ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20190927081408
Original scientific paper

Link Prediction based on Deep Latent Feature Model by Fusion of Network Hierarchy Information

Fei CAI, Jie CHEN, Xin ZHANG, Xiaohui MOU, Rongrong ZHU

Abstract: Link prediction aims at predicting latent edges according to the existing network structure information and it has become one of the hot topics in complex networks. Latent feature model that has been used in link prediction directly projects the original network into the latent space. However, traditional latent feature model cannot fully characterize the deep structure information of complex networks. As a result, the prediction ability of the traditional method in sparse networks is limited. Aiming at the above problems, we propose a novel link prediction model based on deep latent feature model by Deep Non-negative Matrix Factorization (DNMF). DNMF method can obtain more comprehensive network structure information through multi-layer factorization. Experiments on ten typical real networks show that the proposed method has performances superior to the state-of-the-art link prediction methods.

Keywords: complex; deep non-negative matrix factorization; latent feature model; link prediction

1 INTRODUCTION

LINK prediction for complex networks is the research hotpot in recent years and it helps us to explore and understand the evolution mechanism of the complex networks. The objective of link prediction is to predict unobserved links from existing parts of the network or forecast future links from current structures of the network.

At present, the existing link prediction methods for complex networks can be divided into two categories: similarity-based methods and probabilistic methods. Similarity-based methods consider that links between two nodes with more similar ones are of higher existing probability, such as Common Neighbors (CN) [1], Adamic-Adar (AA) [2], Cannistraci-Resource-Allocation (CRA) [3-4]. They rely on the network topology and have the shortcoming of limited prediction ability. Probabilistic methods [5] assume that the network has a known structure, calculating connecting probability for edges between unobserved nodes on the basis of model building and parameter estimation, such as Probabilistic Relationship Model (PRM) [6], Hierarchical structure model (HSM) [7], Stochastic block model (SBM) [8]. Although probabilistic methods have many advantages in network analysis, they have the disadvantages of time-consuming [9].

Besides the above link prediction methods, some novel methods have been proposed. According to the consistency of structural feature of a network, Structural Perturbation Method (SPM) based on perturbed eigenvectors was proposed and was applied to solve link prediction problem [10]. Low-Rank (LR) method based on robust principal component analysis and sparse property of the adjacency matrix of network was proposed to predict the missing edges in a network [11]. Methods are based on non-negative matrix factorization by kernel function including Lineal Kernel (LK) and Covariance Kernel (CK) for network reconstruction and link prediction [12].

Relationship between the nodes in complex networks not only depends on network topological information, but also depends on the latent properties and features of network nodes which cannot be observed directly from networks. Therefore, latent feature model is widely used to predict the potential connections in network analysis and link prediction [13-16]. The latent feature model is used to express network nodes by direct projection of network nodes into latent space. The key idea of latent features model is to map the features of the original problem into the latent feature space with less dimension.

If we constrain the elements in two matrices to be non-negative, we can obtain the corresponding solution by non-negative matrix [17-20]. The basic idea of the Non-negative Matrix Factorization (NMF) method is to decompose a non-negative matrix into two low-rank non-negative matrixes. The matrix factorization method cannot only extract the latent features, but also itself is a method of reducing dimension [14, 21]. For example, Shin et al. [22] proposed a multi-scale link prediction method based on the clustering method of low order approximation. The latent feature model based on non-negative matrix decomposition is widely used in link prediction. The results show the latent feature model based on non-negative matrix decomposition can find potential structure of network relations between entities, has strong explanatory of network information, and can automatically learn from latent features and has good adaptability and extensibility. Although different NMF-based methods have good performance on some networks, they still cannot fully characterize the deep structure information complex networks.

Most big real-world networks are very sparse, the average degree of the network is much smaller than the number of nodes, and the number of observed edges is much smaller than the maximum possible edges in the network. Due to the limited availability of information and the network sparsity, it is very difficult to get good performances using the traditional link prediction methods. Therefore, this paper proposes a novel link prediction model based on Deep Non-negative Matrix Factorization (DNMF) by fusion of network hierarchy information. Firstly, a hierarchical network structure learning model is formed by decomposing the coefficient matrix many times. Then, unsupervised learning tactic, which has been used successfully on autoencoder networks [23], is adopted as the training method, multiple-layer factorization is as pre-decomposition results and then the basis matrices and the coefficients matrix can be adjusted as fine-tuning result. Finally, the similarity matrix is calculated according to the fine-tuned basis matrix and fine-tuned coefficients matrix. This model can guarantee the expression of hierarchy structure information on real networks, at the same time, and can get much richer and more comprehensive potential feature information, and improve the prediction accuracy of link prediction.

In conclusion, the contribution of this paper is:

1) On the basis of non-negative matrix factorization, multilayer factorization is applied to latent feature model, the hierarchical structure information of a network can be learned by multi-layer factorization.
2) We learn from the unsupervised learning strategy of the autoencoder network and adopt the two-stage including pre- training and fine-tuning for link prediction.
3) Similarity matrix can be obtained by a group of basis matrices and the coefficients matrix.

The remainder of this paper is organized as follows. First, we introduce problem statement and present proposed methodology in Section II. We then give the experimental evaluation metrics, experiment data and experimental results in Section III. Finally, we conclude the paper in Section IV.

2 PROBLEM STATEMENT AND PROPOSED METHODOLOGY

2.1 Problem Definition

As we know, a network consists of nodes and edges. Given an undirected network $G = (V, E)$, where V and E represent the set of nodes and set of edges respectively, $N = |V|$ and $M = |E|$ represent the number of nodes and edges of the network respectively. The network can be expressed by an adjacency matrix $\boldsymbol{A}$, where the size of $\boldsymbol{A}$ is $N \times N$, where $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji} = 1$ if there is a connection between node i and node j otherwise $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji} = 0$.

In order to verify the performance of the proposed method for link prediction, the observed links are randomly divided into a training set E_{train} and a testing set E_{test}, where $E_{\text{train}} \cup E_{\text{test}} = E$ and $E_{\text{train}} \cap E_{\text{test}} = \emptyset$. Here, training set E_{train} is used to establish prediction model while testing set E_{test} is only used to verify the accuracy for link prediction in complex networks. A_{test} and A_{train} represent the adjacency matrix of the training set and the testing set respectively, all elements of them are 1 or 0, where $\boldsymbol{A}_{\text{train}} + \boldsymbol{A}_{\text{test}} = \boldsymbol{A}$. Let $L = |\,E_{\text{test}}\,|$ represent the number of edges in the test set. So, the number of training set edges is $|E_{\text{train}}| = M - L$. Except for the training set, the number of all possible edges in the network are regarded as the candidate set, which is $\left|\bar{E}\right| = \frac{N(N-1)}{2} - (M - L)$. Then, the prediction model is studied from the training set E_{train} and the probability value of each possible edge is calculated, and the results of the test set E_{test} are verified according to different evaluation metrics.

2.2 Link Prediction Based on Deep Latent Feature Model

2.2.1 Non-Negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a matrix factorization algorithm which is a recent method for making the latent structure in data more explicit and reducing its dimensionality [24].

Given adjacency matrix of a network $\boldsymbol{A} \in \boldsymbol{R}^{M\times N}$ which can be well approximated by two non-negative matrices $\boldsymbol{W} \in \boldsymbol{R}^{M\times k}$ and $\boldsymbol{H} \in \boldsymbol{R}^{k\times N}$ such that:

$$\boldsymbol{A} \approx \boldsymbol{WH} \tag{1}$$

In order to quantify the quality of the approximation, the cost function with the square of the Euclidean distance can be written as follows:

$$O = \boldsymbol{A} - \boldsymbol{WH}_F^2 = \sum_{ij}\left(\boldsymbol{A}_{ij} - \sum_{k=1}^{k} w_{ik}\cdot h_{kj}\right)^2 \tag{2}$$

where, $\boldsymbol{W}$ and $\boldsymbol{H}$ represent the basis matrix and the coefficients matrix respectively. According to iterative update algorithm [24], the iterative algorithm minimizing the objective function O in Eq. (2) is as follows:

$$\boldsymbol{W}_{ik} \leftarrow \boldsymbol{W}_{ik}\frac{\left(\boldsymbol{XH}^{\mathrm{T}}\right)_{ik}}{\left(\boldsymbol{WHH}^{\mathrm{T}}\right)_{ik}} \tag{3}$$

$$\boldsymbol{H}_{kj} \leftarrow \boldsymbol{H}_{kj}\frac{\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)_{kj}}{\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{WH}\right)_{kj}} \tag{4}$$

2.2.2 Deep Non-Negative Matrix Factorization

Based on the decomposition of non-negative matrix, this paper proposes an algorithm named Deep Non-negative Matrix Factorization. Through the multiple factorization of the coefficients matrix, the multi-layer structure information of the network is fused, and its factorization schematic diagram is shown in Fig. 1.

The Deep NMF forms a multi-level network structure learning model through the multiple factorization of the coefficients matrix $\boldsymbol{H}$. The factorization steps of $\boldsymbol{H}$ are as follows:

Step 1: we first factorize the network adjacency matrix $\boldsymbol{A} \approx \boldsymbol{W}_1\boldsymbol{H}_1$, where $\boldsymbol{W}_1 \in \boldsymbol{R}^{N\times k1}$ and $\boldsymbol{H}_1 \in \boldsymbol{R}^{k1\times N}$; $k_1 = \left\lceil \frac{N}{4} \right\rceil$; $\left\lceil \frac{N}{4} \right\rceil$ represents the smallest integer not less than $\left\lceil \frac{N}{4} \right\rceil$; R represents the real number field.

Step 2: Following Step1, the coefficients matrix H_1 can be factorized to $\boldsymbol{H}_1 \approx \boldsymbol{W}_2\boldsymbol{H}_2$, where $\boldsymbol{W}_2 \in \boldsymbol{R}^{k1\times k2}$ and $\boldsymbol{H}_2 \in \boldsymbol{R}^{k2\times N}$; $k_2 = \left\lceil \frac{k_1}{4} \right\rceil$;

Step 3: By analogy, after m times of factorization, the network adjacency matrix $\boldsymbol{A} \approx \boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3, \ldots, \boldsymbol{W}_m\boldsymbol{H}_m$, where $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_m, \boldsymbol{H}_m$ are non-negative. $\boldsymbol{W}_m \in \boldsymbol{R}^{km-1\times km}$, $\boldsymbol{H}_m \in \boldsymbol{R}^{km\times N}$; $k_m = \left\lceil \frac{k_{m-1}}{4} \right\rceil$.

After m times of factorization on the coefficients matrix $\boldsymbol{H}$, it can be expressed by $m + 1$ factors multiplied, including m basis matrices and a coefficients matrix. Each additional basis matrix which is added is equivalent to adding an additional layer of abstraction to automatically learn the network hierarchy information and explore the latent features more accurately and comprehensively. The loss function of Deep-NMF can be expressed as:

$$\begin{aligned}
C_{\text{Deep\_}NMF} &= \left\| \boldsymbol{A}-\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m \right\|_F^2 \\
&= \mathrm{Tr}((\boldsymbol{A}-\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)^{\mathrm{T}} \\
&\quad (\boldsymbol{A}-\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)) \\
&= \mathrm{Tr}((\boldsymbol{A}^{\mathrm{T}}-\boldsymbol{W}_1{}^{\mathrm{T}}\boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{W}_3{}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_{m-1}{}^{\mathrm{T}}\boldsymbol{W}_m{}^{\mathrm{T}}) \\
&\quad (\boldsymbol{A}-\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)) \\
&= \mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m - \\
&\quad -\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{A}+ \\
&\quad +\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_1{}^{\mathrm{T}}\boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{W}_3{}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_{m-1}{}^{\mathrm{T}}\boldsymbol{W}_m{}^{\mathrm{T}}\boldsymbol{W}_1 \\
&\quad \boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m) \\
&= \mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}-2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m + \\
&\quad +(\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m{}^{\mathrm{T}}\boldsymbol{H}_m)^{\mathrm{T}} \\
&\quad \boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m) \\
&= \mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}-2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m + \\
&\quad +\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_1{}^{\mathrm{T}}\boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{W}_3{}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_{m-1}{}^{\mathrm{T}}\boldsymbol{W}_m{}^{\mathrm{T}}\boldsymbol{W}_1 \\
&\quad \boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)
\end{aligned} \tag{5}$$

where, $\boldsymbol{W} \geq 0$, $\boldsymbol{H} \geq 0$.

In Eq. (5), let $\Lambda_l = [\lambda_{ik}]_l$ and $M = [u_{jk}]$ be the Lagrange multiplier for constraint $\boldsymbol{W} \geq 0$, and $\boldsymbol{H} \geq 0$ respectively, where l =1, 2, …, m, $\lambda_{ik} = \geq 0$, $u_{jk} \geq 0$. The Lagrange function can be expressed as follows:

$$\begin{aligned}
C_{\text{Deep\_}NMF} &= \mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}-2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m + \\
&\quad +\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_1{}^{\mathrm{T}}\boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{W}_3{}^{\mathrm{T}},\ldots,\ \boldsymbol{W}_{m-1}{}^{\mathrm{T}}\boldsymbol{W}_m{}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots, \\
&\quad \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)+\sum_{l=1}^{l=m}\ \sum_{i=1}^{i=K_{m-1}}\ \sum_{k=1}^{k=K_m} \lambda_{ik}^{(l)} w_{ik}^{(l)} + \\
&\quad +\sum_{j=1}^{j=N}\ \sum_{k=1}^{k=K_m} u_{jk}h_{jk} \\
&= \mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}-2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m + \\
&\quad +\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_1{}^{\mathrm{T}}\boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{W}_3{}^{\mathrm{T}},\ldots,\ \boldsymbol{W}_{m-1}{}^{\mathrm{T}}\boldsymbol{W}_m{}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots, \\
&\quad \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m)+\sum_{l=1}^{l=m}\mathrm{Tr}(\Lambda_l\boldsymbol{W}_l)+\mathrm{Tr}(M\boldsymbol{H}_m^{\mathrm{T}})
\end{aligned} \tag{6}$$

The optimization objective function based on non negative matrix factorization is a non-convex optimization problem, and its prediction results depend on the initial value of the basis matrix $\boldsymbol{W}$ and coefficients matrix $\boldsymbol{H}$. Traditional non-negative matrix factorization methods tend to be random initializations $\boldsymbol{W}$ and $\boldsymbol{H}$, but it is easy to get into the local optimal solution, which may also result in under fitting phenomenon. In order to improve the generalization ability of the proposed method, we draw from the unsupervised learning strategy of auto encoder network [25], so the two-stage including pre-training and fine-tuning is adopted for link prediction.

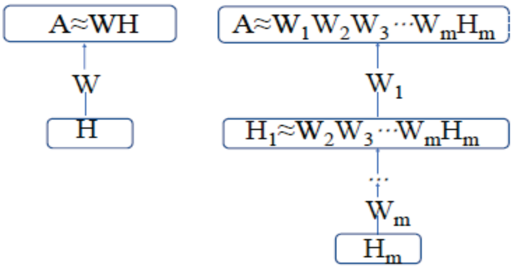


Figure 1 Comparison schematic diagram of NMF and Deep NMF

(1) Pre-training stage

Step 1: we first decompose the network adjacency matrix $\boldsymbol{A} \approx \boldsymbol{W}_1\boldsymbol{H}_1$, where $\boldsymbol{W}_1 \in \boldsymbol{R}^{N\times k1}$ and $\boldsymbol{H}_1 \in \boldsymbol{R}^{k1\times N}$;

Step 2: Following Step 1, the coefficients matrix $\boldsymbol{H}_1$ can be decomposed to $\boldsymbol{H}_1 \approx \boldsymbol{W}_2\boldsymbol{H}_2$, where $\boldsymbol{W}_2 \in \boldsymbol{R}^{k1\times k2}$ and $\boldsymbol{H}_2 \in \boldsymbol{R}^{k2\times N}$;

Step 3: Continuing to do so until all of the layers have been pre-trained, the network adjacency matrix $\boldsymbol{A} \approx \boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3$, …, $\boldsymbol{W}_m\boldsymbol{H}_m$, where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, …, $\boldsymbol{W}_m$, $\boldsymbol{H}_m$ are non-negative.

(2) Fine-tuning stage

In Eq. (6), the partial derivatives of $C_{\text{Deep\_}NMF}$ with respect to $\boldsymbol{W}_m$ and $\boldsymbol{H}_m$ is as follows:

$$\begin{aligned}
\frac{C_{\text{Deep\_}NMF}}{\partial \boldsymbol{W}_m} &= -2(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1})^{\mathrm{T}}\boldsymbol{H}_m^{\mathrm{T}} + \\
&\quad +(\boldsymbol{W}_{m-1}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1})^{\mathrm{T}} \\
&\quad \boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}+\boldsymbol{W}_{m-1}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3, \\
&\quad \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}+\Lambda_m \\
&= -2(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1})^{\mathrm{T}}\boldsymbol{H}_m^{\mathrm{T}} + \\
&\quad +\boldsymbol{W}_{m-1}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3, \\
&\quad \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}+\boldsymbol{W}_{m-1}^{\mathrm{T}},\ \ldots, \\
&\quad \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}+\Lambda_m \\
&= -2(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1})^{\mathrm{T}}\boldsymbol{H}_m^{\mathrm{T}} + \\
&\quad +2\boldsymbol{W}_{m-1}^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots, \\
&\quad \boldsymbol{W}_{m-1}\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}+\Lambda_m
\end{aligned} \tag{7}$$

$$\begin{aligned}
\frac{C_{\text{Deep\_}NMF}}{\partial \boldsymbol{H}_m} &= -2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_m+\boldsymbol{H}_m^{\mathrm{T}}\boldsymbol{W}_m^{\mathrm{T}},\ \ldots, \\
&\quad \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_m\boldsymbol{H}_m+M \\
&= -2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_m+\boldsymbol{W}_m^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}} \\
&\quad \boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_m\boldsymbol{H}_m+(\boldsymbol{W}_m^{\mathrm{T}},\ \ldots, \\
&\quad \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{H}_m+M \\
&= -2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3\ldots\boldsymbol{W}_m+2\boldsymbol{W}_m^{\mathrm{T}},\ \ldots,\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}} \\
&\quad \boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_m\boldsymbol{H}_m+M
\end{aligned} \tag{8}$$

Let $\boldsymbol{\Psi} = \boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}$,

$\boldsymbol{\Psi}^{\mathrm{T}} = \left(\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3,\ \ldots,\ \boldsymbol{W}_{m-1}\right)^{\mathrm{T}}$,

so, the Eq. (7) and Eq. (8) can be rewritten as follows:

$$\frac{C_{\text{Deep\_}NMF}}{\partial \boldsymbol{W}_m} = -2\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{H}_m^{\mathrm{T}} + 2\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}} + \Lambda_m \tag{9}$$

$$\begin{aligned}\frac{C_{\text{Deep\_}NMF}}{\partial \boldsymbol{H}_m} &= -2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3, \ldots, \boldsymbol{W}_m + 2\boldsymbol{W}_m^{\mathrm{T}}, \ldots, \\ &\quad \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{W}_2\boldsymbol{W}_3, \ldots, \boldsymbol{W}_m\boldsymbol{H}_m + M \\ &= -2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m + 2\boldsymbol{W}_m^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m\boldsymbol{H}_m + M\end{aligned} \tag{10}$$

Using the KKT conditions $\lambda_{ik} w_{ik}^{(m)} = 0$ and $u_{jk}h_{jk} = 0$, so we get the following equation:

$$\left(\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{H}_m^{\mathrm{T}}\right)_{ik} w_{ik}^{(m)} + \left(\boldsymbol{\Psi}^{\mathrm{T}}\Psi\boldsymbol{W}_m\boldsymbol{H}_m\boldsymbol{H}_m^{\mathrm{T}}\right)_{ik} w_{ik}^{(m)} = 0 \tag{11}$$

$$\left(\boldsymbol{A}^{\mathrm{T}}\Psi\boldsymbol{W}_m\right)_{jk} h_{jk} + \left(\boldsymbol{W}_m^{\mathrm{T}}\Psi^{\mathrm{T}}\Psi\boldsymbol{W}_m\boldsymbol{H}_m\right)_{jk} h_{jk} = 0 \tag{12}$$

According to the literature [18], we can get the following multiplication updating rules for $\boldsymbol{W}_m$ and $\boldsymbol{H}_m$:

$$(\boldsymbol{W}_m)_{ik} \leftarrow (\boldsymbol{W}_m)_{ik} \cdot \frac{\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{A}\tilde{\boldsymbol{H}}_m^{\mathrm{T}}}{\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m\tilde{\boldsymbol{H}}_m\boldsymbol{H}_m^{\mathrm{T}}} \tag{13}$$

$$(\boldsymbol{H}_m)_{jk} \leftarrow (\boldsymbol{H}_m)_{jk} \cdot \frac{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m}{\boldsymbol{W}_m^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{W}_m\boldsymbol{H}_m} \tag{14}$$

(3) Predicting links using Deep Non-negative matrix Factorization

Inputting a network data, the proposed algorithm for link prediction has three steps. Firstly, we get the number of latent features of the original network adjacency matrix Aby Colibri method. Secondly, DNMF is used to find two non-negative matrix factors ***W*** and ***H***. Thirdly, the network can be reconstructed by ***W*** and ***H*** to make the final prediction (Algorithm 1).

Algorithm 1: The framework for the proposed algorithm with network hierarchy information

```
input: Given the network adjacency matrix $A$, the proportion of training set $f$ and layer number $m$.
output: The similarity matrix of the network $A^*$.
1: procedure
2: divide A into $A^{train}$ and $A^{test}$ with parameter f
3:
for $r=1:m$ do
    get the number of latent features $kr$
    if $r=1$
    then $W_1, H_1 \leftarrow NMF(A)$
    else
    $W_r, H_r \leftarrow NMF(H_{r-1})$
end
repeat
    for $r=1:m$ do
        $\Psi \leftarrow W_1W_2W_3\cdots W_{r-1}$
        $W_r \leftarrow W_r \odot \frac{\Psi^{\mathrm{T}}AH_r^{\mathrm{T}}}{\Psi^{\mathrm{T}}\Psi W_rH_rH_r^{\mathrm{T}}}$
        $H_r \leftarrow H_r \odot \frac{A^{\mathrm{T}}\Psi W_r}{W_r^{\mathrm{T}}\Psi^{\mathrm{T}}\Psi W_rH_r}$
    end
until the value of less function is less than the tolerance
according to $A^* = W_1W_2, \ldots, W_mH_m$, calculate the similarity matrix $A^*$
end procedure
```

3 EXPERIMENT AND COMPARISON

3.1 Evaluation Metrics

In this work, in order to verify the performance of the proposed method, three evaluation metrics are used to compare the performance of the proposed method and the baseline methods. Three evaluation metrics which include AUC, Precision and Prediction-Power (PP) are defined as follows:

(1) AUC [26]: AUC is general evaluation metrics, it means the area under curve for the receiver operating characteristics (ROC) analysis. Given the top L links as predicted links, a ROC curve is obtained by plotting true positive rates versus false positive rates for varying L values. Thus AUC can be interpreted as the probability that a randomly chosen missing link has a higher score than a randomly chosen non-existent link in the rank of all non-observed links. In algorithmic implementation, if among *n* times of independent comparisons, there are *n*' times in which the score of the missing link is higher than that of the non-existent link and nJJ times in which the two have the same score, then AUC can be defined as follows:

$$AUC = \frac{n' + 0.5n''}{n} \tag{15}$$

If all the scores are generated from an independent and identical distribution, AUC will be approximately 0.5.

(2) Precision [27]: Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the number of items selected. Precision can be defined as:

$$Precision = \frac{L_r}{L} \tag{16}$$

where, L represents the size of the predicted links, L_r represents the size of correctly predicted links. Clearly, higher precision can denote higher accuracy.

(3) Prediction-Power (*PP*) [3]: In order to characterize the difference between the proposed prediction algorithm and random prediction, literature [3] puts forward the prediction ability evaluation which is used to evaluate the overall predictive effect for link prediction methods. The higher Prediction value can denote higher Prediction effect. Predictive Power (*PP*) is defined as:

$$PP = 10 \times \log_{10} \frac{Precision}{Precision_{\text{Random}}} \tag{17}$$

where, $Precision_{\mathrm{Randon}}$ is the precision value of the random prediction, it means that predicting edges are ranked randomly. The average random prediction accuracy is approximately equal to $\frac{L}{N(N-1)/2-(M-L)}$, where N and M represent the number of nodes and the number of edges respectively in the network.

3.2 Baseline Methods for Comparison

In order to verify the performance of the proposed method DNMF, we compare our proposed method with fourteen state-of-the-art link prediction methods for performance comparison.

The fourteen methods can be classified into local methods and global methods. Local methods include Common Neighbors (CN) [1], Adamic-Adar (AA) [2], Cannistraci-Resource-Allocation (CRA) [3-4], Resource Allocation (RA) [28], Local Path (LP) [28], Preferential Attachment (PA) [29], Jaccard [30]. Global methods include NMF, Katz [31], Average Commute Time (ACT) [32], Structural Perturbation Method (SPM) [10], Low Rank (LR) [11], LK [12] and CK [12]. The detailed description of these Baseline methods is shown in Tab. 1.

Table 1 Fourteen typical Baseline methods

Similarity	Scores	Detailed Description
Katz	$S_{xy}^{Katz}=\left(\left(I-\alpha A^{-1}\right)-I\right)_{xy}$	α is adjustable parameter, I represents diagonal matrix
ACT	$S_{xy}^{ACT}=\frac{1}{l_{xx}^{+}+l_{yy}^{+}-2l_{xy}^{+}}$	l_{xy}^{+} represents the element in the position of the x row y column in $\boldsymbol{L}^{+}$ matrix, $\boldsymbol{L}^{+}$ represents pseudo inverse of the network Laplasse matrix
CN	$S_{xy}^{CN}=\lvert\Gamma(x)\cap\Gamma(y)\rvert$	$\Gamma(x)$ and $\Gamma(y)$ represent the set of neighbors of x and y
AA	$S_{xy}^{AA}=\sum_{Z\in\Gamma(x)\cap\Gamma(y)}\frac{1}{\log k_z}$	k_z represents degree of node
CRA	$S_{xy}^{CRA}=\sum_{Z\in\Gamma(x)\cap\Gamma(y)}\frac{d_z}{k_z}$	d_z represents the subset of neighbors of node z that are also common neighbors of nodes x and y
RA	$S_{xy}^{RA}=\sum_{Z\in\Gamma(x)\cap\Gamma(y)}\frac{1}{k_z}$	*k_z represents degree of node*
LP	$S_{xy}^{LP}=(A^2+\alpha A^3)_{xy}$	α is adjustable parameter, represents adjacency matrix
PA	$S_{xy}^{PA}=k_x k_y$	k_x represents degree of node x
Jaccard	$S_{xy}^{Jaccard}=\frac{\lvert\Gamma(x)\cap\Gamma(y)\rvert}{\lvert\Gamma(x)\cup\Gamma(y)\rvert}$	$\Gamma(x)$ and $\Gamma(y)$ represent the set of neighbors of x and y
NMF	$O=\min A-WH_F^2$	$\boldsymbol{A}$ is adjacent matrix
SPM	$S^{SPM}=\sum_{k=1}^{N}(\lambda_k+\Delta\lambda_k)x_k x_k^{\mathrm{T}}$	λ_k is the eigenvalue of the observed matrix, x_k is the corresponding orthogonal normalized eigenvector, $\Delta\lambda_k$ is the eigenvalue of a perturbation set respectively, size of $\Delta\lambda_k$ is dependent on perturbation ratio η
LR	$\min_{X^*,E} rank\left(X^*\right)+\gamma E_0$	$X^*=\boldsymbol{A}-\boldsymbol{E}$, rank($\boldsymbol{X}^*$) denotes the rank of matrix $\boldsymbol{X}^*$, the operator $\Vert.\Vert$is the l_0 - norm(i.e., the number of nonzero entries of a matrix), and γ is the parameter balancing these two terms.
LK	$k(x,y)$and $\min K-WH_F^2$	$k(x, y)$ is Linear Kernel
CK	$k(x,x)$and $\min K-WH_F^2$	$k(x, x)$ is Covariance Kernel

3.3 Experiment Data

In order to verify the performance of the proposed method, we consider the following 10 real world networks: Jazz, a network of jazz bands [33]; NS, a network of co-author-ship between scientists working on network theory [34]; PB, a political blogs network of hyper-links between weblogs on politics [35]; Power, the network representing the topology of the power grid of US [36]; Router, a network of internet route [37]; SmaGri, a network of citation on network theory and experiment [38]; USAir, a network of USA airlines [38]; Yeast, a network of protein-protein interaction on yeast [39]; Karate, a social network of individuals of a karate club [40]; School, a friendship network in a high school [41].

Table 2 The topological features of the ten real-world networks

Network−index	$\lvert V\rvert$	$\lvert E\rvert$	*LD*	*<K>*	*<d>*	*C*	*CC*	*r*	*LPC-corr*	*H*
Jazz	198	2742	0.141	27.697	2.235	0.002	0.618	0.020	0.949	1.395
NS	379	914	0.013	4.823	6.402	0.000	0.798	−0.082	0.922	1.660
PB	1222	16714	0.022	27.355	2.738	0.000	0.320	−0.221	0.929	2.971
Power	4941	6594	0.001	2.669	18.989	0.000	0.107	0.003	0.846	1.450
Router	5022	6258	0.000	2.492	6.449	0.000	0.033	−0.138	0.807	5.503
SmaGri	1024	4916	0.009	9.602	2.981	0.000	0.307	−0.193	0.946	3.947
USAir	332	2126	0.039	12.807	2.738	0.001	0.749	−0.208	0.980	3.460
Yeast	2361	6646	0.002	5.630	5.096	0.000	0.388	0.454	0.969	3.476
karate	34	78	0.139	4.588	2.408	0.013	0.571	−0.476	0.756	1.693
School	69	220	0.094	6.377	2.965	0.005	0.461	0.014	0.901	1.198

Tab. 2 provides topological features of the ten real-world networks. Where, *V* and *E* represent the set of nodes and set of edges respectively. *LD* and (*K*) are link density and the average degree; *APL* and *C* are the shortest distance and the average closeness for all the pair nodes of the network, *CC* and *r* are clustering coefficient and the degree-degree correlation coefficient respectively. *LCP-corr* represents the correlation coefficient between *LCP* (Local Community Paradigm, *LCP*) and *CN* [3]. *H* denotes the degree heterogeneity.

3.4 Experimental Results

In order to test the performance of the proposed method, we compare the proposed method with fourteen well-known methods on 10 real networks. The observed links are randomly divided into a training set and a test set. Here, training set is used to establish prediction model while test set is only used to verify the accuracy for link prediction in complex networks. As represented in Tabs. 3 to 5, the performance on the ten real world networks is shown based on AUC, Precision and PP, respectively. The largest value in each column is represented in bold face.

We compared our methods (DNMF) with other methods on the 10 network data sets and the AUC values are returned with the average over 100 runs. In our experiments, set $\alpha = 0.0001$ for *LP*, parameter $\alpha = 0.01$ for Katz, $m = 2$ for DNMF, $\eta = 0.1$ for SPM, $\gamma = 0.15$ for LR. For each data set, the observed links are randomly divided into training set (90%) and test set (10%).

As shown in Tab. 3, DNMF is better than traditional NMF. Furthermore, DNMF has the best AUC values on several real networks, including PB, SmaGri, Yeast and School. AUC values of our proposed method are very close to the highest ones on the other networks.

As shown in Tab. 4, DNMF has better precision values than traditional NMF as a whole. DNMF has the best precision values on several networks including PB, Power, Router, USAir and Yeast. On other networks, such as Jazz, Karate, DNMF has the second best precision values. Under precision metric, the traditional methods do not perform well on sparse networks, such as Router, PB, Yeast, while DNMF performs much better. This indicates that DNMF is superior to the traditional NMF and other classical methods, especially on sparse networks, such as router, PB, Router, Yeast, Power, etc.

Tab. 5 shows a comparison of the prediction accuracy measured by PP on ten typical real-world networks. The mean value of PP of each method across all the networks is shown at the last column and it is an indicator of average performance. Different methods are presented in increasing order of mean PP. As seen from Tab. 5, DNMF has the best overall performance and SPM has the second best overall performance. In overall performance aspect, DNMF is better than CK and LK, which indicates DNMF can extract more useful and richer organization of features hidden in the original network.

To accurately test our proposed method, we analyze the experimental results on the six networks with different fraction of training set from 0.3 to 0.9. As shown in Figs. 2 to 4, we show the results of six networks based on AUC, Precision and PP, respectively. The results are returned with the average of over 100 runs. The six networks are Yeast, Jazz, PB, SmaGri, USAir and School. The red line with asterisk represents the performance of the proposed DNMF.

Table 3 Comparison of link prediction accuracy measured by AUC on ten real-world networks

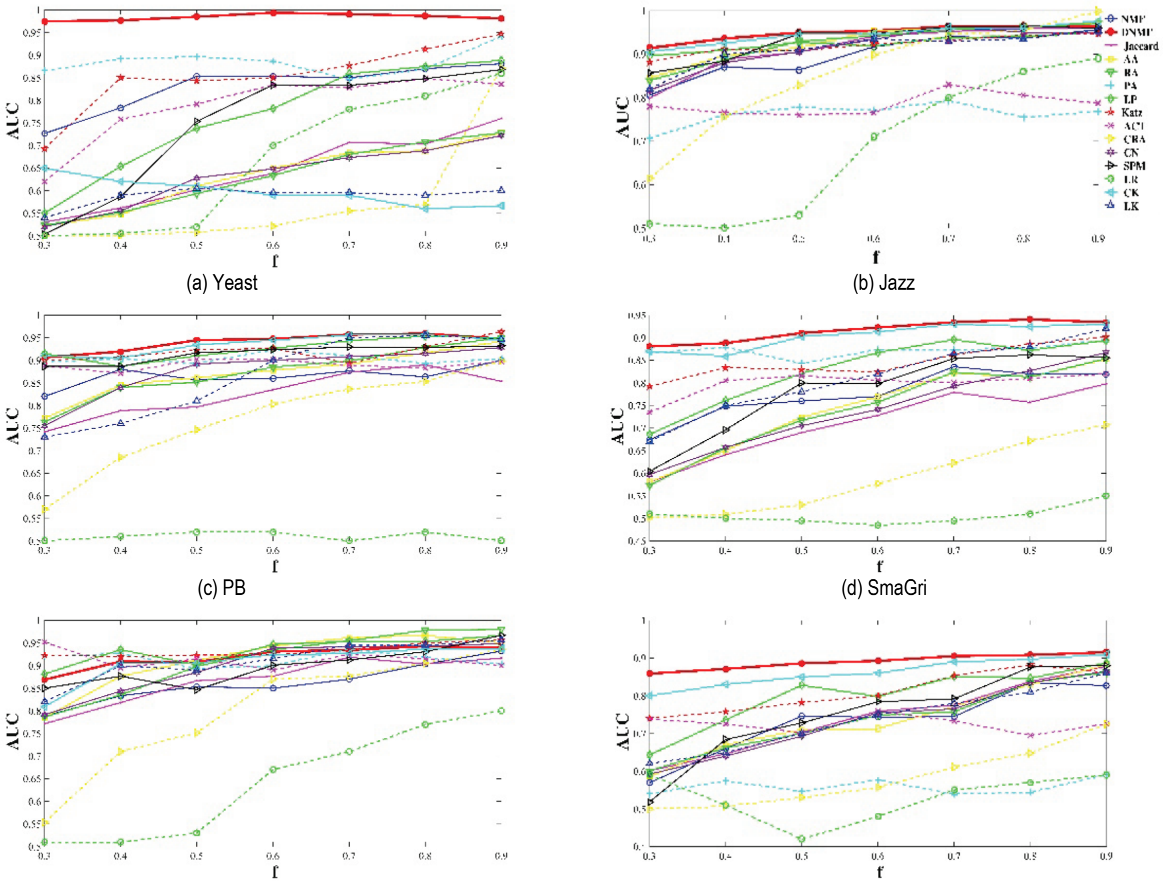
AUC	Jazz	NS	PB	Power	Router	SmaGri	USAir	Yeast	Karate	School
NMF	0.959	0.791	0.910	0.670	0.705	0.819	0.932	0.881	0.723	0.827
DNMF	0.963	0.982	**0.948**	0.785	0.860	**0.935**	0.939	**0.981**	0.731	**0.915**
Katz	0.946	0.986	0.933	**0.964**	**0.977**	0.874	0.952	0.933	0.738	0.867
ACT	0.787	0.934	0.893	0.892	0.964	0.829	0.901	0.900	0.611	0.702
CN	0.940	0.983	0.923	0.625	0.652	0.842	0.954	0.915	0.674	0.847
AA	0.967	0.983	0.927	0.625	0.652	0.790	0.966	0.916	0.711	0.886
CRA	**0.982**	0.827	0.899	0.513	0.964	0.707	0.935	0.872	0.530	0.725
RA	0.973	**0.986**	0.928	0.625	0.652	0.855	**0.972**	0.918	0.719	0.879
LP	0.945	0.952	0.936	0.698	0.944	0.909	0.952	0.970	**0.746**	0.880
PA	0.783	0.912	0.909	0.579	0.955	0.846	0.912	0.864	0.726	0.891
Jaccard	0.961	0.977	0.877	0.625	0.651	0.781	0.915	0.914	0.591	0.859
SPM	0.961	0.921	0.930	0.901	0.930	0.870	0.955	0.875	0.730	0.890
LR	0.895	0.792	0.541	0.515	0.621	0.552	0.810	0.861	0.541	0.590
CK	0.975	0.930	0.945	0.890	0.910	0.930	0.937	0.567	0.705	0.900
LK	0.955	0.920	0.947	0.901	0.925	0.921	0.956	0.600	0.723	0.860

Table 4 Comparison of link prediction accuracy measured by Precision on ten typical real-world networks

Precision	Jazz	NS	PB	Power	Router	SmaGri	USAir	Yeast	Karate	School
NMF	0.548	0.265	0.143	0.022	0.025	0.053	0.320	0.139	0.156	0.172
DNMF	0.600	0.470	**0.240**	**0.059**	**0.205**	0.121	**0.473**	**0.170**	0.190	0.201
Katz	0.449	0.299	0.175	0.058	0.060	0.099	0.365	0.108	0.169	0.142
ACT	0.169	0.190	0.077	0.034	0.160	0.035	0.332	0.000	0.128	0.142
CN	0.509	0.330	0.174	0.051	0.057	0.090	0.372	0.104	0.164	0.162
AA	0.524	0.542	0.172	0.030	0.038	0.103	0.396	0.104	0.163	0.148
CRA	0.557	0.321	0.177	0.033	0.062	0.118	0.391	0.123	**0.199**	0.210
RA	0.545	**0.586**	0.151	0.030	0.020	0.102	0.425	0.083	0.165	0.187
LP	0.495	0.299	0.175	0.054	0.059	0.095	0.370	0.107	0.169	0.113
PA	0.130	0.012	0.069	0.054	0.025	0.051	0.318	0.012	0.096	0.025
Jaccard	0.521	0.301	0.017	0.001	0.017	0.001	0.064	0.001	0.001	0.180
SPM	**0.650**	0.576	0.231	0.055	0.204	0.120	0.322	0.160	0.190	**0.221**
LR	0.550	0.533	0.160	0.026	0.011	0.112	0.310	0.143	0.150	0.140
CK	0.541	0.555	0.070	0.050	0.145	**0.122**	0.435	0.169	0.185	0.113
LK	0.560	0.521	0.064	0.057	0.071	0.103	0.471	0.169	0.166	0.098

Table 5 Comparison of link prediction accuracy measured by PP on ten typical real-world networks

PP	Jazz	NS	PB	Power	Router	SmaGri	USAir	Yeast	Karate	School	mean
DNMF	15.70	25.63	20.21	30.15	36.17	20.06	20.71	28.52	10.67	12.93	22.08
SPM	16.05	15.38	20.04	30.07	36.13	21.03	19.04	28.25	15.38	13.34	21.47
CRA	15.38	23.98	18.89	27.85	30.96	20.95	19.88	27.11	10.87	13.12	20.90
CN	14.99	24.10	18.81	29.74	30.60	19.78	19.67	26.38	10.03	11.99	20.61
Katz	14.44	23.67	18.84	30.30	30.82	20.19	19.58	26.55	10.16	11.42	20.60
CK	15.25	15.32	14.86	29.66	34.65	21.10	20.35	28.49	15.32	10.43	20.54
LP	14.87	23.67	18.84	29.99	30.75	20.01	19.64	26.51	10.16	10.43	20.49
AA	15.11	26.25	18.76	27.44	28.84	20.36	19.94	26.38	10.00	11.60	20.47
RA	15.28	26.59	18.20	27.44	26.05	20.32	20.59	25.40	10.05	12.61	20.25
LK	15.40	15.25	14.47	30.23	31.55	20.36	20.69	28.49	15.25	9.81	20.15
NMF	15.31	23.14	17.96	26.09	27.02	17.48	19.01	27.64	9.81	12.25	19.57
LR	15.32	16.05	18.45	26.82	23.45	20.73	18.88	27.77	16.05	11.36	19.49
PA	9.06	9.70	14.80	29.99	27.02	17.31	18.99	17.00	7.70	3.88	15.55
ACT	10.20	21.70	15.27	27.98	35.08	15.68	19.17	−13.79	8.95	11.42	15.17
Jaccard	15.09	23.70	8.71	12.67	25.34	0.23	12.02	6.21	−12.12	12.45	10.43



(a) Yeast

(b) Jazz

(c) PB

(d) SmaGri

(e) USAir

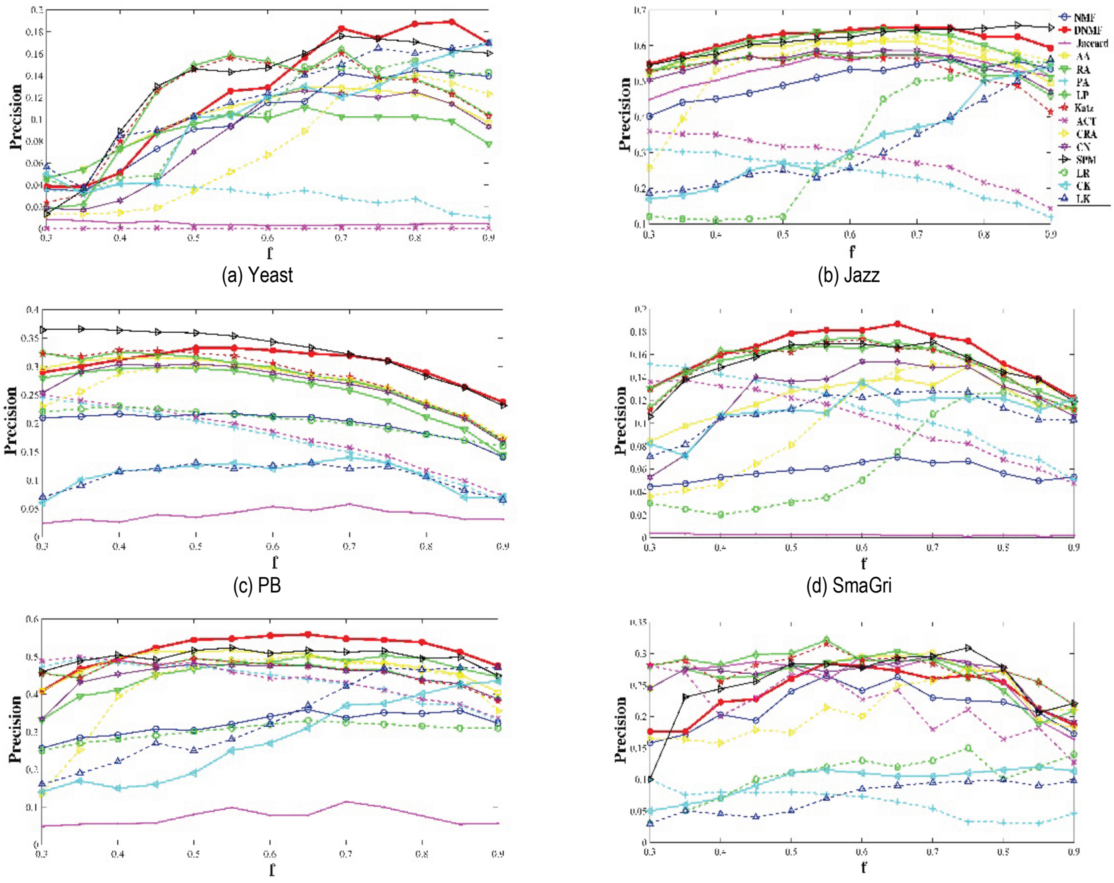
(f) School

Figure 2 Comparison of AUC of methods under different fraction of training sets on six real networks

To accurately test our proposed method, we analyze the experimental results on the six networks with different fraction of training set from 0.3 to 0.9. As shown in Figs. 2 to 4, we show the results of six networks based on AUC, Precision and PP, respectively. The results are returned with the average of over 100 runs. The six networks are Yeast, Jazz, PB, SmaGri, USAir and School. The red line with asterisk represents the performance of the proposed DNMF.

In Fig. 2, the AUC value of DNMF is consistently higher than AUC value of other methods on Yeast network, School network and SmaGri network, indicating that our method has the stable performance and can better perform when the training set is very small.

In Fig. 3, on Yeast network, PB network and USAir network, the precision value of DNMF is higher than other methods when the ratio of the training set increases from 0.7 to 0.9. This shows that DNMF can obtain a more obvious improvement than other methods. On the three evaluation indices, it can be seen that the proposed method is either the best or very close to the best, even with the size of the training set varied. Overall, it is shown that DNMF is superior to the traditional latent feature model based on non-negative matrix factorization. This suggests our proposed method for link prediction not only inherits the advantages of traditional NMF, but also takes full advantage of hierarchical latent structure information of networks by multi-layer learning. In general, it is obvious that our proposed method has better and competitive performance compared with baseline methods on the ten networks.



(a) Yeast

(b) Jazz

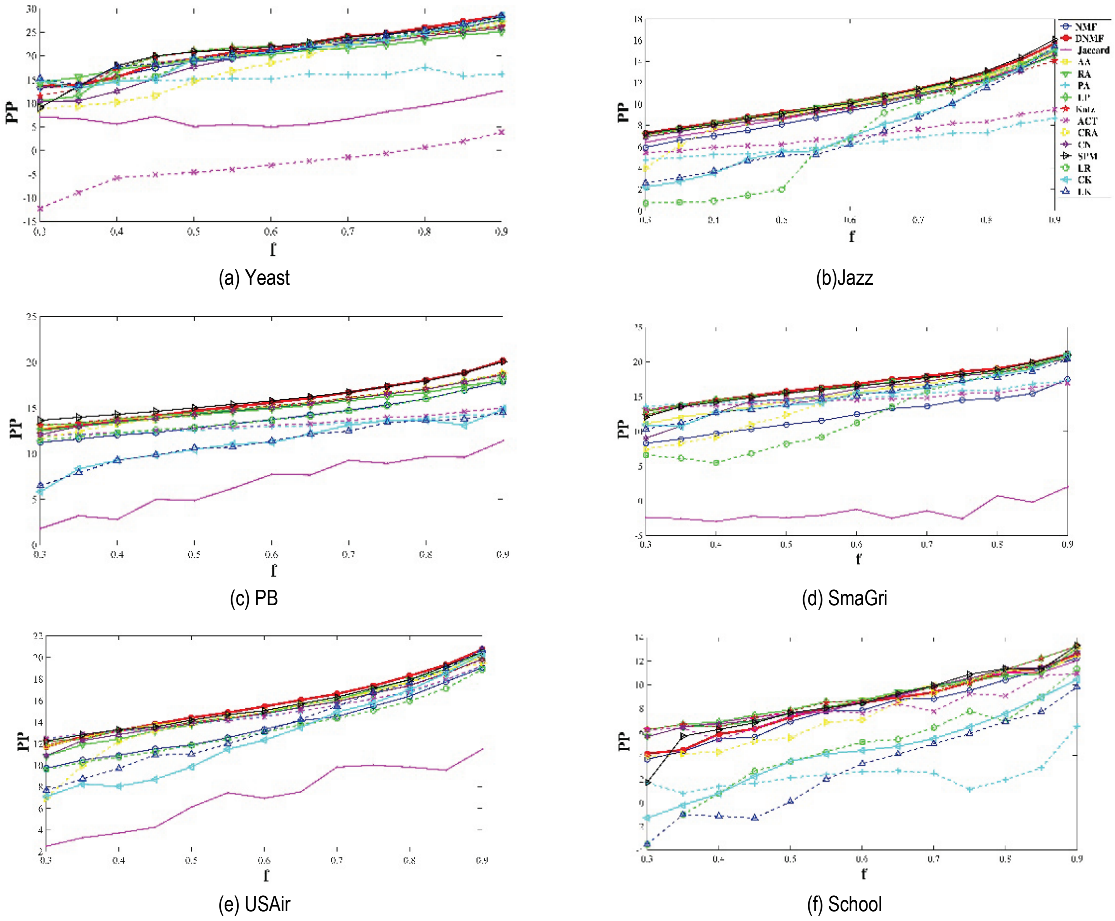
(c) PB

(d) SmaGri

(e) USAir

(f) School

Figure 3 Comparison of Precision of methods under different fraction of training sets on six real networks



(a) Yeast

(b)Jazz

(c) PB

(d) SmaGri

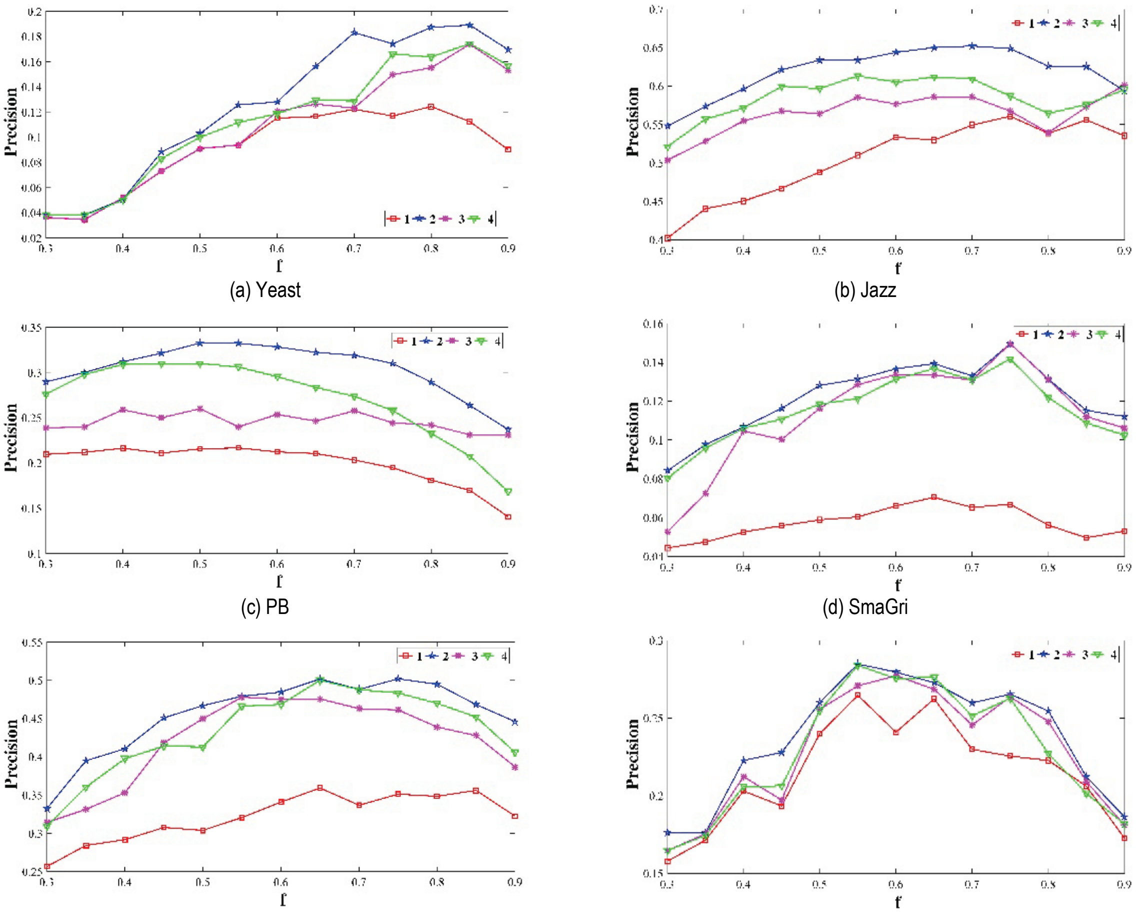
(e) USAir

(f) School

Figure 4 Comparison of PP of methods under different fraction of training sets on six real networks

3.5 Parameter Analysis

In order to analyze the effect of the layer number parameter m on the proposed algorithm DNMF, we show the precision of DNMF as the parameter m varying from 1 to 4 for the six networks including Yeast, Jazz, PB, SmaGri, USAir and School. As depicted in Fig. 5, we set fraction of training set from 0.3 to 0.9 and take the widely used evaluation index *Precision* for link precision as evidence. It is obvious that the performances are better when *m* is equal to 2, so we set *m* = 2 in most of experiments.



(a) Yeast

(b) Jazz

(c) PB

(d) SmaGri

(e) USAir

(f) School

Figure 5 Precision of DNMF with respect to the layer number m on six real networks

4 CONCLUSIONS

Most of the real networks are sparse, the traditional single-layer latent features model cannot fully characterize structure organization of complex networks. In order to resolve this problem, on the basis of non-negative matrix factorization and hierarchy information of latent features, a novel algorithm called Deep Non-negative Matrix Factorization (DNMF) is proposed for link prediction. In order to verify the performance of the proposed method, three evaluation metrics including AUC, Precision and Predictive Power (PP) are used. The experimental results of 10 real networks show that the proposed method DNMF is feasible, effective and competitive.

As an extension to the nonnegative matrix factorization, our proposed method DNMF for link prediction not only inherits merit of the traditional latent feature model, but also can reconstruct network through multi-layer factorization and extract more useful and richer feature information hidden in the original network. In order to reduce the training time of link prediction, the unsupervised learning strategy of the deep autoencoder network is applied in DNMF to improve the generalization ability of the method. So the proposed method has two stages including pre-training and fine-tuning for link prediction.

There are some improved studies and limitations for proposed method in the future. How to set the parameter layer number to be adaptive automatically on different networks and how to optimize the time complexity of the algorithm still are our next work. Parallelization computation can be used to reduce the computation time.

Acknowledgements

This work was supported by the Housing and Urban Rural Development Science and Technology Project of Shandong Province (2018-K2-03), University Science and technology project of Shandong Province (J18KB088), National College Students' innovation and entrepreneurship training project (201710430050), National Natural Science Foundation of China (41601489), Innovation and Entrepreneurship Training Program for Shandong Grade College Students (201910430090).

5 REFERENCES

[1] Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, *64*(2), 025102. https://doi.org/10.1103/PhysRevE.64.025102

[2] Adamic L. A. & Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, *25*(3), 211-230. https://doi.org/10.1016/S0378-8733(03)00009-1

[3] Cannistraci, C. V., Alanis-Lobato, G., & Ravasi, T. (2013). From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific Reports*, 3, 1613-1625. https://doi.org/10.1038/srep01613

[4] Daminelli, S., Thomas, J. M., Durán, C., & Cannistraci, C. V. (2015). Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics, 17*(11), 113037. https://doi.org/10.1088/1367-2630/17/11/113037

[5] Martínez, V., Berzal, F., & Cubero, J. C. (2017). A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR), 49*(4), 69. https://doi.org/10.1145/3012704

[6] Agli, H., Bonnard, P., Gonzales, C., & Wuillemin, P. H. (2018). Incremental inference for probabilistic relational models and application to object-oriented rule-based systems, *Revue d'Intelligence Artificielle, 32*(1), 111-132. https://doi.org/10.3166/ria.32.111-132

[7] Sales-Pardo, M., Guimera, R., Moreira, A. A., & Amaral, L. A. N. (2007). Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences, 104*(39), 15224-15229. https://doi.org/10.1073/pnas.0703740104

[8] Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research, 9*(Sep), 1981-2014.

[9] Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and Its Applications, 390*(6), 1150-1170. https://doi.org/10.1016/j.physa.2010.11.027

[10] Lü, L., Pan, L., Zhou, T., Zhang, Y. C., & Stanley, H. E. (2015). Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences, 112*(8), 2325-2330. https://doi.org/10.1073/pnas.1424644112

[11] Pech, R., Hao, D., Pan, L., Cheng, H., & Zhou, T. (2017). Link prediction via matrix completion. *EPL (Europhysics Letters), 117*(3), 38002. https://doi.org/10.1209/0295-5075/117/38002

[12] Wang, W., Feng, Y., Jiao, P., & Yu, W. (2017). Kernel framework based on non-negative matrix factorization for networks reconstruction and link prediction. *Knowledge-Based Systems, 137*, 104-114. https://doi.org/10.1016/j.knosys.2017.09.020

[13] Song, X. R., Gao, S., & Chen, C. B. (2018). A novel vehicle feature extraction algorithm based on wavelet moment, *Traitement du Signal, 35*(3-4), 223-242. https://doi.org/10.3166/ts.35.223-242

[14] Menon, A. K. & Elkan, C. (2011). Link prediction via matrix factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 437-452. https://doi.org/10.1007/978-3-642-23783-6_28

[15] Miller, K., Jordan, M. I., & Griffiths, T. L. (2009). Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, 1276-1284.

[16] Hoff, P. D. (2009). Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, *15*(4), 261-272. https://doi.org/10.1007/s10588-008-9040-4

[17] Yang, Q., Dong, E., & Xie, Z. (2014). Link prediction via nonnegative matrix factorization enhanced by blocks information. In *2014 10th International Conference on Natural Computation (ICNC)*, 823-827. https://doi.org/10.1109/ICNC.2014.6975944

[18] Cai, D., He, X., Han, J., & Huang, T. S. (2010). Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(8), 1548-1560. https://doi.org/10.1109/TPAMI.2010.231

[19] Cai, D., He, X., Wang, X., Bao, H., & Han, J. (2009). Locality preserving nonnegative matrix factorization. In *Twenty-First International Joint Conference on Artificial Intelligence*, 1010-1015.

[20] Wang, W., Chen, X., Jiao, P., & Jin, D. (2017). Similarity-based regularized latent feature model for link prediction in bipartite networks. *Scientific reports, 7*(1), 16996. https://doi.org/10.1038/s41598-017-17157-9

[21] Meeds, E., Ghahramani, Z., Neal, R. M., & Roweis, S. T. (2007). Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems*, 977-984.

[22] Shin, D., Si, S., & Dhillon, I. S. (2012, October). Multi-scale link prediction. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 215-224. https://doi.org/10.1145/2396761.2396792

[23] Bouchra, N., Aouatif, A., Mohammed, N., & Nabil, H. (2019). Deep belief network and auto-encoder for face classification. *IJIMAI*, 5(5), 22-29. https://doi.org/10.9781/ijimai.2018.06.004

[24] Lee, D. D. & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788-791. https://doi.org/10.1038/44565

[25] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504-507. https://doi.org/10.1126/science.1127647

[26] Hanley, J. A. & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29-36. https://doi.org/10.1148/radiology.143.1.7063747

[27] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS), 22*(1), 5-53. https://doi.org/10.1145/963770.963772

[28] Zhou, T., Lü, L., & Zhang, Y. C. (2009). Predicting missing links via local information. *The European Physical Journal B, 71*(4), 623-630. https://doi.org/10.1140/epjb/e2009-00335-8

[29] Barabási, A. L. & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*(5439), 509-512. https://doi.org/10.1126/science.286.5439.509

[30] Hamers, L. (1989). Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula. I*nformation Processing and Management, 25*(3), 315-318. https://doi.org/10.1016/0306-4573(89)90048-

[31] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika, 18*(1), 39-43. https://doi.org/10.1007/BF02289026

[32] Klein, D. J. & Randić, M. (1993). Resistance distance. *Journal of mathematical chemistry, 12*(1), 81-95. https://doi.org/10.1007/BF01164627

[33] Li, F., He, J., Huang, G., Zhang, Y., & Shi, Y. (2014). Retracted: A clustering-based link prediction method in social networks. *Procedia Computer Science, 29*, 432-442. https://doi.org/10.1016/j.procs.2014.05.039

[34] Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E, 74*(3), 036104. https://doi.org/10.1103/PhysRevE.74.036104

[35] Adamic, L. A. & Glance, N. (2005, August). The political blogosphere and the 2004 US election: divided they blog. In

Proceedings of the 3rd International Workshop on Link Discovery, 36-43. https://doi.org/10.1145/1134271.1134277

[36] Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. *Nature, 393*(6684), 440-442. https://doi.org/10.1038/30918

[37] Spring, N., Mahajan, R., & Wetherall, D. (2002, August). Measuring ISP topologies with Rocketfuel. *ACM SIGCOMM Computer Communication Review, 32*(4), 133-145. https://doi.org/10.1145/964725.633039

[38] Batagelj, V. & Mrvar, A. (2009). Pajek datasets (2006). https://link_springer.gg363.site/referenceworkentry/10.1007%2F978-1-4614-6170-8_310

[39] Von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., & Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, *417*(6887), 399-403. https://doi.org/10.1038/nature750

[40] Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, *33*(4), 452-473. https://doi.org/10.1086/jar.33.4.3629752

[41] Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (csur), 45*(4), 1-35. https://doi.org/10.1145/2501654.2501657

Contact information:

Fei CAI, associate professor
(Corresponding author)
College of Surveying and Geo-Informatics,
Shandong Jianzhu University,
Jinan 250101, China
E-mail: caifei@sdjzu.edu.cn

Jie CHEN, Master Degree Candidate
College of Surveying and Geo-Informatics,
Shandong Jianzhu University,
Jinan 250101, China
E-mail: Jiechensdjzu@163.cn

Xin ZHANG, Master Degree Candidate
College of Surveying and Geo-Informatics,
Shandong Jianzhu University,
Jinan 250101, China
E-mail: zxzhangxin@sina.com

Xiaohui MOU, Master Degree Candidate
College of Surveying and Geo-Informatics,
Shandong Jianzhu University,
Jinan 250101, China
E-mail: xhmou@sina.cn

Rongrong ZHU, college student
College of Surveying and Geo-Informatics,
Shandong Jianzhu University,
Jinan 250101, China
E-mail: zhurongrong19@sina.com