative matrix

- factorization for semi-supervised data clustering,” *Knowledge and Information Systems*, vol. 17, no. 3, pp. 355–379, 2008.
- [30] S. Huh, M. Das Gupta, and J. Xiao, “Supervised nonnegative matrix factorization,” Patent, July, 2013, uS Patent 8,498,949. [Online]. Available: <http://www.google.com/patents/US8498949>
- [31] Q. Gu, J. Zhou, and C. H. Ding, “Collaborative filtering: weighted nonnegative matrix factorization incorporating user and item graphs,” in *SIAM International Conference on Data Mining (SDM10)*. Columbus, Ohio, USA: SIAM, 2010, pp. 199–210.
- [32] D. Cai, X. He, J. Han, and T. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, Aug 2011.
- [33] R. Zhi, M. Flierl, Q. Ruan, and W. Kleijn, “Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 38–52, Feb 2011.
- [34] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun, “Who should share what?: Item-level social influence prediction for users and posts ranking,” in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’11. New York, NY, USA: ACM, 2011, pp. 185–194.
- [35] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang, “Flexible and robust co-regularized multi-domain graph clustering,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’13. New York, NY, USA: ACM, 2013, pp. 320–328.
- [36] Y.-R. Lin, J. Sun, H. Sundaram, A. Kellihier, P. Castro, and R. Konuru, “Community discovery via metagraph factorization,” *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 3, pp. 17:1–17:44, Aug. 2011.
- [37] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, “Community discovery using nonnegative matrix factorization,” *Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 493–521, May 2011.
- [38] D. M. Blei, “Probabilistic topic models,” *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.
- [39] A. McCallum, X. Wang, and A. Corrada-Emmanuel, “Topic and role discovery in social networks with experiments on enron and academic email,” *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 249–272, Oct. 2007.
- [40] Y. Cha and J. Cho, “Social-network analysis using topic models,” in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’12. New York, NY, USA: ACM, 2012, pp. 565–574.
- [41] V. H. Tuulos and H. Tirri, “Combining topic models and social networks for chat data mining,” in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 206–213.
- [42] E. Wang, J. Silva, R. Willett, and L. Carin, “Dynamic relational topic model for social network analysis with noisy links,” in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2011, pp. 497–500.
- [43] Y. Cha, B. Bi, C.-C. Hsieh, and J. Cho, “Incorporating popularity in topic models for social network analysis,” in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’13. New York, NY, USA: ACM, 2013, pp. 223–232.
- [44] Q. Mei, D. Cai, D. Zhang, and C. Zhai, “Topic modeling with network regularization,” in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW ’08. New York, NY, USA: ACM, 2008, pp. 101–110.
- [45] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson, “Social topic models for community extraction,” in *the 2nd SNA-KDD Workshop*, vol. 8. Citeseer, 2008.
- [46] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” *Journal of Machine Learning Research*, vol. 9, pp. 1981–2014, Jun. 2008.
- [47] J. Chang and D. M. Blei, “Relational topic models for document networks,” in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 81–88.
- [48] Y. Zhu, X. Yan, L. Getoor, and C. Moore, “Scalable text and link analysis with mixed-topic link models,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’13. New York, NY, USA: ACM, 2013, pp. 473–481.
- [49] Y. Sun, J. Han, J. Gao, and Y. Yu, “itopicmodel: Information network-integrated topic modeling,” in *Ninth IEEE International Conference on Data Mining (ICDM ’09)*, Dec 2009, pp. 493–502.
- [50] Y. Liu, A. Niculescu-Mizil, and W. Gryc, “Topic-link lda: joint models of topic and author community,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML ’09. New York, NY, USA: ACM, 2009, pp. 665–672.
- [51] J. Tang, R. Jin, and J. Zhang, “A topic modeling approach and its integration into the random walk framework for academic search,” in *Eighth IEEE International Conference on Data Mining (ICDM ’08)*, Dec 2008, pp. 1055–1060.
- [52] X. Wang and A. McCallum, “Topics over time: A non-markov continuous-time model of topical trends,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’06. New York, NY, USA: ACM, 2006, pp. 424–433.
- [53] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers, “Learning author-topic models from text corpora,” *ACM Transactions on Information Systems*, vol. 28, no. 1, pp. 4:1–4:38, Jan. 2010.
- [54] H. Kim, Y. Sun, J. Hockenmaier, and J. Han, “Etm: Entity topic models for mining documents associated with entities,” in *2012 IEEE 12th International Conference on Data Mining (ICDM)*, Dec 2012, pp. 349–358.
- [55] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, and Y. Yu, “Mining social emotions from affective text,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1658–1670, Sept 2012.
- [56] J. D. McAuliffe and D. M. Blei, “Supervised topic models,” in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 121–128.
- [57] C.-J. Lin, “On the convergence of multiplicative update algorithms for nonnegative matrix factorization,” *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, Nov 2007.
- [58] —, “Projected gradient methods for nonnegative matrix factorization,” *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.
- [59] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, “Complex networks: Structure and dynamics,” *Physics Reports*, vol. 424, no. 45, pp. 175 – 308, 2006.
- [60] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [61] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, “Collective classification in network data,” *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.
- [62] I. Cantador, P. Brusilovsky, and T. Kuflik, “2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011),” in *Proceedings of the 5th ACM conference on Recommender systems*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.



Junyu Xuan is received the bachelor’s degree in 2008 from China University of Geosciences, Beijing. Currently he is working toward the dual-doctoral degree both in Shanghai University and University of Technology, Sydney. His main research interests include Machine Learning, Complex Network and Web Mining.



Jie Lu is a full professor and Head of School of Software at the University of Technology, Sydney. Her research interests lie in the area of decision support systems and uncertain information processing. She has published five research books and 270 papers, won five Australian Research Council discovery grants and 10 other grants. She received a University Research Excellent Medal in 2010. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for International Journal of

Computational Intelligence Systems (Atlantis), editor for book series on Intelligent Information Systems (World Scientific) and guest editor of six special issues for international journals, as well as delivered six keynote speeches at international conferences.



Xiangfeng Luo is a professor in the School of Computers, Shanghai University, China. Currently, he is a visiting professor in Purdue University. He received the master's and PhD degrees from the Hefei University of Technology in 2000 and 2003, respectively. He was a postdoctoral researcher with the China Knowledge Grid Research Group, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), from 2003 to 2005. His main research interests include Web Wisdom, Cognitive Informatics, and Text Understanding. He has authored or co-authored more than 50 publications and his publications have appeared in IEEE Trans. on Automation Science and Engineering, IEEE Trans. on Systems, Man, and Cybernetics-Part C, IEEE Trans. on Learning Technology, Concurrency and Computation: Practice and Experience, and New Generation Computing, etc. He has served as the Guest Editor of ACM Transactions on Intelligent Systems and Technology. Dr. Luo has also served on the committees of a number of conferences/workshops, including Program Co-chair of ICWL 2010 (Shanghai), WISM 2012 (Chengdu), CTUW2011 (Sydney) and more than 40 PC members of conferences and workshops.

He has authored or co-authored more than 50 publications and his publications have appeared in IEEE Trans. on Automation Science and Engineering, IEEE Trans. on Systems, Man, and Cybernetics-Part C, IEEE Trans. on Learning Technology, Concurrency and Computation: Practice and Experience, and New Generation Computing, etc. He has served as the Guest Editor of ACM Transactions on Intelligent Systems and Technology. Dr. Luo has also served on the committees of a number of conferences/workshops, including Program Co-chair of ICWL 2010 (Shanghai), WISM 2012 (Chengdu), CTUW2011 (Sydney) and more than 40 PC members of conferences and workshops.



Guangquan Zhang is an associate professor in Faculty of Engineering and Information Technology at the University of Technology Sydney (UTS), Australia. He has a PhD in Applied Mathematics from Curtin University of Technology, Australia. He was with the Department of Mathematics, Hebei University, China, from 1979 to 1997, as a Lecturer, Associate Professor and Professor. His main research interests lie in the area of multi-objective, bilevel and group decision making, decision support system tools,

fuzzy measure, fuzzy optimization and uncertain information processing. He has published four monographs, four reference books and over 200 papers in refereed journals and conference proceedings and book chapters. He has won four Australian Research Council (ARC) discovery grants and many other research grants.