

RESEARCH ARTICLE

Link prediction based on non-negative matrix factorization

Bolun Chen^{1,2}, Fenfen Li¹, Senbo Chen^{2*}, Ronglin Hu¹, Ling Chen^{3,4}

1 College of Computer Engineering, Huaiyin Institute of Technology, Huaian, China, **2** School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, **3** Department of Computer Science, Yangzhou University, Yangzhou, China, **4** State Key Lab of Novel Software Tech, Nanjing University, Nanjing, China

* chensenbo@nuaa.edu.cn



Abstract

With the rapid expansion of internet, the complex networks has become high-dimensional, sparse and redundant. Besides, the problem of link prediction in such networks has also obtained increasingly attention from different types of domains like information science, anthropology, sociology and computer sciences. It makes requirements for effective link prediction techniques to extract the most essential and relevant information for online users in internet. Therefore, this paper attempts to put forward a link prediction algorithm based on non-negative matrix factorization. In the algorithm, we reconstruct the correlation between different types of matrix through the projection of high-dimensional vector space to a low-dimensional one, and then use the similarity between the column vectors of the weight matrix as the scoring matrix. The experiment results demonstrate that the algorithm not only reduces data storage space but also effectively makes the improvements of the prediction performance during the process of sustaining a low time complexity.

OPEN ACCESS

Citation: Chen B, Li F, Chen S, Hu R, Chen L (2017) Link prediction based on non-negative matrix factorization. PLoS ONE 12(8): e0182968. <https://doi.org/10.1371/journal.pone.0182968>

Editor: Zi-Ke Zhang, Hangzhou Normal University, CHINA

Received: April 5, 2017

Accepted: July 27, 2017

Published: August 30, 2017

Copyright: © 2017 Chen et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: This work was supported by China Scholarship Council; National Natural Science Foundation of China NO.61602202, 61379066 <https://isisn.nsf.gov.cn/egrantindex/funcindex/prjsearch-list>; and Natural Science Foundation of Jiangsu Province No.BK20160428, BK20161302 <http://www.jskj.gov.cn/13kjskj2/>.

Competing interests: The authors have declared that no competing interests exist.

Introduction

With the unprecedentedly rapid development of technology, the world has become increasingly complicated with frequent networking. In the real world, a number of information, biological, and social systems, ranging from interpersonal relationships to the colony structure, from transportation to the online world, from the ecosystem to the nervous system, can be considered as a network, in which vertices stand for the interactions between vertices or links and entities denote relations. Undoubtedly, there will be some potential links which are undetected and meanwhile there will be redundant links or some errors during the process of complicated network due to the limitations of space or time as well as the experimental conditions. Besides, based on the known network information, we are required to forecast the potential and missing links. This is the objectivity of the forecast challenge linked with the network, due to the dynamic development of the links of complicated network [1, 2].

In fact, the link prediction method can be utilized as an auxiliary means to investigate the structure of social network. Actually, it is essential to forecast the potential link of people or the future. There is an extensive scope of practical application values in varieties of areas in the

link prediction problem. For example, users' potential friends can be shown by the prediction of link, and even introduced to others in networks online [3]. Besides, the potential links between people can be found based on the analysis on the social relations [4–6]. Furthermore, the prediction of link could be utilized in the academic network to forecast the cooperators as well as the kind of an article [7]. The link between the nodes shows an interactive relation in the biological networks, such as disease-gene and metabolic networks as well as protein-protein interaction networks [8]. In addition, the research on link prediction has significant theoretical importance and an extensive scope of practical value [9–12]. For instance, it can offer a unified and convenient platform which can compare the mechanisms of network development more fairly and help comprehend the mechanism of the development of complicated networking theoretically, in order to develop the theoretical study on the model of complicated network evolution.

In recent decades, one of the link prediction problems of increasing interest revolves around the expansion of network size, the scoring matrix sparsely and the noise of the data. Due to high-dimensionality of user-rating matrix, the time and space complexity of personalized link prediction are increasing, which will cause a great impact on the performance of the prediction system. To deal with this problem, numerous new methods have been reported to improve the efficiency of link prediction. For example, Liu and Lv [13] used the local random walk instead of the global random walk to solve the link prediction problem, which can achieve good result. In [14], G. Rossetti et al defined the concept the multidimensional link prediction problem and proposed the several predictors based on node correlation. H. H. Song et al. [15] considered that the new edges would be linked in the near future and proposed the incremental update algorithm to deal with the large-scale networks.

In addition, with the consideration of the importance of network organization principle, Pan [16] proposed a predefined structural Hamiltonian algorithm which can be used to calculate the probability of non-observed links. Liu et al [17] proposed a local naive Bayes model based on the fact that various neighbors might have various functions, thus resulting in various outcomes. Hu [18] put forward a new model that examines the performances of the algorithms of state-of-the-art recommendation in the constantly developing network. Besides, it was also found that with the passage of time, the accuracy of recommendation gradually reduces if the online network evolution completely depends on the recommendation. In order to optimize the weights applied in a linear combination of node similarity indices and sixteen neighborhood, a method is provided by Bliss to predict the links in the future with the application of Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [19]. Barzel B [20] used global silencing of indirect correlations to make the prediction of missing links. A global algorithm of optimization is presented by Zhu which can infer the potential space effectively. Two alternative algorithms of optimization with incremental and local updates are also put forward to make the model scale to the bigger network without compromising the accuracy of forecast [2]. A nonparametric method has been proposed to link prediction in large-scale dynamic networks by Sarkar [21]. They considered that the characteristics of pairs of nodes based on graph and those of the local neighborhoods are utilized through using the model to forecast whether the nodes can be linked at every step of time. Richard [22] proposed a vector autoregressive (VAR) model based on the node degree to enhance the forecast accuracy. Zhang et al [23] proposed a link prediction algorithm based on an integrated diffusion on user-item-tag tripartite graphs and found that the usage of tag information can significantly improve novelty of link prediction. Recently, some swarm intelligence methods are also involved in link prediction. For example, Sherkat [24] proposed a new unsupervised structural ant colony approach algorithm which has the best outcome in some networks. Despite that those methods are designed

especially for the large scale networks, the accuracy of the results cannot be guaranteed due to the limit of computation time [25].

Nowadays, matrix decomposition technology has been widely applied, because it is relatively simple and able to obtain high prediction accuracy [26]. There are numerous well-known matrix decomposition methods, such as: singular value decomposition (SVD) [27], principal component analysis (PCA) [28, 29], independent component analysis (ICA) [30–32]. In SVD, it requires complementing scoring matrix in order to solve the sparsity of the network, but the operation would seriously increase the data storage space, therefore and it is not practical in the link prediction systems. Furthermore, due to the high time complexity of SVD, it is not applicable for large-scale network. With the aim to improve the performance of the SVD based method, SimonFunk [33] proposed LFM models. However, in the actual scoring system, the user ratings for goods do not have a uniform standard, which are of large arbitrariness due to the personalized habits on selecting goods. LFM model does not take the impact of the users' history buying into consideration. To overcome such shortcomings of LFM model, Koren [34] proposed a SVD++ model based on the LFM joined with the user's history scoring.

However, the existing matrix factorization models do not consider the situation that negative elements exist in the matrix. In real world applications, the a negative score by a user for an object does not necessarily mean that this object will never be selected by the user As time goes on, maybe this user will become interested in this object. Therefore, Lee and Seung [35, 36] proposed a method based on non-negative matrix factorization (NMF). It is a collection of algorithms which depends on linear algebra and multivariate analysis where two matrices without any negative elements consist in the original matrix. It is because of this non-negativity that makes it easier to analysis the final matrices. Therefore, NMF technology has been widely used in data mining [37, 38], image analysis [39–41], medical imaging [42] and voice processing [43], etc. However, in the link prediction, it has not attracted broad attentions. In this paper, we will apply NMF technology in link prediction and propose the NMF-LP model.

Materials and methods

Problem formulation and evaluation methods

Using an undirected and simple network which has node attitudes $G(U, V, E)$, in which E means the series of links, U refers to the series of nodes, and V is the series of node attitudes can represent a network. G does not allow various self-connections and links. Let $M = |V|$ be the amount of node attitudes and $N = |U|$ be the amount of nodes in G . U is considered to represent the universal set which all the possible links of $N(N - 1)/2$ are consisted in. Link prediction aims to identify the links which will occur in the future or the missing links in the series of nonexistent links $U - E$.

Assigning $S(x, y)$ as a score which stands for the similarities between the two nodes to every pair of the nodes $(x, y) \in U$ is our method. In a pair of nodes, (x, y) in $U \setminus E$, larger $S(x, y)$ tends to lead to higher possibility in a link between the nodes.

To test the accuracy of the results by our algorithm, the observed links in E are randomly divided into two parts: the training set, E^T , which is treated as known information, while the probe set (i.e., validation subset), E^P , which is used for testing and no information in this set is used for prediction. $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. In principle, a link prediction algorithm provides an ordered list of all non-observed links (i.e., $U - E^T$) or equivalently gives each non-observed link, say $(x, y) \in U - E^T$, a score S_{xy} to quantify its existence likelihood. To evaluate the accuracy of prediction algorithms, there are two standard metrics: *AUC* and *Precision*.

(1) *AUC*

The value of AUC is the area under the curve of ROC(receiver operating characteristic). we randomly choose a missing link and nonexistent link. After doing n times' independent evaluations, there should have n' times' missing link which shows a similar score and a higher score. Under this situation, the value of AUC is as below:

$$AUC = \frac{n' + 0.5n''}{n} \tag{1}$$

Generally, a bigger value of AUC shows better performance; thus, the AUC result through a randomly selected predictor is 0.5 while the AUC value of the perfect result is 1.0.

(2) Precision

Given the ranking of the non-observed links, the Precision is defined as the ratio of relevant objects selected to the total number of objects selected. That is to say, if we take the Top- L links as the predicted ones, among which m links are right, then the Precision can be expressed as:

$$Precision = \frac{m}{L} \tag{2}$$

Clearly, higher precision means higher prediction accuracy.

Non-negative matrix factorization and link prediction

For the reader's reference, Table 1 summarizes frequently used notations.

Here $A \in R_+^{n \times n}$ and $B \in R_+^{n \times m}$, where m is the number of attributes. The values of B can be defined within the range $[0,1]$ by normalizing its row vectors. We assume the adjacency matrix A as a non-negative characteristic matrix where each column represents the characteristic vector of a vertex and the goal of NMF is to obtain two non-negative matrices $U \in P_{n \times k}$ and $V \in P_{k \times n}$ so that their product is very close to matrix A :

$$A = UV \tag{3}$$

Here, k is the dimension of the latent space ($k < n$). U consists of the bases of the latent space, and is called the base matrix. V represents the combination coefficients of the bases for reconstructing the matrix A , and is called the coefficient matrix. Generally, this decomposition

Table 1. Notations and their meanings.

Notation	Meaning
A	adjacent matrix of the network
B	attribute similarity matrix of the network
U	U consists of the bases of the latent space, and is called the base matrix of A
V	V represents the combination coefficients of the bases for reconstructing the matrix A , and is called the coefficient matrix.
$U^{(B)}$	$U^{(B)}$ consists of the bases of the latent space, and is called the base matrix of B
$V^{(B)}$	$V^{(B)}$ represents the combination coefficients of the bases for reconstructing the matrix B , and is called the coefficient matrix.
$Q, Q^{(B)}$	$Q, Q^{(B)}$ are auxiliary diagonal matrixes
F, G	$Q, F \in R_+^{n \times m}$ and $G \in R_+^{k \times k}$ are auxiliary diagonal matrixes
ε	error between the original matrix and the product of the base matrix and weight matrix

<https://doi.org/10.1371/journal.pone.0182968.t001>

problem can be modeled as the following Frobenius norm optimization problem:

$$\min_{u,v} \| A - UV \|_F^2 \quad s.t. U \geq 0, V \geq 0 \tag{4}$$

Here, $\|\cdot\|_F$ is the Frobenius norm, constrain $U \geq 0$ and $V \geq 0$ requires that all the elements in matrixes U and V are nonnegative. The similarity between nodes i and j in the latent space can be represented by the similarity between the i^{th} and j^{th} row vectors in matrix V . Similarly, NMF form of the attribute matrix B is as follows:

$$B = U^{(B)} V^{(B)} \tag{5}$$

Therefore, we need to solve the optimization problem as follows:

$$\min_{u^{(B)}, v^{(B)}} \| B - U^{(B)} V^{(B)} \|_F^2 \quad s.t. U^{(B)} \geq 0, V^{(B)} \geq 0 \tag{6}$$

Here, $U^{(B)} \geq 0$ and $V^{(B)} \geq 0$ are respectively $n * k$ and $k * m$ non-negative matrixes.

In order to integrate the topology matrix A and the attribute matrix B , we map them into a latent space by a same projection. Such mapping can be implemented by non-negative matrix factorizings on A and B . However, when we factorize A and B at the same time, we cannot ensure that the projection matrixes V and $V^{(B)}$ are identical. Therefore, we use matrix V^* to make V and $V^{(B)}$ has the minimal distance between V and $V^{(B)}$, namely, we need to minimize the formula as follows:

$$\lambda \| V - V^* \|_F^2 + \mu \| V^{(B)} - V^* \|_F^2 \quad s.t. U^{(B)} \geq 0, V^B \geq 0. \tag{7}$$

Therefore, our goal is to solve the following the optimization problem:

$$\begin{aligned} \min_{u, u^{(B)}, v, v^{(B)}} & \| A - UV \|_F^2 + \| B - U^{(B)} V^{(B)} \|_F^2 + \lambda \| V - V^* \|_F^2 + \mu \| V^{(B)} - V^* \|_F^2 \\ s.t. & \| U \|_1 = 1, \| U^{(B)} \|_1 = 1, U \geq 0, V \geq 0, U^{(B)} \geq 0, V^{(B)} \geq 0. \end{aligned} \tag{8}$$

In order to remove the constrains $\|U\|_1 = 1$ and $\|U^{(B)}\|_1 = 1$ in Eq (8), we define auxiliary diagonal matrixes Q and $Q^{(B)}$ as follows:

$$Q = \text{diag}(\sum_i u_{i1}, \sum_i u_{i2}, \dots, \sum_i u_{ik}) \tag{9}$$

$$Q^{(B)} = \text{diag}(\sum_i u_{i1}^{(B)}, \sum_i u_{i2}^{(B)}, \dots, \sum_i u_{ik}^{(B)}) \tag{10}$$

The matrixes U and $U^{(B)}$ can be normalized into UQ^{-1} and $U^{(B)}Q^{(B)-1}$. Since $UV = (UQ^{-1})(QV)$ and $U^{(B)}V^{(B)} = (U^{(B)}Q^{(B)-1})(Q^{(B)}V^{(B)})$, the matrixes V and $V^{(B)}$ can be normalized into VQ and $V^{(B)}Q^{(B)}$ respectively. Therefore the optimal problem (8) is equivalent to the one as follows:

$$\min_{u, u^{(B)}, v, v^{(B)}, v^*} J(u, u^{(B)}, v, v^{(B)}, v^*) \quad s.t. U \geq 0, V \geq 0, U^{(B)} \geq 0, V^B \geq 0. \tag{11}$$

Here,

$$\begin{aligned} J(u, u^{(B)}, v, v^{(B)}, v^*) = & \| A - UV \|_F^2 + \| B - U^{(B)} V^{(B)} \|_F^2 \\ & + \lambda \| QV - V^* \|_F^2 + \mu \| Q^{(B)} V^{(B)} - V^* \|_F^2 \end{aligned}$$

Therefore, our goal is to find the optimal resolution of the formula (11).

Iterative update rule of non-negative matrix

To solve the optimization problem in Eq (11), we present an iterative method. Since $U, V, U^{(B)}, V^{(B)}$ and V^* are variables in Eq (11), we fix four of the variables at each time in the iteration, and obtain the best value the fifth variable to minimize the objective function J . After setting proper initial values for matrices $U, V, U^{(B)}, V^{(B)}$ and V^* , each iteration consists of five steps: a. Fix the matrices $V, U^{(B)}, V^{(B)}$ and V^* , update U to minimize J ; b. Fix the matrices $U, U^{(B)}, V^{(B)}$ and V^* , update V to minimize J ; c. Fix the matrices $U, V, V^{(B)}$ and V^* , update $U^{(B)}$ to minimize J ; d. Fix the matrices $U, V, U^{(B)}$ and V^* , update $V^{(B)}$ to minimize J ; e. Fix the matrices $U, V, U^{(B)}$ and $V^{(B)}$, update V^* to minimize J .

The updating steps above are repeated until convergence.

Updating U . Fixing the matrices $V, U^{(B)}, V^{(B)}$ and V^* , minimizing J by updating U is equivalent to minimize the objective function:

$$F(U) = \| A - UV \|_F^2 + \lambda \| QV - V^* \|_F^2 \tag{12}$$

Derivation 1 To optimize the objective function (12), the following updating rule can be used:

$$U_{lm} = U_{lm} \frac{\sum_j A_{ij} V_{mj} + \lambda \sum_j V_{mj}^* V_{mj}}{\lambda \sum_j V_{mj}^2 \sum_h V_{hm} + \sum_j V_{mj} \sum_k U_{lk} V_{kj}} \tag{13}$$

Proof:

According to the definition of Frobenius norm definition, we can get:

$$\begin{aligned} F(U) &= \sum_i \sum_j A_{ij} - \sum_k U_{ik} V_{kj}^2 + \lambda \sum_i \sum_j \sum_k Q_{ik} V_{kj} - V_{ij}^{*2} \\ &= \sum_i \sum_j [A_{ij}^2 - 2A_{ij} \sum_k U_{ik} V_{kj} + (\sum_k U_{ik} V_{kj})^2] \\ &\quad + \lambda \sum_i \sum_j [(\sum_k Q_{ik} V_{kj})^2 - 2V_{ij}^* \sum_k Q_{ik} V_{kj} + (V_{ij}^*)^2] \end{aligned}$$

We take the derivative of $F(U)$ on U_{lm} :

$$\frac{\partial F(U)}{\partial U_{lm}} = -2 \sum_j (A_{ij} V_{mj}) + 2 \sum_j V_{mj} \sum_k U_{lk} V_{kj} + 2\lambda \sum_j V_{mj}^2 \sum_h U_{hm} - 2\lambda \sum_j V_{mj}^* V_{mj}$$

By Karush-Kuhn-Tucker (KKT) condition we know that $\frac{\partial F(U)}{\partial U_{lm}} = 0$, and can get:

$$-2 \sum_j (A_{ij} V_{mj}) + 2 \sum_j V_{mj} \sum_k U_{lk} V_{kj} + 2\lambda \sum_j V_{mj}^2 \sum_h U_{hm} - 2\lambda \sum_j V_{mj}^* V_{mj} = 0$$

Namely:

$$\sum_j (A_{ij} V_{mj}) + \lambda \sum_j V_{mj}^* V_{mj} = \sum_j V_{mj} \sum_k U_{lk} V_{kj} + \lambda \sum_j V_{mj}^2 \sum_h U_{hm}$$

So, the update rule of U is as follows:

$$U_{lm} = U_{lm} \frac{\sum_j A_{ij} V_{mj} + \lambda \sum_j V_{mj}^* V_{mj}}{\lambda \sum_j V_{mj}^2 \sum_h V_{hm} + \sum_j V_{mj} \sum_k U_{lk} V_{kj}}$$

Updating $U^{(B)}$. Derivation 2 we construct the function as follows:

$$F(U^{(B)}) = \|A - U^{(B)}V^{(B)}\|_F^2 + \mu \|QV^{(B)} - V^*\|_F^2 \tag{14}$$

the update rules for $F(U^{(B)})$ can be written as:

$$U_{lm}^{(B)} = U_{lm}^{(B)} \frac{\sum_j B_{ij} V_{mj}^{(B)} + \mu \sum_j V_{mj}^* V_{mj}^{(B)}}{\mu \sum_j (V_{mj}^{(B)})^2 + \sum_h U_{hm}^{(B)} + \sum_j U_{mj}^{(B)} \sum_k U_{lk} V_{kj}} \tag{15}$$

Derivation 2 can be proved in a similar way as Derivation 1.

Updating V . Derivation 3 we construct the function as follows:

$$G(V) = \|A - UV\|_F^2 + \lambda \|V - V^*\|_F^2 \tag{16}$$

the update rules for $G(V)$ can be written as:

$$V_{lm} = V_{lm} \frac{\sum_i A_{im} U_{il} + \lambda \sum_j V_{lm}^*}{\lambda \sum_i U_{il} \sum_k U_{ik} V_{km} + \lambda V_{lm}} \tag{17}$$

Proof:

According to the definition of Frobenius norm definition, we can get:

$$\begin{aligned} G(V) &= \sum_i \sum_j A_{ij} - \sum_k U_{ik} V_{kj}^2 + \lambda \sum_i \sum_j V_{ij} - V_{ij}^{*2} \\ &= \sum_i \sum_j [A_{ij}^2 - 2A_{ij}^2 \sum_k U_{ik} V_{kj} + (\sum_k U_{ik} V_{kj})^2] \\ &\quad + \lambda \sum_i \sum_j V_{ij} - V_{ij}^{*2} \end{aligned}$$

We take the derivative of $G(V)$ on V_{lm} :

$$\frac{\partial G(V)}{\partial V_{lm}} = -2 \sum_i (A_{im} V_{il}) + 2 \sum_i U_{il} \sum_k U_{ik} V_{km} + 2\lambda (V_{lm} - V_{lm}^*)$$

By Karush-Kuhn-Tucker (KKT) condition we know that $\frac{\partial G(V)}{\partial V_{lm}} = 0$, and can get:

$$-2 \sum_i (A_{im} V_{il}) + 2 \sum_i U_{il} \sum_k U_{ik} V_{km} + 2\lambda (V_{lm} - V_{lm}^*) = 0 \tag{18}$$

Namely:

$$\sum_i (A_{im} U_{il}) + \lambda V_{lm}^* = \sum_i U_{il} \sum_k U_{ik} V_{km} + \lambda V_{lm}$$

So, the update rule of V_{lm} is as follows:

$$V_{lm} = V_{lm} \frac{\sum_i A_{im} U_{il} + \lambda V_{lm}^*}{\sum_i U_{il} \sum_k U_{ik} V_{km} + \lambda V_{lm}}$$

Updating $V^{(B)}$. Derivation 4 we construct the function as follows:

$$G(V^{(B)}) = \|A - U^{(B)}V^{(B)}\|_F^2 + \mu \|V^{(B)} - V^*\|_F^2 \tag{19}$$

the update rules for $G(V^{(B)})$ can be written as:

$$V_{lm}^{(B)} = V_{lm}^{(B)} \frac{\sum_i B_{im} U_{il}^{(B)} + \mu V_{lm}^*}{\sum_i U_{il}^{(B)} \sum_k U_{ik}^{(B)} V_{km}^{(B)} + \mu V_{lm}^{(B)}} \tag{20}$$

Derivation 4 can be proved in a similar way as Derivation 3.

Updating V^* . Derivation 5 we construct the function as follows:

$$H(V^*) = \| QV - V^* \|_F^2 + \| Q^{(B)}V^{(B)} - V^* \|_F^2 \tag{21}$$

the update rules for $H(V^*)$ can be written as:

$$V_{lm}^* = \frac{1}{2} (Q_{ll} V_{lm} + Q_{ll}^{(B)} V_{lm}^{(B)}) \tag{22}$$

Proof:

According to the definition of Frobenius norm definition, we can get:

$$H(V^*) = \sum_i \sum_j (\sum_k V_{ik} Q_{kj} - V_{lm}^*)^2 - \sum_i \sum_j (\sum_k V_{ik}^{(B)} Q_{kj}^{(B)} - V_{ij}^*)^2$$

We take the derivative of $H(V^*)$ on V_{lm}^* :

$$\frac{\partial V^*}{\partial V_{lm}^*} = -2(Q_{ll} V_{lm} - V_{lm}^*) - 2(Q_{ll}^{(B)} V_{lm}^{(B)} - V_{lm}^*)$$

By Karush-Kuhn-Tucker (KKT) condition we know that $\frac{\partial H(V^*)}{\partial V_{lm}^*} = 0$, and can get:

$$Q_{ll} V_{lm} + Q_{ll}^{(B)} V_{lm}^{(B)} = 2V_{lm}^*$$

So, we can get the update rule of V_{lm}^* as follows:

$$V_{lm}^* = \frac{1}{2} (Q_{ll} V_{lm} + Q_{ll}^{(B)} V_{lm}^{(B)})$$

Framework of the NMF-LP algorithm for link prediction

Based on the iterative method for NMF computing, we present an algorithm named NMF-LP for link prediction based on non-negative matrix factorization. The framework of our algorithm NMF-LP is as follows.

Algorithm: NMF-LP (Non-negative Matrix Factorization based Link Prediction)

Input: A : Adjacency matrix of network;

B : Attitude matrix of network;

Output: S : Scoring matrix between the nodes;

Begin:

1. Matrix A, B regularization
2. Parameter initialization:
3. Set the initial values of $U, U^{(B)}, V, V^{(B)}, V^*$;
4. **Repeat**
5. Update U according to formula (13);
6. Calculate the the main diagonal elements of Q according to U ;
7. Update $U^{(B)}$ according to formula (15);
8. Update V according to formula (17);
9. Update $V^{(B)}$ according to formula (20);
10. Update V^* according to formula (22);

11. **Until** $\|A - UV\|_F^2 \leq \epsilon$
 12. Calculate the similarity between the column vectors of V^* ;
 13. Output the Scoring matrix S between the nodes;
- End**

Line 12 of the algorithm calculates the similarity between the column vectors of weight matrix V^* , and store the similarities in the the scoring matrix S , which is output on line 13 as the final result of link prediction.

Time complexity analysis

In each iteration, steps 5 to 10 require $O(n^2k)$ time. Since each row in matrix V^* is a k -dimensional vector, which takes $O(k)$ time to compute the similarity between such vectors. Therefore, step 12 requires $O(n^2k)$ time for compute the similarities for all pairs of the row vectors in V^* . Since k and t can be treated as constants, complexity of the algorithm is $O(n^2)$. In the similarity based link prediction methods, n^2 similarity scores between the node pairs must be computed. Therefore, $O(n^2)$ is the lower bound the time complexity of the similarity based link prediction methods.

Convergence analysis of the algorithm NMF-LP

In this section, we will prove the convergence and correctness of $U, V, U^{(B)}, V^{(B)}$ and V^* in their iterative process. In order to prove convergence of the algorithm, we will make use of an auxiliary function based on the following lemma:

Lemma 1 Let $F \in R_+^{n \times n}, G \in R_+^{k \times k}$ be two symmetric matrixes, $S \in R_+^{n \times k}$ and $S' \in R_+^{n \times k}$ be two $n \times k$ matrixes. Then we can get:

$$\sum_{i=1}^n \sum_{j=1}^k \frac{(FS'G)_{ij} S_{ij}^2}{S'_{ij}} \geq Tr[SF^T SG] \tag{23}$$

Proof:

We set $S_{ij} = S'_{ij} p_{ij}$, then the difference value between the two sides of the formula (22) is:

$$\Delta = \sum_{i,x=1}^n \sum_{j,y=1}^k F_{ix} S'_{xy} G_{yy} S'_{ij} (p_{ij}^2 - p_{ij} p_{xy})$$

Noticing that F, G are symmetric matrixes, we get:

$$\begin{aligned} \Delta &= \sum_{i,x=1}^n \sum_{j,y=1}^k F_{ix} S'_{xy} G_{yy} S'_{ij} \left(\frac{p_{ij}^2 + p_{xy}^2}{2} - p_{ij} p_{xy} \right) \\ &= \frac{1}{2} \sum_{i,x=1}^n \sum_{j,y=1}^k F_{ix} S'_{xy} G_{yy} S'_{ij} (p_{ij} + p_{xy})^2 \\ &\geq 0 \end{aligned}$$

Namely:

$$\sum_{i=1}^n \sum_{j=1}^k \frac{(FS'G)_{ij} S_{ij}^2}{S'_{ij}} \geq Tr[SF^T SG]$$

Theorem 1 is an auxiliary function for $L(H)$ if the conditions

$$Z(H, \tilde{H}) \geq L(H), Z(H, H) = L(H) \tag{24}$$

are satisfied.

Lemma 2 If Z is an auxiliary function, then L is nonincreasing under the update rule:

$$H(\tau + 1) = \arg \min_H Z(H, H(\tau)) \tag{25}$$

Based on the above thoughts of constructing auxiliary functions, we give the following lemma to prove the convergence of the algorithm NMF-LP.

Lemma 3 Fixing any four matrices in $U, V, U^{(B)}, V^{(B)}$ and V^* , and using the updating rules (13), (15), (17), (20) and (22) in each iteration of algorithm NMF-LP the value of objective function J is non-increasing.

Proof:

Firstly, we fix the value of $U, U^{(B)}, V^{(B)}$ and V^* to update V , we can translate formula (12) into the following optimization problem:

$$J(V) = \|A - UV\|_F^2 + \lambda \|V - V^*\|_F^2 \tag{26}$$

We can get the following equation according to the definition of Frobenius norm:

$$\begin{aligned} J(V) &= \text{tr}(-2AUV + V^T U^T UV) + \text{tr}(\lambda VV^T - 2\lambda V^*V) \\ &= \text{tr}(V^T U^T UV + \lambda VV^T) - \text{tr}(2AUV + 2\lambda V^*V) \end{aligned} \tag{27}$$

We construct the following function:

$$F(V', V) = \sum_{jk} \frac{(U^T UV + \lambda V)_{jk} (V'_{jk})^2}{V_{jk}} - \sum_{jk} 2(AU + \lambda V^*)_{jk} V_{jk} \left(1 + \log \frac{V'_{jk}}{V_{jk}}\right) \tag{28}$$

Then we prove $F(V', V)$ is an auxiliary function of $J(V)$. First, when $V' = V$, it is obvious $F(V, V) = J(V)$. If $V' \neq V$, and $\exists \frac{V'_{jk}}{V_{jk}} > 0$, we get:

$$\sum_{jk} \frac{(U^T UV + \lambda V)_{jk} (V'_{jk})^2}{V_{jk}} > \text{tr}(V^T U^T UV + \lambda VV^T)$$

because of $\frac{V'_{jk}}{V_{jk}} \geq 1 + \log \left(\frac{V'_{jk}}{V_{jk}}\right)$, $\forall \frac{V'_{jk}}{V_{jk}} > 0$, we know:

$$\sum_{jk} 2(AU + \lambda V^*)_{jk} V_{jk} \left(1 + \log \frac{V'_{jk}}{V_{jk}}\right) < \text{tr}(2AUV^T + 2\lambda V^*V)$$

Therefore, we get the following inequality:

$$F(V', V) \geq J(V')$$

Therefore $F(V', V)$ is an auxiliary function of $J(V')$.

Based on the definition of auxiliary function and Lemma 2, we know that if we find the V' value to reach the local minimum of auxiliary function $F(V', V^{[t]})$, then the value of $J(V)$ is non-increasing over t . Therefore, we find the V' value to minimize $F(V', V^{[t]})$ through fixing $V^{[t]}$.

In order to obtain the minimum value of function $F(V', V^{[t]})$, according to KKT conditions, we can obtain:

$$\frac{\partial F(V', V)}{\partial V'_{jk}} = \left\{ \frac{2(U^T UV + \lambda V)_{jk} V'_{jk}}{V_{jk}} - 2(AU + \lambda V^*)_{jk} \frac{V_{jk}}{V'_{jk}} \right\} = 0 \tag{29}$$

So, we can get the update following formula:

$$V'_{jk} = V_{jk} \frac{(AU)_{jk} + \lambda V_{jk}^*}{(U^T UV)_{jk} + \lambda V_{jk}} \tag{30}$$

Therefore, $J(V^{(t)})$ is non increasing according to update rule (17) and we can get the update formula about V_{jk} by replacing V'_{jk} and V_{jk} by V_{jk}^{t+1} and $V_{jk}^{(t)}$, respectively. Therefore, using Eq (17) at each iteration of algorithm NMF-LP, the value of objective function J is non-increasing.

In a similar way, we can prove that using the rules (13), (15) and (20) at each iteration of algorithm NMF-LP, for updating U , $U^{(B)}$, $V^{(B)}$ and V^* , respectively, the value of objective function J is also non-increasing.

Although it is not guaranteed that the process in Algorithm NMF-LP will converge to a global minimum, the end condition of iterations $\|A - UV\|_F^2 \leq \epsilon$ can ensure that the result matrixes UV is an acceptable factorization of A and will meet our requirements in solving the problem of link prediction.

Based on Eq (25), we know that the value of V by Eq (30) is also non-increasing. Since $V > 0$, it converges. The correctness of the converged solution is assured by the fact that at convergence, from Eq (30), the solution will satisfy

$$-2 \sum_i (A_{im} V_{il}) + 2 \sum_i U_{il} \sum_k U_{ik} V_{km} + 2\lambda (V_{lm} - V_{lm}^*) = 0$$

It is the same as the fixed point condition of Eq (18). In a similar way, the convergence and correctness of formula for updating U , $U^{(B)}$, $V^{(B)}$ and V^* can be proved.

Results

Test on networks without node attributes

In this section, we testified the reliability of our algorithm NMF-LP on six benchmark data sets which served networks without node attributes: US airport network(USAir), US political blogs (PB) network, coauthor-ships network between scientists(NS), protein-protein interaction network (PPI), electrical resource grid of the western US(Grid) and Internet(INT). For each non-connected network, we figured out the largest connected component. Table1 listed the topological characteristics of these largest ones from these applied networks, where N , M respectively mean the amount of nodes and links. $NUMC$ indicates the number of the components connected within network as well as the size of the largest component. For instance, 1222/2 can be explained as: for this network, there are 2 connected components, while the largest consists of 1222 nodes. In this table, e represents the network's performance, C and r are clustering and assortative coefficients. K represents the degree of average of network.

To estimate the reliability of outputs, a 10-fold cross-validation, which was randomly produced, was applied. For this applied cross-validation, the original nodes were randomly divided into 10 subsets. Among these 10, one subset served as the criterion data for testifying the reliability of our algorithm, while the other 9 subsets served as data for training. The process of cross-validation was then repeated for 10 times. The average of the 10 outcomes obtained from the folds can be taken as a single estimation.

In the first step of the non-negative matrix factorization, we need to set the column number of base matrix. We assume that the original adjacency matrix of N rows M columns, and then the column number (λ) of base vectors are required to satisfy: $(N + M)\lambda < NM$. Since $M = N$ in the network without node attitudes, we can get $\lambda < N/2$. In our experiments, we set $\lambda = N/2^i$,

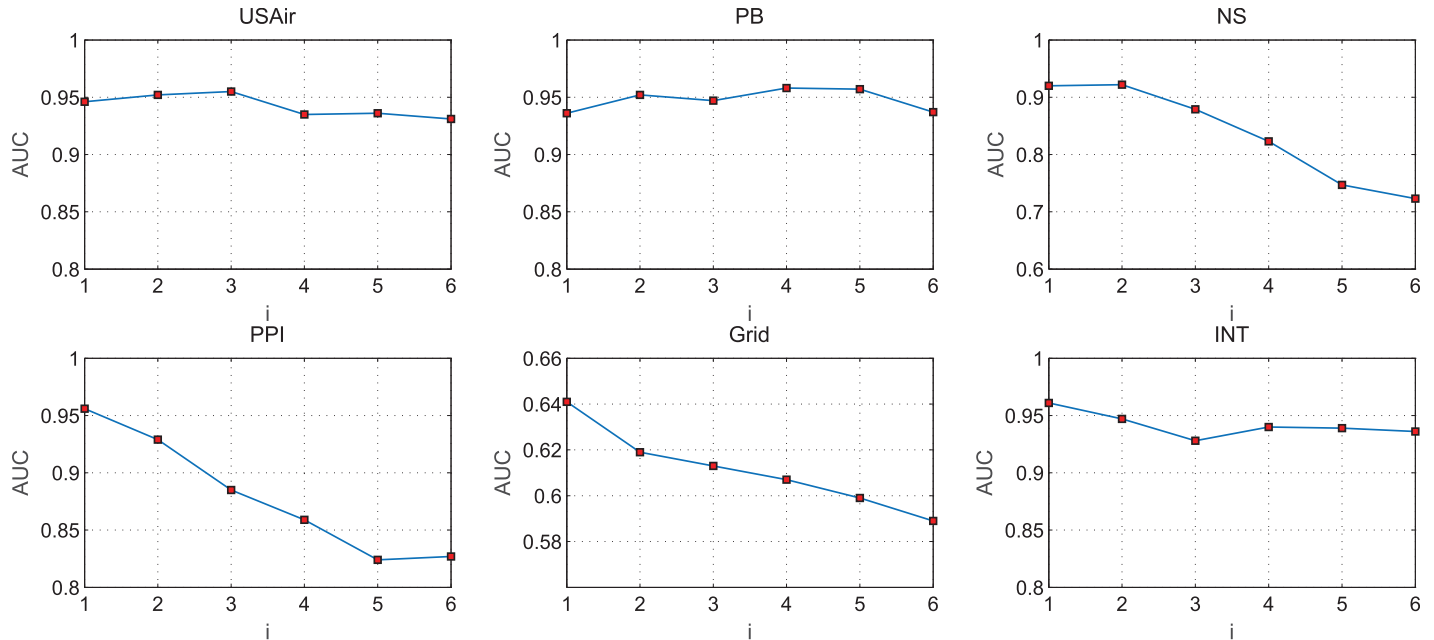


Fig 1. The AUC scores of NMF-LP algorithm with different values of i .

<https://doi.org/10.1371/journal.pone.0182968.g001>

where $i = [1, 2, \dots, 6]$. In the six data sets tested, the changes of the *AUC* scores with different i values of NMF-LP algorithm are shown in Fig 1.

From Fig 1, it can be seen that the *AUC* of NMF-LP algorithm shows a decrease trend when the i value increases. However it always keeps a high value in six data sets, indicating that NMF-LP algorithm not only can obtain a relatively higher *AUC* value on real networks than other algorithms, but also reduce data storage space.

And we also made comparison between the results produced by NMF-LP and the results of other algorithms such as CN, Jaccard, Sorensen, PA, Salton, HDI, LHN_I and HPI, the criterion applied was *AUC* score. Table 2 listed the averaged *AUC* scores of these tests via NMF-LP and other algorithms. In this table, ‘bold-face’ was applied to mark the highest *AUC* score for each data set through the 9 algorithms.

As can be seen from Table 3, for the 9 algorithms applied in this study, NMF-LP gained the highest *AUC* scores of most of these data sets. Even according to the data set INT, which was considered as the most difficult one, NMF-LP also performed very well, and the score gained was the highest: 0.961. This helped prove the high reliability of the algorithm NMF-LP.

Table 2. The topological features of the giant components of the six networks tested.

Networks	N	M	$NUMc$	e	C	r	K
USAir	332	2126	332/1	0.440	0.749	-0.228	12.807
PB	1224	19090	1222/2	0.397	0.361	-0.079	31.193
NS	1461	2742	379/268	0.016	0.878	0.462	3.754
PPI	2617	11855	2375/92	0.180	0.387	0.454	9.060
Grid	4941	6594	4941/1	0.056	0.107	0.004	2.669
INT	5022	6258	5022/1	0.167	0.033	-0.138	2.492

<https://doi.org/10.1371/journal.pone.0182968.t002>

Table 3. Comparison of algorithms' accuracy quantified by AUC.

	USAir	PB	NS	PPI	Grid	INT
CN	0.939	0.926	0.987	0.916	0.638	0.650
Salton	0.926	0.878	0.975	0.923	0.612	0.647
Jaccard	0.899	0.865	0.980	0.920	0.622	0.657
Sorensen	0.917	0.885	0.985	0.917	0.633	0.642
HPI	0.840	0.861	0.983	0.910	0.635	0.651
HDI	0.890	0.876	0.980	0.921	0.632	0.652
LHN_I	0.727	0.754	0.972	0.910	0.626	0.650
PA	0.896	0.908	0.671	0.854	0.577	0.959
NMF-LP	0.955	0.958	0.922	0.956	0.641	0.961

<https://doi.org/10.1371/journal.pone.0182968.t003>

Test on networks with node attributes

We also testified the reliability of our algorithm NMF-LP on networks with node attributes. We tested 8 sets of data, which can serve as networks. And all of these networks were selected from the website: Digital Bibliography Library Project (DBLP, <http://www.informatik.uni-trier.de/ley/db/>), a library website for computer science created by University Trier, Germany. It was built in 1980s, and had provided over 2.3 million articles involving computer science. Every significant journal of computer science can be tracked through this website. And also, proceeding papers of important conferences were also recorded by this website. It incorporated lots of collections of publications for CS research. According to our experiment, six datasets were applied, they are: CS Conference (ACM), Complex, Intelligent and Software Intensive Systems (CISIS), International World Wide Web Conferences(WWW), International Conference on Machine Learning (ICML), Applications of Natural Language to Data Bases (NLDB), International Conference on Information and Communication Security (ICICS). We chose some authors from each database. For example, in database ACM, the authors we chose are the people who had been a membership of the ACM conference from 1986 to 1996. We also set a network for every database in which each node was a specific author. And co-authorship between two authors were recorded by the link between the corresponding nodes. And for each author, we could master his overall publications. Key words in the paper titles could represent the attributes of the author. Because networks of some databases were not connected, our tests were only conducted on the largest ones. Table 4 listed the topological characteristics of these largest ones from these applied networks, where N , M respectively mean the amount of nodes and links. $NUMC$ indicates the number of the components connected within network as well as the size of the largest component. And e represents the network's performance, C and r are clustering and assortative coefficients. K represents the degree of average of network.

Table 4. The topological features of the giant components of the eight networks tested.

Networks	N	M	$NUMC$	e	C	r	K
ACM	1465	1960	3392/11	0.0014	0.3621	0.5570	3.3010
CISIS	2122	2385	4496/4	0.0078	0.7811	0.1491	7.7766
ICICS	888	1066	1944/3	0.0139	0.7484	0.2726	6.2973
ICML	2640	2213	4843/3	0.0719	0.6470	0.0132	7.2402
NLDB	847	1041	1863/6	0.0072	0.7130	0.2112	5.7190
WWW	5400	3421	1995/897	0.0182	0.7592	0.3724	7.8859

<https://doi.org/10.1371/journal.pone.0182968.t004>

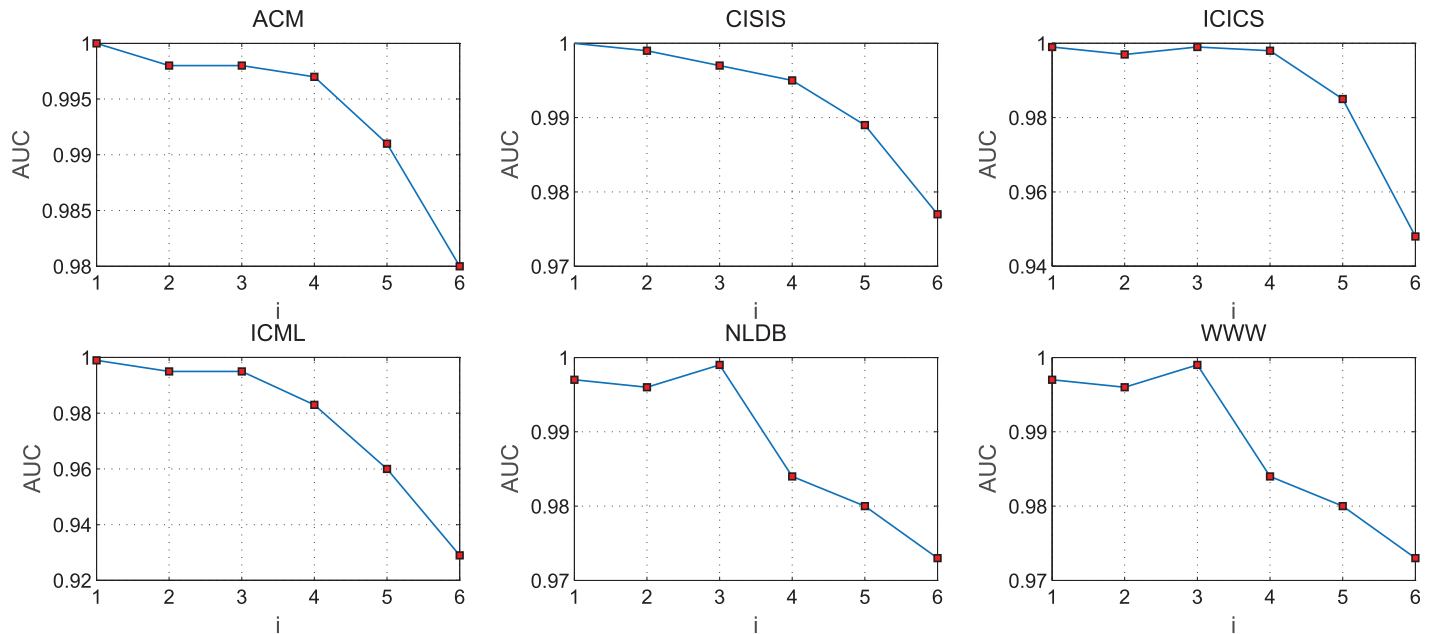


Fig 2. The AUC scores of NMF-LP algorithm with different values of i .

<https://doi.org/10.1371/journal.pone.0182968.g002>

In the process of the non-negative matrix factorization, we set $\lambda = (N + M)/2^{i+1}$, $i = [1, 2, \dots, 6]$, where N and M are the number of users and attributes. In the eight data sets tested, the changes of the AUC scores with different i values of NMF-LP algorithm are shown in Fig 2.

Similar to the test results on networks without node attitudes, Fig 2 shows that the AUC of NMF-LP algorithm presents a decrease trend when the i value increases. But it always keeps a high value as i increases in eight data sets. It also indicates that NMF-LP algorithm not only can decompose the scale of the original matrix and reduce dimension of the matrix, but also maintain a relatively higher AUC value with reducing the dimension of base vectors.

Apart from that, the AUC scores of the results have also been evaluated and compared by NMF-LP and other algorithms such as Salton, Sorensen, HPI, HDI, LHN_I, PA and CN. 10-fold CV tests which use NMF-LP and other algorithms can test the average AUC value, as is shown in Table 4. In the table, what is highlighted in bold-face is the highest AUC scores for each data which is set by the 9 algorithms.

In Table 5, we can see that NMF-LP has the highest AUC scores on all of the data sets in 9 algorithms. For instance, other algorithms get AUC scores less than 0.8635, yet algorithm NMF-LP obtains the highest AUC score 0.9980 in dataset ACM, which shows that the algorithm NMF-LP can obtain high quality results that have strong robustness.

Test on the time requirement of the algorithm

In the process of the non-negative matrix factorization, we calculate the error between the original matrix A and the product of the base matrix U and weight matrix V . Error formula is $error = \|A - UV\|_F^2$, we plot the relation between the error and the number of iterations in non-negative matrix factorization on different datasets. The results of tests on networks without node attributes are as shown in Fig 3, and Fig 4 shows the results of tests on networks with node attributes.

Table 5. Comparison of algorithms' accuracy quantified by AUC.

	ACM	CISIS	ICICS	ICML	NLDB	WWW
CN	0.8635	0.9618	0.9598	0.9236	0.9214	0.9505
Salton	0.7932	0.9620	0.9603	0.9238	0.9217	0.9417
Jaccard	0.4223	0.4475	0.3401	0.5223	0.3352	0.4426
Sorensen	0.8552	0.9620	0.9573	0.9236	0.9217	0.9525
HPI	0.8222	0.9621	0.9503	0.9238	0.9216	0.9305
HDI	0.8263	0.9619	0.9602	0.9235	0.9216	0.9505
LHN_I	0.8552	0.9618	0.9342	0.9234	0.9214	0.9504
PA	0.5541	0.5842	0.5335	0.5767	0.5154	0.5846
NMF-LP	0.9980	0.9770	0.9850	0.9600	0.9800	0.9600

<https://doi.org/10.1371/journal.pone.0182968.t005>

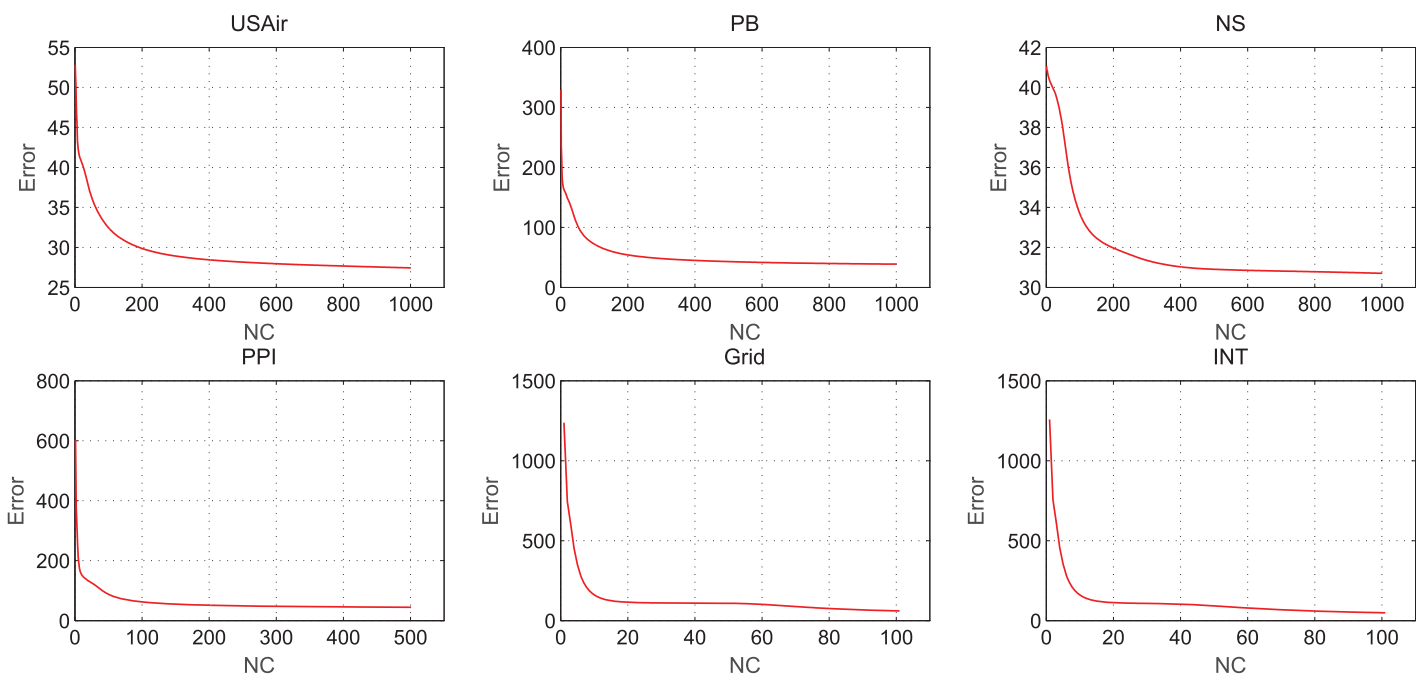


Fig 3. Error changes with the iteration number of NMF on networks without node attitudes.

<https://doi.org/10.1371/journal.pone.0182968.g003>

Based on Figs 3 and 4, we can see that in the process of non-negative matrix factorization, the tendency of error decreases sharply and it quickly arrives at the condition of convergence. The experiment results demonstrate that the algorithm not only reduces data storage space but also effectively improves the prediction performance while keeping a low time complexity.

Conclusion

With the expansion of network size, a large amount of redundant information causing by high-dimensionality and sparsely of actual network reduces the performance of the link prediction. In this paper, a new link prediction algorithm was proposed on the basis of non-negative matrix factorization. In the algorithm, the original adjacent and attribute matrixes of the network are effectively decomposed into the base matrix and the weight matrix which are non-negative. Thereafter, we use the similarity of the weight matrix as the scoring matrix

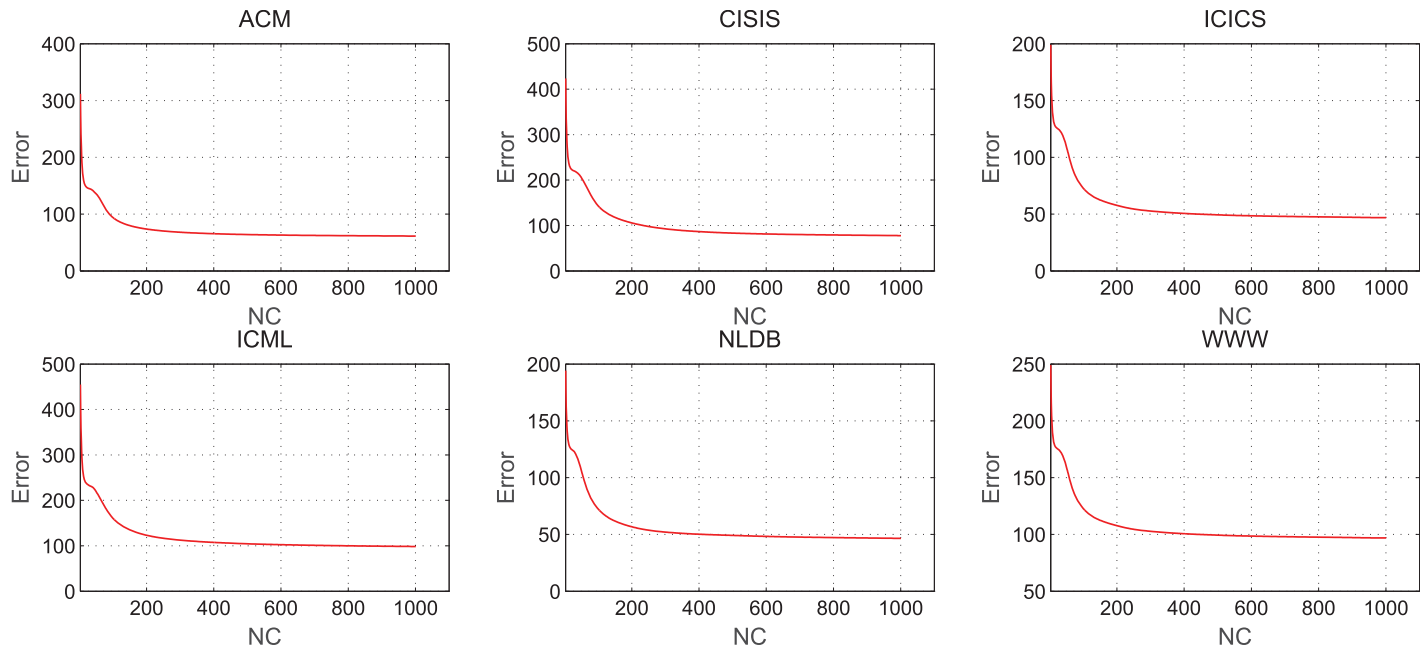


Fig 4. Error changes with the iteration number of NMF on networks with node attitudes.

<https://doi.org/10.1371/journal.pone.0182968.g004>

between the nodes. The experimental results show that one thing it can gain is higher quality results on real networks comparing with other algorithms, another thing is that it reduces data storage space while maintaining low time complexity.

Author Contributions

Conceptualization: BLC LC.

Data curation: FFL SBC.

Formal analysis: BLC FFL RLH.

Funding acquisition: BLC LC.

Investigation: FFL SBC.

Methodology: BLC FFL SBC.

Project administration: BLC SBC.

Resources: FFL RLH.

Software: BLC RLH.

Supervision: LC.

Validation: BLC LC.

Visualization: BLC SBC LC.

Writing – original draft: BLC FFL SBC LC.

Writing – review & editing: BLC FFL SBC RLH LC.

References

1. Tan F, Xia Y, Zhu B. Link prediction in complex networks: a mutual information perspective. *PLoS one*, 2014, 9(9): e107056. <https://doi.org/10.1371/journal.pone.0107056> PMID: 25207920
2. Zhu L, Guo D, Yin J, VerSteeg G, Galstyan A. Scalable Temporal Latent Space Inference for Link Prediction in Dynamic Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(10): 2765–2777. <https://doi.org/10.1109/TKDE.2016.2591009>
3. Papadimitriou A, Symeonidis P, Manolopoulos Y. Fast and accurate link prediction in social networking systems. *Journal of Systems and Software*, 2012, 85(9): 2119–2132. <https://doi.org/10.1016/j.jss.2012.04.019>
4. Tang J, Chang S, Aggarwal C, Liu H. Negative link prediction in social media. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 2015: 87–96.
5. Zeng A, Yeung CH. Predicting the future trend of popularity by network diffusion. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2016, 26(6): 063102. <https://doi.org/10.1063/1.4953013>
6. Hulovatyy Y, Solava RW, Milenkovic T. Revealing missing parts of the interactome via link prediction. *PLoS one*, 2014, 9(3): e90073. <https://doi.org/10.1371/journal.pone.0090073> PMID: 24594900
7. Liang W, He X, Tang D, Zhang X. S-Rank: A Supervised Ranking Framework for Relationship Prediction in Heterogeneous Information Networks. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer International Publishing, 2016: 305–319.
8. Hwang H, Petrey D, Honig B. A hybrid method for protein-protein interface prediction. *Protein Science*, 2016, 25(1): 159–165. <https://doi.org/10.1002/pro.2744> PMID: 26178156
9. Wang P, Xu BW, Wu YR, Zhou XY. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 2015, 58(1): 1–38.
10. Barbieri N, Bonchi F, Manco G. Who to follow and why: link prediction with explanations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014: 1266–1275.
11. Tan SY, Wu J, Lv L, Li M, Lu X. Efficient network disintegration under incomplete information: the comic effect of link prediction. *Scientific reports*, 2016, 6.
12. Yu L, Huang J, Zhou G, Liu C, Zhang ZK. TIIREC: A Tensor Approach for Tag-Driven Item Recommendation with Sparse User Generated Content. *Information Sciences*, 2017. <https://doi.org/10.1016/j.ins.2017.05.025>
13. Liu W, Lv L. Link prediction based on local random walk. *EPL (Europhysics Letters)*, 2010, 89(5): 58007. <https://doi.org/10.1209/0295-5075/89/58007>
14. Rossetti G, Berlingerio M, Giannotti F. Scalable link prediction on multidimensional networks. *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011: 979–986.
15. Song HH, Cho TW, Dave V, Zhang Y, Qiu LI. Scalable proximity estimation and link prediction in online social networks. *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009: 322–335.
16. Pan L, Zhou T, Lv L, Hu CK. Predicting missing links and identifying spurious links via likelihood analysis. *Scientific reports*, 2016, 6.
17. Liu Z, Zhang QM, Lv L, Zhou T. Link prediction in complex networks: A local naive Bayes model. *EPL (Europhysics Letters)*, 2011, 96(4): 48007.
18. Hu X, Zeng A, Shang MS. Recommendation in evolving online networks. *The European Physical Journal B*, 2016, 89(2): 1–7. <https://doi.org/10.1140/epjb/e2016-60509-9>
19. Bliss CA, Frank MR, Danforth CM, Dodds PS. An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, 2014, 5(5): 750–764. <https://doi.org/10.1016/j.jocs.2014.01.003>
20. Barzel B, Barabasi AL. Network link prediction by global silencing of indirect correlations. *Nature biotechnology*, 2013, 31(8): 720–725. <https://doi.org/10.1038/nbt.2601> PMID: 23851447
21. Sarkar P, Chakrabarti D, Jordan M. Nonparametric link prediction in large scale dynamic networks. *Electronic Journal of Statistics*, 2014, 8(2): 2022–2065. <https://doi.org/10.1214/14-EJS943>
22. Richard E, Gaffas S, Vayatis N. Link prediction in graphs with autoregressive features. *Journal of Machine Learning Research*, 2014, 15(1): 565–593.
23. Zhang ZK, Zhou T, Zhang YC. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 2010, 389(1): 179–186. <https://doi.org/10.1016/j.physa.2009.08.036>

24. Sherkat E, Rahgozar M, Asadpour M. Structural link prediction based on ant colony approach in social networks. *Physica A: Statistical Mechanics and its Applications*, 2015, 419: 80–94. <https://doi.org/10.1016/j.physa.2014.10.011>
25. Burgess M, Adar E, Cafarella M. Link-prediction enhanced consensus clustering for complex networks. *PLoS one*, 2016, 11(5): e0153384. <https://doi.org/10.1371/journal.pone.0153384> PMID: 27203750
26. Yu L, Liu C, Zhang ZK. Multi-linear interactive matrix factorization. *Knowledge-Based Systems*, 2015, 85: 307–315. <https://doi.org/10.1016/j.knosys.2015.05.016>
27. Franceschini A, Lin J, Mering CV, Jensen LJ. SVD-phy: improved prediction of protein functional associations through singular value decomposition of phylogenetic profiles. *Bioinformatics*, 2015: btv696. <https://doi.org/10.1093/bioinformatics/btv696> PMID: 26614125
28. Bro R, Smilde AK. Principal component analysis. *Analytical Methods*, 2014, 6(9): 2812–2831. <https://doi.org/10.1039/C3AY41907J>
29. Vidal R, Ma Y, Sastry SS. Principal Component Analysis. *Generalized Principal Component Analysis*. Springer New York, 2016: 25–62.
30. Du KL, Swamy MNS. Independent component analysis. *Neural Networks and Statistical Learning*. Springer London, 2014: 419–450.
31. Efimov D. Independent Component Analysis. *Encyclopedia of Social Network Analysis and Mining*, 2014: 724–728.
32. Xie J, Douglas PK, Wu YN, Brody AL, Anderson AE. Decoding the Encoding of Functional Brain Networks: an fMRI Classification Comparison of Non-negative Matrix Factorization (NMF), Independent Component Analysis (ICA), and Sparse Coding Algorithms. *Journal of Neuroscience Methods*, 2017, 282: 81–94. <https://doi.org/10.1016/j.jneumeth.2017.03.008> PMID: 28322859
33. Funk S. Netflix update: Try this at home. 2006.
34. Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2010, 4(1): 1. <https://doi.org/10.1145/1644873.1644874>
35. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999, 401(6755): 788–791. <https://doi.org/10.1038/44565> PMID: 10548103
36. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*. 2001: 556–562.
37. Ding C, He X, Simon HD. On the equivalence of nonnegative matrix factorization and spectral clustering. *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2005: 606–610.
38. Li Y, Ngom A. The non-negative matrix factorization toolbox for biological data mining. *Source code for biology and medicine*, 2013, 8(1): 10. <https://doi.org/10.1186/1751-0473-8-10> PMID: 23591137
39. Sandler R, Lindenbaum M. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, 2011, 33(8): 1590–1602. <https://doi.org/10.1109/TPAMI.2011.18>
40. Nikitidis S, Tefas A, Nikolaidis N, Pitsa I. Subclass discriminant nonnegative matrix factorization for facial image analysis. *Pattern Recognition*, 2012, 45(12): 4080–4091. <https://doi.org/10.1016/j.patcog.2012.04.030>
41. Sotiras A, Resnick SM, Davatzikos C. Finding imaging patterns of structural covariance via non-negative matrix factorization. *NeuroImage*, 2015, 108: 1–16. <https://doi.org/10.1016/j.neuroimage.2014.11.045> PMID: 25497684
42. Prajapati SJ, Jadhav KR. Brain Tumor Detection By Various Image Segmentation Techniques With Introduction To Non Negative Matrix Factorization. *Brain*, 2015, 4(3).
43. Mohammadiha N, Smaragdīs P, Leijon A. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *Audio, Speech, and Language Processing*, *IEEE Transactions on*, 2013, 21(10): 2140–2151. <https://doi.org/10.1109/TASL.2013.2270369>