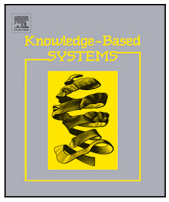
Knowledge-Based Systems 280 (2023) 110998

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Link prediction by adversarial Nonnegative Matrix Factorization

Reza Mahmoodi, Seyed Amjad Seyedi, Fardin Akhlaghian Tab, Alireza Abdollahpouri *

Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Keywords:
Link prediction
Adversarial training
Robustness
Generalization
Nonnegative matrix factorization

ABSTRACT

Networks are now more crucial than ever for modeling complex systems with interconnected components. Many methods have been developed to infer unobserved links or predict latent links based on observed network topology. However, the effectiveness of current methods is often limited due to the nature of sparseness in large-scale networks. Nonnegative Matrix Factorization (NMF) is one of the most popular low-rank approximation methods which has been successfully applied to large-scale data. However, maintaining generalization from limited observations is still an open challenge. To overcome these challenges, we propose a novel link prediction method based on adversarial NMF, which reconstructs a sparse network by an efficient adversarial training algorithm. Unlike the conventional NMF methods, our model considers potential test adversaries beyond the pre-defined bounds and provides a robust reconstruction with good generalization power. Besides, to preserve the local structure of a network, we use the common neighbor algorithm to extract the node similarity and apply it to low-dimensional latent representation. Simultaneously, we use Frobenius-norm regularization to prevent the factorization from overfitting. We provide an effective Majorization-Minimization method to learn the model parameters. Our method outperforms the state-of-the-art methods, as shown by extensive experiment results on twelve real-world datasets.

1. Introduction

Complex networks, such as microbial networks, protein networks, neural networks, social networks, etc., have been utilized to characterize complex systems in nature, biology, engineering, and human society [1]. Complex networks' latent functions and common rules may be discovered and interpreted by analyzing their structure. As a result, research on the study and analysis of complex networks, such as those involving network dynamics, community, and influential node discoveries [2], has grown popular across many different scientific fields. A connection between two nodes in a network is known as a link. This simple idea may be used to describe extremely complex systems in which several constituents interact with one another. The data proliferation that can be represented as networks has created new opportunities and challenges in the data mining field.

Link prediction aims to predict the possibility of unknown or unobserved links by analyzing the observed structure of complex networks, which can also explore the evolution of the dynamics in networks [3]. Link prediction has great value in various modern applications such as, recommender systems [4], author citation networks [5], protein–protein interaction networks [6], gene regulatory networks [7], and Terrorist Organization Identification [8]. For instance, the link prediction is used to examine the generative process of enormous actual networks or to anticipate the roles or interactions of proteins in biological networks [9]. It can also be used to suggest friends on online social networks [10].

Several methods for link prediction have been proposed in recent years [11]. The commonly used link prediction algorithms fall under the similarity-based algorithms, which operate on the presumption that two nodes are more likely to be linked if they are similar, given the observed network topology. Common Neighbors [12] and Adamic-Adar score [13] are two popular similarity-based algorithms that use the local structure network. Other path-based similarity-based algorithms, such as Katz [14] and Rooted PageRank [11] methods, heavily depend on network data regarding their general structure. Other algorithms use maximum-likelihood estimation, such as stochastic relational models [15], probabilistic entity-relationship models [16], probabilistic relational models [17], stochastic block models [18], and hierarchical structure models [19]. These methods endeavor to model complex networks and estimate the parameters of network that could be used to determine the probability of connection between each pair of nodes.

The link prediction issues that have drawn more attention in recent years focus on the growth of the network size, the sparse scoring matrix, and the data noise. The performance of the prediction system will

* Corresponding author.
E-mail addresses: reza.mahmoodi@uok.ac.ir (R. Mahmoodi), amjadseyedi@uok.ac.ir (S.A. Seyedi), f.akhlaghian@uok.ac.ir (F. Akhlaghian Tab), abdollahpouri@uok.ac.ir (A. Abdollahpouri).

https://doi.org/10.1016/j.knosys.2023.110998
Received 19 October 2022; Received in revised form 17 June 2023; Accepted 12 September 2023
Available online 18 September 2023

be significantly affected by the high dimensionality of the adjacency matrix, which increases the time and space complexity of customized link prediction. Numerous novel techniques have been developed to increase effectiveness of link prediction in solving high-dimensionality problems. Data representation is a fundamental problem in pattern recognition and machine learning, and it is often used in both supervised and unsupervised learning [20]. Its goal is to learn features in the hidden space from the high-dimensional data. To analyze the high-dimensional data, several data representation techniques have been developed, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), deep representation learning [21], and Nonnegative Matrix Factorization (NMF) [22].

However, link prediction still faces some other difficulties. The adjacency matrix is very sparse, and the number of unobserved links often exceeds that of the observed links. In sparse graphs, there may not be enough data available to learn meaningful embeddings for all nodes. This leads to overfitting in the deep models, where embeddings are not representative of the true underlying structure of the graph. Second, there are a few glaring flaws in the earlier related algorithms. Similarity-based algorithms are limited to the network's topological structure and cannot provide extremely insightful information about how the network is organized [23]. Therefore, creating an effective algorithm to address the issues mentioned above is important.

One of the widely used techniques for link prediction [24] and community discovery [25,26] is Nonnegative Matrix Factorization, which offers more interpretability and flexibility than other representation techniques [27–29]. NMF is a holistic model which can effectively capture the underlying patterns and relationships between observations. Moreover, it is capable to use additional information to make a better representation. This factorization model can handle data sparsity by decomposing a sparse matrix into two lower-rank nonnegative matrix factors, $\boldsymbol{W}$ (also known as NMF bases) and $\boldsymbol{H}$ (also known as its combining coefficients). In order to forecast the missing connections in link prediction task, the network is rebuilt using $\boldsymbol{W}$ and $\boldsymbol{H}$ [22]. For example, Menon and Elkan [30] have introduced a latent feature learning method via matrix factorization for network reconstruction. Nevertheless, the Matrix Factorization models suffer from the limitations of getting stuck in bad local minima, which is inevitable due to the non-convexity of the problem. Therefore, most of the existing MF models are still vulnerable to noisy links.

Goodfellow et al. [31] in their seminal work showed that machine learning methods are vulnerable to adversarial attacks on observations. To mitigate this vulnerability and to enhance generalization capabilities, adversarial training is becoming more popular. Recently generative adversarial-based methods [32,33] have been expanded for learning robust representation. More recently, Sinha et al. [34] and Farnia et al. [35] provided theoretical proof that adversarial training improves generalization in broad machine learning tasks. Adversarial nonnegative matrix factorization incorporates adversarial learning to increase robustness of factorization and to prevent model from getting stuck in local minima [36]. Therefore, the Adversarial NMF can learn more meaningful embedding for data.

The most link prediction models can easily overfit to the limited number of links in the network, resulting in poor generalization. By incorporating adversarial learning into the NMF framework, the factorization model can overcome some of the challenges of link prediction in sparse networks. Specifically, adversarial learning can help the model learn more robust factors that capture the underlying structure of the network, even in the presence of missing or noisy data. The proposed model employs an adversarial training technique by incorporating an attacker matrix into the input adjacency matrix. In this training, the attacker iteratively generates fake links to prevent model from fitting to the limited observed links. In other words, by deliberately perturbing both existing and non-existent links within a defined limit, we purposefully augment the training links and mitigate the sparsity issue. Therefore, the adversarial training encourages the model to produce embeddings that are robust to small changes in the input, which can improve the model's ability to generalize to new links.

In this research, we investigate an adversarial version of the conventional NMF objective function to enhance the generalization of NMF on network reconstruction tasks. We incorporate an adversarial training scheme by adding an attacker to the NMF model for discovering a general and robust graph representation. The proposed method not only minimizes the reconstruction errors of the topological structure but also extracts similar representations according to a local similarity. Our algorithm encodes the network in the low-dimensional space by exploiting local graph structure using a manifold regularization. Graph representation and adversarial learning are jointly optimized in a unified framework to benefit each other and lead to better graph reconstruction. Besides, a smoothness regularization is added to the proposed model to increase generalization capacity and avoid overfitting observed links. Our contributions can be summarized below:

- Motivated by the generalization properties of adversarial training, we propose a robust adversarial matrix completion algorithm. In this method, to improve the reconstruction performance, an attacker adds a bounded adversary matrix to the adjacency matrix, and the learner tries to minimize the reconstruction error.
- We propose a graph regularization that exploits more useful link information and preserves the local topology information.
- We use a Majorization-Minimization method for optimizing our adversarial framework and derive multiplicative updating rules to learn the parameters of LPANMF.
- Experiments on twelve real-world networks with two measurements are conducted and show that LPANMF outperforms state-of-the-art link prediction algorithms.

The remainder of this paper is divided into the following sections. We first introduce the background of Nonnegative Matrix Factorization and link prediction in Section 2. The adversarial training based on NMF for link prediction is introduced in Section 3. Section 4 represents the numerical and graphical results on real-world datasets. Finally, the conclusion is described and future work is presented in Section 5.

2. Background

In this section, we discuss the background of our research. Section 2.1 presents the notations that we use in this paper. Section 2.2 introduces the formulation of nonnegative matrix factorization. Section 2.3 discusses prior works related to link prediction.

2.1. Notations

In this work, scalars are denoted by lowercase italic letters (i.e., i, j, n, etc.) while vectors and matrices are denoted by bold lowercase letters (i.e., $\boldsymbol{a}$, $\boldsymbol{x}$, etc.) and bold uppercase notations (i.e., $\boldsymbol{W}$, $\boldsymbol{H}$, etc.) show vectors and matrices, respectively. For any matrix $\boldsymbol{A}$, $\boldsymbol{a}_i$ means the ith column of $\boldsymbol{A}$, and A_{ij} means the (i, j)-element of $\boldsymbol{A}$. $\mathrm{Tr}(\boldsymbol{A})$ is the trace of $\boldsymbol{A}$ and $\boldsymbol{A}^\top$ displays the transposed matrix of $\boldsymbol{A}$. The *Frobenius* norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is introduced as $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{A})}$.

2.2. Nonnegative matrix factorization

Given a d dimensional vector $\boldsymbol{x}$ with nonnegative entries, whose n observations are denoted as $\boldsymbol{x}_j, j = 1, 2, \ldots, n$, let input matrix be $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}_{\geq 0}$, NMF seeks to factorize $\boldsymbol{X}$ into nonnegative basis $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k] \in \mathbb{R}^{d\times k}_{\geq 0}$ and nonnegative $k \times n$ coefficient matrices $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n] \in \mathbb{R}^{k\times n}_{\geq 0}$, such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$ [22].

Given the observation $\boldsymbol{x}_i \in \mathbb{R}^d_{\geq 0}$ (ith data point), NMF decomposes it into basis $\boldsymbol{W} \in \mathbb{R}^{d\times k}_{\geq 0}$ and the representation $\boldsymbol{h}_i \in \mathbb{R}^k_{\geq 0}$, i.e.,

$$\min_{\boldsymbol{W},\boldsymbol{h}\geq 0} \sum_i z(\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{h}_i) \tag{1}$$

where $z(.)$ is a loss function.

It is clear that $\boldsymbol{h}_i$ represents the weight coefficient of the observed entries $\boldsymbol{x}_i$ on the latent vectors of $\boldsymbol{X}$ (columns of $\boldsymbol{W}$). As a result, NMF adds each different pieces of data to the basis vectors in either a linear or nonlinear way. Because of the precondition $k \ll min(d, n)$, the obtained base vectors are incomplete over the original space. In other words, the perfect approximation can only be accomplished if the inherent characteristics in $\boldsymbol{W}$ are found since this method aims to depict a high-dimensional pattern with much fewer bases. The least square error is the most common loss for NMF as follows:

$$z_2(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) = \sum_i \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|^2_F, \tag{2}$$

2.3. Related work

This section will cover the link prediction approach closely related to this work. The mainstreaming types of link prediction algorithms include similarity-based methods, maximum likelihood methods, probabilistic models, and embedding models. By assuming that the network edges are uncorrelated, maximum likelihood approaches fail to consider the relationships between nodes that are not being observed. Additionally, the probabilistic approaches take a lot of time and are not the most accurate.

The similarity-based methods are the most basic framework of link prediction techniques, which assume that two nodes are more likely to have a link when they are more similar than others. For instance, Common Neighbors (CN) [12] is a similarity-based method that considers two nodes to be similar if they have a lot of shared neighbors. The most straightforward technique to determine neighborhood overlap is to count directly. A few similarity indices were presented that took the impact of the degree of the two nodes into consideration, including the Salton Index [37], also known as cosine similarity, and the Jaccard index, which was put forth by Jaccard more than a century ago. Adamic-Adar Index (AA) was developed in [13] and it is based on sum of the inverse logarithmic degree centrality of the shared neighbors. Each node in the network was given a weight value based on its degree of common neighbors, with the general notion being that nodes with a smaller degree of common neighbors contribute more than ones with a greater degree. Resource Allocation Index (RA) [38] was introduced as a way to measure resource allocation dynamics on complex networks. *Gao et al.* [39] proposed a new similarity measure based on linear dynamical response for the task of link prediction between nodes. To improve efficiency, they devised an innovative iterative procedure to circumvent the explicit calculation of the linear dynamical response (LDR) index. Similarity-based methods suffer from high complexity and noisy data, and they are unable to learn complicated structure of networks. Therefore, latent space-based algorithms are promising alternatives that can effectively study local and global network information.

Recently, deep embedding methods have been proposed for link prediction, based on Graph Neural Network (GNN) models. These models predict links in a graph by learning node embeddings that capture the local and global structural information of the graph. GNNs use message-passing algorithms or graph-level embeddings to update node embeddings and predict the likelihood of a link between two nodes. *Zhao et al.* in [40] introduced a technique that employs Graph Neural Network and a novel data augmentation-based for link prediction. This method enhances the graph data for improved representation by generating counterfactual links. Deep Graph Auto-encoder (DGAE) [41], is a deep embedding to link prediction in attributed networks by taking advantage of the close relationship between node and edge information. This model incorporates the use of standard auto-encoders (AEs) in GAE architectures, resolving the issue of previous methods that had limited usage of shallow GAEs and variational GAEs. Despite effectiveness of these models in link prediction tasks, there are some potential flaws such as expensive computational cost, overfitting to limited number of observations, and sensitivity to hyperparameters.

One of the methods based on dimensionality reduction is the Nonnegative Matrix Factorization which can propose a latent feature learning method for link prediction. In the Nonnegative Matrix Factorization method, we have $\boldsymbol{A} \approx \boldsymbol{W}\boldsymbol{H}$, in which $\boldsymbol{A}$ is the data matrix (in the adjacency matrix), $\boldsymbol{W}$ is the base matrix or mapping matrix, and $\boldsymbol{H}$ is called the coefficient or representation matrix [22]. After completing the learning algorithm, the constructed network matrix is rebuilt by multiplying the two base and the coefficient matrices to compare with the main adjacency matrix and predict the links.

A neighborhood-based Nonnegative Matrix Factorization model is a pioneering method proposed in [42] as a solution to the link prediction problem. This approach derives latent feature components from the underlying network's global topological and local neighborhood structures. *Chen et al.* [43] use the similarity between the columns of the matrix as the scoring matrix and reconstruct the correlation between various matrix types by projecting a high-dimensional vector space to a low-dimensional latent. *Wang et al.* [23] proposed a kernel method (NMFKL) for link prediction based on Nonnegative Matrix Factorization, which may get both local and global network information via kernel mapping. Two kernel functions, which have principled interpretations for the network reconstruction, map the adjacency matrix to feature space. This model is proposed as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{K} - \boldsymbol{W}\boldsymbol{H}\|^2_F, \tag{3}$$

where $K_{i,j} = \phi(\boldsymbol{a}_i, \boldsymbol{a}_j)$, ϕ is a linear or covariance kernel, and $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are the ith and jth columns of $\boldsymbol{A}$ (adjacency matrix).

Wang et al. [44] proposed SASNMF, a unified link prediction framework that considers both internal and external auxiliary information and the graph structure, which includes node attributes and latent structural features extracted from the graph. To address the link prediction issue, two potential combinations of internal and external information are suggested and included in the model, which is defined as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}_1,\boldsymbol{H}_2} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}_1\|^2_F + \lambda_1\|\boldsymbol{S} - \boldsymbol{W}\boldsymbol{H}_2\|^2_F + \lambda_2(\|\boldsymbol{H}_1\|^2_F + \|\boldsymbol{H}_2\|^2_F), \tag{4}$$

where $\boldsymbol{A}$ is adjacency matrix and $\boldsymbol{S} = \boldsymbol{S}^A$ (Structure Similarity) or $\boldsymbol{S} = \boldsymbol{S}^X$ (Attribute Similarity).

A Graph Regularized Weighted NMF method has been introduced by [45] to combine local topology information with the weight of links for network reconstruction. GWNMF is defined as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{S} \odot (\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}^\top)\|^2_F + \lambda_1(\|\boldsymbol{W}\|^2_F + \|\boldsymbol{H}\|^2_F) + \lambda_2 \mathrm{Tr}(\boldsymbol{H}^\top \boldsymbol{L}\boldsymbol{H}), \tag{5}$$

where $\boldsymbol{S}$ is the weighted cosine similarity and $\boldsymbol{L}$ is the Laplacian matrix. This model combines two kinds of local topological information and link weights and computes the weighted similarity between local nodes using the weighted cosine similarity (WCS) approach. To collect more valuable information, the WCS score matrix was used as the indication weighted matrix. The WCS score matrix combines the graph regularization technique to capture the local structure.

Chen et al. [46] introduced a Manifold regularization and Sparse learning method based on the Robust NMF (MS-RNMF) method for link prediction. This method is proposed as follows:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{P}} &\|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|_{2,1} + \lambda_1\|\boldsymbol{C} - \boldsymbol{W}\boldsymbol{P}\|_{2,1} + \lambda_2(\|\boldsymbol{W}\|_{2,1} + \|\boldsymbol{H}\|_{2,1}) \\ &+ \lambda_3 \mathrm{Tr}(\boldsymbol{H}^\top \boldsymbol{L}\boldsymbol{H}),\end{aligned} \tag{6}$$

where $\boldsymbol{A}$ is adjacency matrix, $\boldsymbol{C}$ is the global clustering similarity with k-medoids algorithms and $\mathrm{Tr}(\boldsymbol{H}^\top \boldsymbol{L}\boldsymbol{H})$ is manifold regularization where $\boldsymbol{L}$ is the Laplacian matrix. To maintain the local and global

topology information for the network, the MS-RNMF first combines the k-medoids technique with manifold regularization. In addition, the MS-RNMF uses a $\ell_{2,1}$ regularization norm called random noise to eliminate noisy and spurious links.

Chen et al. [47] utilized the PageRank algorithm in the NMF-AP approach to determine the node's impact score and collect data on the overall network topology. Their approach also maintains the knowledge of the local network topology while calculating the link clustering coefficient score using the asymmetric link clustering approach.

Wang an Mu [48] proposed a regularized Convex NMF method, namely RC-NMF, to link prediction and community detection in the signed graphs. With graph regularization, this method groups negative-edge nodes into distinct communities and positive-edge nodes into the same communities as far as possible. RC-NMF method is represented as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{W}\boldsymbol{H}^{\top}\|_F^2 + \lambda_1 \mathrm{Tr}(\boldsymbol{H}^{\top}(\boldsymbol{D}^P - \boldsymbol{A}^P)\boldsymbol{H}) + \lambda_2 \|\boldsymbol{H}\|_1^2, \tag{7}$$

where $\boldsymbol{A}$ is signed adjacency matrix, $\boldsymbol{A}^P$ is adjacency matrix with positive links, and $\boldsymbol{D}^P$ is a diagonal degree matrix as $D_{ii}^P = \sum_i A_{ij}^P$.

Finally, *Chen et al.* have suggested a deep network reconstruction model for the link prediction problem based on deep nonnegative matrix factorization called FSSDNMF [49], which combines structural data and scattering limitations with deep matrix factorization. In order to learn the topological details of each hidden layer, they employed the common neighbor approach to calculate the similarity and transfer it to the low-dimensional multi-layer hidden space. Each hidden layer is also given a $\ell_{2,1}$ norm-based regularization to eliminate random noises.

3. Proposed model

Inspired by adversarial training and local structure preservation, this section develops an effective approach to optimize a Nonnegative Matrix Factorization model with improved generalization capabilities for link prediction. Four main factors contribute to its success: (1) It leverages a Nonnegative Matrix Factorization structure for representing the input matrix (adjacency matrix) into the hidden space to obtain a more abstract latent representation (Section 3.1); (2) An adversarial training model employs an attack approach for training an adversarial NMF. The training process employs the adversaries on the proposed model to enhance the generalization abilities (Section 3.2); (3) To preserve the local network topology in the latent embedding space, a graph regularization based on the common neighbor similarity is added to factorization. (Section 3.3); (4) It incorporates the aforementioned factors into one joint learning problem, and applies an effective alternating min–max method for optimization (Section 3.4). The structure of the LPANMF model is shown in Fig. 1.

3.1. Problem description

The network is often defined as $G = (V, E)$, where V is the node set, and E is the link set. Mathematically, the graph may be represented by the adjacency matrix $\boldsymbol{A}$ of $n \times n (n = |V|)$, where $a_{ij} = 1$ if node i is linked to node j and 0 otherwise. The diagonal entries are set to 0, indicating that each node has no connection to itself. Finding missing and nonexistent links is the goal of link prediction. Using our method, we calculate the scores for all unobserved links and quantify their probability using scores in the product matrix. The links that are more likely to exist, are displayed first, after we sort all the unseen links according to the score. Different prediction algorithms, however, could achieve different ordered lists of unobserved relationships. To test and compare the performance of the algorithms, we randomly split the existing links into a training set E^T and probe set (test set) E^P according to the proportion p, where $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. Two common criteria for evaluating link prediction accuracy are AUC [50] and Precision [51].

3.2. Basic adversarial model

Most Nonnegative Matrix Factorization methods are non-convex optimization problems; therefore, they usually get stuck at the local minimum, especially when missing data and outliers are prevalent. In this paper, we want to address this problem by including it in adversarial learning. This section introduces the Adversarial NMF (ANMF) formulation. We suppose that an attacker exists which adds an arbitrary attack $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ to the input matrix $\boldsymbol{A}$. The adversarial matrix $\boldsymbol{R}$ maximizes the Frobenius norm (least square loss) between $\boldsymbol{A}$ and the prediction $\hat{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{H}$. Because the adversary usually has limited strength, we suppose $\boldsymbol{R}$ belongs to a bounded set. As a result, ANMF can be written as

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} \max_{\boldsymbol{R}\in\mathcal{R}} \|\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{W}\boldsymbol{H}\|_F^2 \tag{8}$$

where the defined bound constraint is

$$\mathcal{R} := \{\boldsymbol{R} \in \mathbb{R}^{n\times n} : \|\boldsymbol{R}\|_F^2 \leq \epsilon, \boldsymbol{A} + \boldsymbol{R} \geq 0\}, \tag{9}$$

where constant $\epsilon > 0$ is the power of the adversary; the larger ϵ leads to a strong adversary and vice versa. We suppose that $\boldsymbol{R}$ belongs to the set of Frobenius bounded matrices, and $\boldsymbol{A}+\boldsymbol{R}$ is forced to be nonnegative so that for any fixed $\boldsymbol{R}$, the optimization in (8) corresponds to the ANMF problem with effective input matrix $\boldsymbol{A} + \boldsymbol{R}$.

In general, the problem mentioned in (8) and (9) is tough to solve. Therefore, inspired by Lagrangian duality, we utilize the relaxation [52]. The norm constraint $\|\boldsymbol{R}\|_F^2 \leq \epsilon$ in (9) is dualized with a Lagrange multiplier $\lambda > 0$; as a result, the basic cost function is

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} \max_{\boldsymbol{R}:\boldsymbol{A}+\boldsymbol{R}\geq 0} \|\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{W}\boldsymbol{H}\|_F^2 - \lambda\|\boldsymbol{R}\|_F^2 \tag{10}$$

The inner optimization is a maximization; therefore, a negative regularization term is added to (8). The interpretation of λ is dual to that of ϵ; it demonstrates the strength of the adversary. In fact, as $\lambda \to 0^+$, $\|\boldsymbol{R}\|_F^2$ will be unbounded, implying a very powerful adversary. In contrast, as $\lambda \to \infty$, the impact of regularization fades away, and $\|\boldsymbol{R}\|_F^2$ bends to zero, implying that there is no adversary [53].

The optimization in (10) is naturally divided into two sections, an inner maximization that will optimize $\boldsymbol{R}$ and an outer minimization that will optimize $\boldsymbol{W}, \boldsymbol{H}$. The inner maximization can be expressed in the same way as a minimization as

$$\boldsymbol{R}^* = \arg\min_{\boldsymbol{R}} -\|\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \lambda\|\boldsymbol{R}\|_F^2, \text{ s.t.} \quad \boldsymbol{A} + \boldsymbol{R} \geq 0. \tag{11}$$

The following shows that this minimization has a well-defined solution in a closed-form. After $\boldsymbol{R}^*$ has been obtained, the objective function can be optimized, over $\boldsymbol{W}, \boldsymbol{H} \geq 0$ as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} \|\boldsymbol{A} + \boldsymbol{R}^* - \boldsymbol{W}\boldsymbol{H}\|_F^2. \tag{12}$$

Through the new input matrix $\boldsymbol{A} + \boldsymbol{R}^*$, we can utilize Majorization-Minimization [54] to obtain solutions for $\boldsymbol{W}$ and $\boldsymbol{H}$.

3.3. Regularized adversarial model

This section will introduce a loss function insensitive to attacks and noises. Various techniques induce structural information to link prediction, including common neighbors and pathways methods. This work uses common neighbors similarity as the most straightforward and effective technique to preserve implicit topological information. Consequently, the following definition applies to the local network structure score matrix,

$$S_{i,j} = |\Gamma_i \cap \Gamma_j|, \tag{13}$$

where Γ_i is the neighbor set of node i. This similarity matrix for an unweighted network can be rewritten as follows:

$$\boldsymbol{S} = \boldsymbol{A}\boldsymbol{A}^{\top}, \tag{14}$$

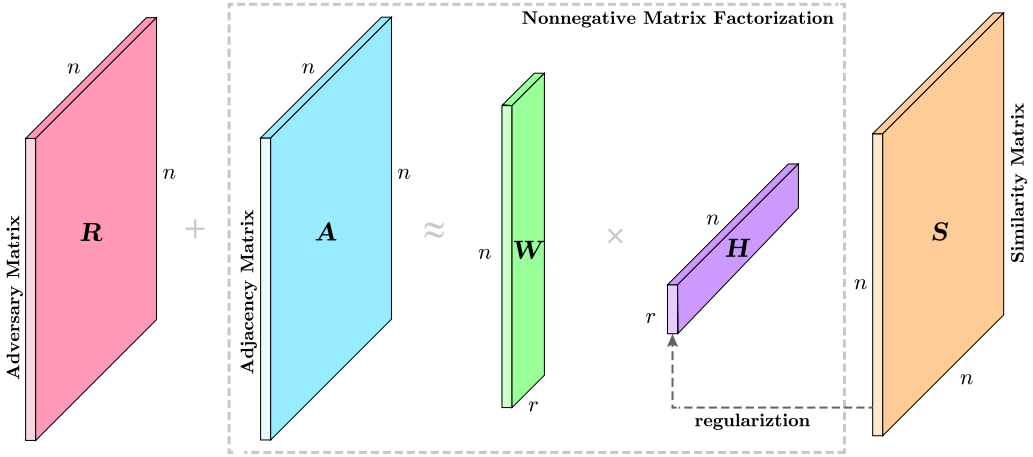


Fig. 1. Diagram of Adversarial Nonnegative Matrix Factorization with similarity regularization that decomposes adjacency matrix $\boldsymbol{A}$ (with an attack) into basis and coefficient matrices for Link prediction.

where $\boldsymbol{A}^\top$ indicates the transpose of adjacency matrix $\boldsymbol{A}$.

In this part, we incorporate the aforementioned similarity (14) in the adversarial factorization model by graph regularization to maintain the topological structure of the network. Our goal is to force similar nodes to have similar hidden features in the representation space. Given $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are feature vectors of node i and j, respectively. The vectors $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ of two nodes should be close in the latent space if their local topological structures are similar at nodes i and j. To calculate this, we utilize the Euclidean distance $d(\boldsymbol{h}_i, \boldsymbol{h}_j) = \|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2$. To reduce the sum similarity errors, we use the graph regularization as follows:

$$G = \frac{1}{2}\sum_{i}^{n}\sum_{j}^{n}\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2 S_{ij} = \mathrm{Tr}(\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top) - \mathrm{Tr}(\boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^\top) = \mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top) \tag{15}$$

where Tr() means the matrix trace, $\boldsymbol{D}$ is a diagonal matrix as $D_{i,i} = \sum_{j=1}^{n} S_{ij}$, and the laplacian matrix is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$.

By adding the graph regularization (15) to the basic model (10), we have

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}}\max_{\boldsymbol{R}}&\|\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{W}\boldsymbol{H}\|_F^2 - \lambda\|\boldsymbol{R}\|_F^2 + \alpha\mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top)\\ &\text{s.t.}\quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0\end{aligned} \tag{16}$$

Empirical investigations also reveal that matrix factorization is prone to overfitting for large latent components, which truncating the latent components significantly reduces the flexibility and capacity of the model. To prevent complicated solutions, probabilistic matrix factorization often uses Frobenius norm regularization on latent components [55]. Therefore, we will use the smoothness regularization $\mathcal{R}(\boldsymbol{W}, \boldsymbol{H}) = \|\boldsymbol{W}\|_F^2 + \|\boldsymbol{H}\|_F^2$ to prevent our model from overfitting the training set.

After making the preparations mentioned above, we are ready to present the Link Prediction by Adversarial NMF (LPANMF) model. The final objective function for LPANMF is as follows:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}}\max_{\boldsymbol{R}}&\|\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{W}\boldsymbol{H}\|_F^2 - \lambda\|\boldsymbol{R}\|_F^2 + \alpha\mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top)\\ &+ \beta(\|\boldsymbol{W}\|_F^2 + \|\boldsymbol{H}\|_F^2)\\ &\text{s.t.}\quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0,\end{aligned} \tag{17}$$

where α and β control the impact of similarity and smoothness regularizations, respectively.

3.4. Numerical solution

To solve Eq. (17), the alternating minimization (Algorithm 1) can be used, since it enables us to update the variables repeatedly until we find a workable solution. We use gradient descent to update one of the variables and fix the others in each iteration. The optimization problem is divided into several smaller, easier-to-solve sub-problems.

3.4.1. Optimizing attacker

In this part, we give specifics on how to solve $\boldsymbol{R}^*$ in (11). Here $\boldsymbol{W}$ and $\boldsymbol{H}$ matrices are considered fixed, and their product is shown as $\hat{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{H}$. As a result, the objective function can be expressed as

$$\begin{aligned}g(\boldsymbol{R}) :&= -\|\boldsymbol{A} + \boldsymbol{R} - \hat{\boldsymbol{A}}\|_F^2 + \lambda\|\boldsymbol{R}\|_F^2\\ &= \mathrm{Tr}(-\boldsymbol{A}\boldsymbol{A}^\top - \boldsymbol{R}\boldsymbol{R}^\top + \hat{\boldsymbol{A}}\hat{\boldsymbol{A}}^\top - 2\boldsymbol{A}\boldsymbol{R}^\top + 2\boldsymbol{A}\hat{\boldsymbol{A}}^\top + 2\boldsymbol{R}\hat{\boldsymbol{A}}^\top\\ &\quad + \lambda\boldsymbol{R}\boldsymbol{R}^\top).\end{aligned} \tag{18}$$

The optimal and unrealistic value for R_{ij} is ∞, demonstrating an unbounded adversary power. For $\lambda = 1$, if $\hat{A}_{ij} - A_{ij} > 0$, then the solution is $R^*_{ij} = -A_{ij}$; if $\hat{A}_{ij} = A_{ij}$, regardless of the value of R_{ij}, the function is 0; finally, if $\hat{A}_{ij} - A_{ij} < 0$, the solution is $R^*_{ij} = \infty$. According to these conditions, the convex objective function is straightforward to represent by a direct differentiation as

$$\frac{\partial g}{\partial \boldsymbol{R}} = -2\boldsymbol{R} - 2\boldsymbol{A} + 2\hat{\boldsymbol{A}} + 2\lambda\boldsymbol{R} = 0, \tag{19}$$

and

$$\lambda\boldsymbol{R} - \boldsymbol{R} = \boldsymbol{A} - \hat{\boldsymbol{A}}. \tag{20}$$

Under $\boldsymbol{A} + \boldsymbol{R} \geq 0$ condition, optimal solution for (18) is:

$$\boldsymbol{R} = \max\left\{\frac{\boldsymbol{A} - \hat{\boldsymbol{A}}}{\lambda - 1}, -\boldsymbol{A}\right\}, \tag{21}$$

where we use an element-wise max operator.

3.4.2. Optimizing learner

Upon the optimization of $\boldsymbol{R}$, the optimization over $(\boldsymbol{W}, \boldsymbol{H})$ is standard if we assume $\boldsymbol{U} := \boldsymbol{A} + \boldsymbol{R}^*$ as the effective input matrix. The updating rules are listed as follows.

• Updating rule for the basis matrix $\boldsymbol{W}$

By fixing all the matrices (except $\boldsymbol{W}$), the objective function in Eq. (17) is

$$\begin{aligned}\min_{\boldsymbol{W}}&\|\boldsymbol{U} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \beta\|\boldsymbol{W}\|_F^2 - \mathrm{Tr}(\boldsymbol{\Theta}\boldsymbol{W}^\top)\\ &= \mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^\top - 2\boldsymbol{U}\boldsymbol{H}^\top\boldsymbol{W}^\top + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{W}^\top + \beta\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{\Theta}\boldsymbol{W}^\top)\end{aligned} \tag{22}$$

To solve Eq. (22), we define a Lagrangian multiplier $\boldsymbol{\Theta}$ to force the nonnegativity on $\boldsymbol{W}$. By setting the partial derivative of (22) with respect to $\boldsymbol{W}$ to 0,

$$\boldsymbol{\Theta} = -2\boldsymbol{U}\boldsymbol{H}^\top + 2\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top + 2\beta\boldsymbol{W} \tag{23}$$

With respect to the complementary slackness condition of the Karush–Kuhn–Tucker, we have

$$\boldsymbol{\Theta} \odot \boldsymbol{W} = 0, \tag{24}$$

where $\odot$ shows the *Hadamard* product. This equation is the fixed point that the solution must guarantee convergence. By solving (24), we have the following updating rule:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{U}\boldsymbol{H}^{\top}}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\top} + \beta \boldsymbol{W}} \tag{25}$$

• Updating rule for the latent matrix $\boldsymbol{H}$

By fixing all the matrices except for $\boldsymbol{H}$, the objective function in (17) is

$$\begin{aligned} \min_{\boldsymbol{H}} & \|\boldsymbol{U} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \alpha \mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{\top}) + \beta \|\boldsymbol{H}\|_F^2 - \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{H}^{\top}) \\ &= \mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^{\top} - 2\boldsymbol{U}\boldsymbol{H}^{\top}\boldsymbol{W}^{\top} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\top}\boldsymbol{W}^{\top} + \beta \boldsymbol{H}\boldsymbol{H}^{\top}) \\ &\quad + \alpha \mathrm{Tr}(\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^{\top}) - \alpha \mathrm{Tr}(\boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^{\top}) - \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{W}^{\top}) \end{aligned} \tag{26}$$

To solve Eq. (26), we define a Lagrangian multiplier $\boldsymbol{\Phi}$ to force the nonnegativity on $\boldsymbol{H}$. By setting the partial derivative of (26) with respect to $\boldsymbol{H}$ to 0,

$$\boldsymbol{\Phi} = -2\boldsymbol{W}^{\top}\boldsymbol{U} + 2\boldsymbol{W}^{\top}\boldsymbol{W}\boldsymbol{H} + 2\alpha \boldsymbol{H}\boldsymbol{D} - 2\alpha \boldsymbol{H}\boldsymbol{S} + 2\beta \boldsymbol{H} \tag{27}$$

With respect to the complementary slackness condition of the Karush–Kuhn–Tucker, we have

$$\boldsymbol{\Phi} \odot \boldsymbol{H} = 0, \tag{28}$$

The updating rule for $\boldsymbol{H}$ is obtained, following like derivation of the update rule for $\boldsymbol{H}$,

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^{\top}\boldsymbol{U} + \alpha \boldsymbol{H}\boldsymbol{S}}{\boldsymbol{W}^{\top}\boldsymbol{W}\boldsymbol{H} + \alpha \boldsymbol{H}\boldsymbol{D} + \beta \boldsymbol{H}} \tag{29}$$

The optimization process of LPANMF is provided in Algorithm 1.

3.4.3. Stopping criteria

The optimization of $\boldsymbol{W}, \boldsymbol{H}$ is achieved by iterative minimization. This part presents the used criterion to terminate this inner minimization before updating a new $\boldsymbol{R}$. We look at the relative error of subsequent iterates and stop when it goes below a specified threshold ϵ_{in}, i.e., if $\boldsymbol{W}, \boldsymbol{H}$ shows the iterate of $\boldsymbol{W}^{(o,i)}, \boldsymbol{H}^{(o,i)}$ at the o^{th} outer iteration and i^{th} inner iteration and $\hat{\boldsymbol{A}}^{(o,i)} := \boldsymbol{W}^{(o,i)}\boldsymbol{H}^{(o,i)}$. The inner minimization will be terminated once the inner iteration step i reaches

$$\left\| \frac{\hat{\boldsymbol{A}}^{(o,i+1)} + \hat{\boldsymbol{A}}^{(o,i)}}{\hat{\boldsymbol{A}}^{(o,i)}} \right\|_F < \epsilon_{in} \tag{30}$$

The entire algorithm is terminated once the outer iteration step o reaches

$$\left\| \frac{\hat{\boldsymbol{A}}^{(o+1,i)} + \hat{\boldsymbol{A}}^{(o,i)}}{\hat{\boldsymbol{A}}^{(o,i)}} \right\|_F < \epsilon_{out} \tag{31}$$

where i is the step of the inner iteration. The algorithm finishes the inner and outer iterates once the predetermined iteration numbers, *maxIn* and *maxOut* are reached.

4. Experiments and results

In this section, the LPANMF method will be compared with four similarity-based methods and eight state-of-the-art NMF and low-rank based link prediction methods. We do three experiments in this part to assess the efficacy of our method. The first assesses overall performance of the proposed model and all comparative methods, the second confirms the convergence of the proposed model, and the third assesses the robustness of LPANMF on all datasets.

Algorithm 1 Link Prediction by Adversarial NMF (LPANMF)

Input:
$\boldsymbol{A}$: adjacency matrix of undirected network $G = (V, E)$
α, β, λ: parameters
r: dimension of latent space
maxInner and maxOuter iterations
Output:
$\hat{\boldsymbol{A}}$: score matrix of the network;

```
Divide A into training network and test network;
Initializing W and H randomly
Compute Common Neighbor similarity matrix by (14)
OuterNotConv = TRUE
while OuterNotConv (31) do
    Updating R Using (21)
    InnerNotConv = TRUE
    while InnerNotConv (30) do
        Updating W by (25)
        Updating H by (29)
        if Condition (30) holds or maxInner reached then
            InnerNotConv = FALSE
        end if
    end while
    if Condition (31) holds or maxOuter reached then
        OuterNotConv = FALSE
    end if
end while
return Â = WH
```

4.1. Evaluation measure

We use two evaluation metrics, AUC [50] and Precision [51], to estimate the link prediction method. AUC and precision criteria are complementary and should be used together to evaluate the performance of a link prediction method. These measures are defined as follows:

(1) AUC (Area Under Curve) is best understood as the likelihood that the scores of a randomly selected missing link (i.e., a link in E^P) would be greater than the scores of a randomly chosen nonexistent connection (i.e., a link in $U - E^T$). We selected nonexistent and missing links at random each time to compare their similarity ratings. The number of missing links with a score greater than the nonexistent connections is n_1, and the number of missing links with a score equal to the nonexistent links is n_2, if n independent comparisons are made, we have,

$$AUC = \frac{n_1 + 0.5 \times n_2}{n}. \tag{32}$$

(2) The Precision is defined as the ratio of the number of relevant edges picked to the total number of edges selected, supposing that we have obtained the ranking of scores of the non-observed links using an algorithm. For example, if there are M connections in the probe set E^P and we consider the top L links to be the predicted edges, then the Precision value equals

$$Precision = \frac{M}{L}. \tag{33}$$

4.2. Datasets

In this study, twelve real-world graphs from different fields, including social, biological, and technical networks [56], are utilized to assess the effectiveness of our method. The networks employed in the

Table 1
The topological characteristics of twelve networks. |V| means the number of nodes, and |E| shows the number of links. CC is the clustering coefficient. M is the max degree, $\bar{d}$ is the mean degree, S is the sparsity coefficient, respectively.

Dataset	$\|V\|$	$\|E\|$	M	CC	$\bar{d}$	S
C.elegans	297	2345	134	0.31	15	5.317
Usair97	332	2126	139	0.63	12	3.858
Metabolic	453	2040	239	0.65	9	1.988
Email-Univ	1133	5451	71	0.22	9	0.849
Bio-Sc-Gt	1715	33987	549	0.35	39	2.311
YeastL	2361	7182	66	0.13	6	0.258
Bio-Ce-Gn	2219	53683	242	0.18	48	2.18
ODLIS	2909	18246	594	0.30	11	0.431
OpenFlights	2939	30501	242	0.45	10	0.706
SciMet	3084	10413	164	0.15	6	0.219
Kohonen	4469	12731	740	0.21	5	0.127
Power-Us	4941	6594	19	0.08	2	0.054

experiments are explained in the following, and the basic statistical features are shown in Table 1.

(1) C.elegans [57]: A network simulating the worm C.elegans brain's neural network. Neurons are represented by a node, while a link shows synaptic connections. It contains 297 nodes and 2,345 links.
(2) USAir97 [58]: The network is the American aviation network, where the connections are the routes and the nodes are the airports. This dataset contains 332 nodes and 2,126 links.
(3) Metabolic [56]: A network illustrating the Escherichia coli bacteria's metabolic processes, which contains 453 nodes and 2,040 links. A directed connection is a product reaction, and a node is a metabolite.
(4) Email-Univ [56]: A sizable European research institution's email data was used to create the network, which contains 1,133 nodes and 5,451 links. All email correspondence between researchers, including arriving and outgoing, is represented by links, while a node designates the community memberships.
(5) Bio-Sc-Gt [59]: The network of biological interactions, where nodes represent genes. This dataset contains 1,715 nodes and 33,987 links.
(6) YeastL [60]: A protein–protein interaction network in budding yeast. This dataset has 2,361 nodes and 7,182 links.
(7) Bio-Ce-Gn [59]: It is gathered on the biological network, which has 53676 links and 2219 nodes. The linkages between bacteria genes and archaea direct organisms are represented by the nodes, which stand in for genes.
(8) ODLIS [56]: The network is intended to serve as a hypertext reference tool for those working in the field of library and information science, as well as for academic staff, students, and patrons of all kinds of libraries. It contains 2,909 nodes and 18,246 links.
(9) OpenFlights (Kunegis, 2017) [56]: The dataset shows flights between international airports, which contains 2,939 nodes and 30,501 links. An airport serves as a node, and a directed connection represents a flight between two airports.
(10) SciMet [56]: A graph of papers from or citing Scientometrics. It contains 3,084 nodes and 10,413 links.
(11) Kohonen [56]: This is an example network dataset produced by Garfield using the Histcitep program. This dataset contains 4,469 nodes and 12,731 links.
(12) Power-US [61]: The network of American western power. It contains 4,941 nodes and 6,594 links.

4.3. Baseline method

In this subsection, we will briefly introduce some popular similarity-based and state-of-the-art NMF-based link prediction methods, which are compared with our method in the following subsections.

4.3.1. Similarity-based methods

(1) Common Neighbors [12] uses the intersection of the common neighbors of the two nodes for computing similarity, which is denoted as

$$S_{xy}^{CN} = |\Gamma_x \cap \Gamma_y| \tag{34}$$

(2) Jaccard coefficient is in fact a normalized version of common neighbor similarity. Jaccard is denoted as

$$S_{xy}^{Jac} = \frac{|\Gamma_x \cap \Gamma_y|}{|\Gamma_x \cup \Gamma_y|} \tag{35}$$

(3) Resource Allocation (RA) [38] has been inspired from the resource allocation in a physical process. Although it takes on a similar structure to AA, RA is stricter than AA in this area, which is denoted as

$$S_{xy}^{RA} = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{k_z} \tag{36}$$

(4) Adamic-Adar (AA) [13] is used to penalize nodes with high common neighbors and reward nodes with low common neighbors. AA is denoted as

$$S_{xy}^{AA} = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{log(k_z)} \tag{37}$$

where k_z denotes the degree of node z.

4.3.2. NMF and low-rank approximation based methods

(5) Standard NMF [22] after completing the learning algorithm, by multiplying the two base matrices and the coefficient, the constructed network matrix is rebuilt to compare with the main adjacent matrix and predict the links.
(6) SASNMF [44] takes into account both the graph structure and the similarity information.
(7) GWNMF [45] integrates local topology information with link weights information with the Weighted Cosine Similarity (WCS) method.
(8) NMFKL [23] uses a linear kernelized NMF method to maintain local and global structure.
(9) LDR [39] Introduces a linear dynamical response-based similarity measure between nodes.
(10) RC-NMF is a regularized convex NMF model [48] from the perspective of the generative model to prediction links in the complex network.
(11) The MS-RNMF [46] employs k-medoids clustering and manifold regularization jointly for local and global structure preservation.
(12) The FSSDNMF [49] utilizes two deep NMF models to map the adjacency and similarity matrices to a shared space.
(13) LRNP [62] uses the fully connected graph as the basis of low-rank representation, which guarantees that each local structure in the observed graph can be reconstructed.

All the methods are set with the value of the parameters that are the most declared results in the original article. For example, in MS-RNMF implementation we set $\alpha = 0.01$, $\beta = 0.1$, $\gamma = 0.05$, and dimension of latent space $k = 70$ which are according to the paper [46].

4.4. Experiment analysis

In our research, 10% of links are removed to create the probe set E^P; the rest of them constitute the training set E^T, which are employed to generate an observable network. After applying each method, we calculate the score of each node pair in the observed graph that is not connected and rank the node pairs in the order of decreasing existence likelihood. The AUC and Precision are used as the assessment

Table 2
Comparison of the prediction performance under the *AUC* in twelve networks. The results represent separate implementations with a random training set 90% and probing set partitioning 10%. The best result is highlighted in **bold** style, while underline style indicates the second-best.

Methods	*C.elegans*	*Usair97*	*Metabolic*	*Email-Un*	*Bio-Sc-Gt*	*YeastL*	*Bio-Ce-Gn*	*ODLIS*	*OpenFlights*	*SciMet*	*Kohonen*	*Power-Us*
Jaccard	0.8007	0.9081	0.7958	0.8575	0.9407	0.6978	0.8771	0.8325	0.9598	0.8006	0.8040	0.5926
CN	0.8522	0.9446	0.9304	0.8596	0.9469	0.7396	0.8861	0.8952	0.9621	0.8032	0.8211	0.5931
RA	0.8779	**0.9677**	**0.9719**	0.8612	0.9549	0.7397	0.8914	0.9053	0.9664	0.8042	0.8263	0.5927
AA	0.8698	0.9594	0.9661	0.8618	0.9511	0.7385	0.8881	0.9050	0.9649	0.8042	0.8264	0.5929
NMF	0.8667	0.9224	0.8646	0.8751	0.9625	0.8233	0.9428	0.8883	0.9248	0.8815	0.8658	0.6731
SASNMF	0.8929	0.9385	0.8925	0.8807	0.9568	0.8250	0.9402	0.9122	0.9307	0.8900	0.8780	0.7449
GWNMF	0.6955	0.8689	0.7399	0.8128	0.8874	0.7274	0.8057	0.7960	0.8636	0.8130	0.8213	0.5416
NMFKL	0.8533	0.9136	0.9048	0.8882	0.9386	0.8111	0.8968	0.9046	0.8999	0.8949	0.8657	0.7597
LDR	0.7922	0.8886	0.8246	0.7534	0.7936	0.8137	0.7926	0.8754	0.8387	0.7845	0.8634	0.5376
RC-NMF	0.8952	0.9455	0.9143	0.8968	0.9624	0.8365	0.9461	0.9219	0.9443	0.9006	0.8870	0.7726
MS-RNMF	0.8808	0.9454	0.9169	0.8742	0.9647	0.8267	0.9418	0.9169	0.9346	0.8916	0.8819	0.7174
FSSDNMF	0.8824	0.9300	0.8820	0.8850	0.9623	0.8108	0.9365	0.8882	0.8970	0.8852	0.8918	0.6424
LRNP	0.7904	0.9429	0.8806	0.7907	0.9323	0.6918	0.8202	0.8245	0.9498	0.6732	0.7072	0.6412
LPANMF	**0.9197**	0.9549	0.9330	**0.9110**	**0.9688**	**0.8550**	**0.9533**	**0.9392**	**0.9692**	**0.9171**	**0.8989**	**0.7893**

Table 3
Comparison of the prediction performance under the *Precision* in twelve networks. The results represent separate implementations with a random training set 90% and probing set partitioning 10%. The best result is highlighted in **bold** style, while underline style indicates the second-best.

Methods	*C.elegans*	*Usair97*	*Metabolic*	*Email-Un*	*Bio-Sc-Gt*	*YeastL*	*Bio-Ce-Gn*	*ODLIS*	*OpenFlights*	*SciMet*	*Kohonen*	*Power-Us*
Jaccard	0.0199	0.0939	0.0835	0.0917	0.1377	0.0450	0.0408	0.0099	0.0044	0.0164	0.0012	0.0159
CN	0.0660	0.3615	0.1597	0.1064	0.2620	0.1076	0.0923	0.0707	0.0694	0.0529	0.0590	0.0152
RA	0.0647	0.3709	0.1695	0.0771	0.3038	0.0960	0.0853	0.0639	0.0163	0.0385	0.0437	0.0015
AA	0.0597	0.3709	0.1548	0.1083	0.2724	0.1171	0.0957	0.0686	0.0490	0.0529	0.0528	0.0091
NMF	0.1286	0.3756	0.2882	0.1385	0.4853	0.1612	**0.2996**	0.0537	0.0870	0.0933	0.0813	0.0273
SASNMF	0.1505	0.4296	0.3251	0.1349	0.4949	0.1633	0.2916	0.0720	0.0957	0.0938	0.0896	0.0273
GWNMF	0.0777	0.3404	0.0813	0.0633	0.3046	0.0496	0.1015	0.0277	0.0630	0.0144	0.0365	0.0144
NMFKL	0.0801	0.3451	0.1281	0.1394	0.2449	0.0993	0.1054	0.0824	0.0609	0.0740	0.0758	0.0273
LDR	0.0960	0.0892	0.1327	0.0073	0.0332	0.0087	0.0078	0.0357	0.0040	0.0106	0.0244	0.0009
RC-NMF	**0.1650**	0.4366	0.3153	0.1440	0.4818	0.1719	0.2935	0.0734	0.0891	0.0942	**0.0899**	0.0243
MS-RNMF	0.1529	0.4390	0.2906	0.1440	0.4981	0.1626	0.2930	0.0774	0.0957	0.0928	0.0797	0.0235
FSSDNMF	0.1508	0.4624	0.3153	0.1450	0.4794	0.1503	0.2833	0.0699	0.0833	0.0766	0.0849	**0.0379**
LRNP	0.0950	0.4170	0.1555	**0.1456**	0.3293	0.1233	0.1355	**0.1138**	**0.1577**	**0.0963**	0.0826	0.0088
LPANMF	0.1633	**0.4648**	**0.3432**	0.1303	**0.5106**	**0.1752**	0.2692	0.0913	0.1082	**0.0963**	0.0861	0.0334

measures to predict missing links in the first experiment. Tables 2 and 3 provide the experimental results. Table 2 reports the AUC value of the twelve real-world datasets, with the bolded number for each dataset (in each row) being the greatest value. In most cases, the proposed model also outperforms other methods. This result shows that, in general, the adversarial training method has performed better in optimizing factorization and predicting missing links. The poor performance of our method in two datasets is due to the high clustering coefficient of those two networks (see Table 1). The findings have shown that higher clustering coefficient lead to a lower AUC in matrix factorization methods [63]. By analyzing the data in Tables 2 and 3, we can observe that certain methods may exhibit satisfactory performance in one criterion, but may not perform well in the other criterion. For example, the LRNP approach has comparable Precision results for some datasets, but it has poor performance when evaluated using the AUC measure. In contrast, the proposed method demonstrates high AUC measure scores for most datasets and achieves acceptable Precision ranks, either first or second-best. Consequently, we can deduce that the LPANMF model has the properties of a universal network reconstruction model based on these findings. Specifically, in AUC, proposed method achieves an improvement over the second-best method by 2.7%, 1.6%, 0.4%, 2.2%, 0.8%, 1.9%, 0.3%, 1.8%, 0.8%, 2.2% for the datasets *Celegans*, *Email-univ*, *Bio-Sc-Gt*, *YeastL*, *Bio-Ce-Gn*, *ODLIS*, *OpenFlight*, *SciMet*, *Kohonen* and *Power-Us*, respectively. Table 3 shows the Precision value of the twelve datasets, with the bolded number for each network (in each column) being the greatest value. In Precision, LPANMF method has a remarkable improvement over the second-best method by 0.5%, 5.6%, 2.5% and 1.9% for the datasets *USAir97*, *Metabolic*, *Bio-Sc-Gt* and *YeastL*, respectively.

4.5. Parameter analysis

The impact of parameters on the suggested model is examined in this subsection. To thoroughly study the three mentioned hyperparameters, we evaluated them on all datasets through a grid-search strategy. According to loss function (17), LPANMF has three parameters, namely λ, α, and β, which correspond to the attack intensity, similarity regularization, and Frobenius regularization terms, respectively. In addition, we have examined the number of dimensions of latent k and finally inner and outer iteration as side parameters in our algorithm. In the pre-training, we first examined a broad range of parameters value and then found the appropriate range of these. We varied k as $\{10, 20, 30, \ldots, 100\}$, α as $\{0, 0.05, 0.1, \ldots, 0.95, 1\}$, β as $\{0, 0.1, 0.2, \ldots, 1.9, 2\}$, λ as $\{1.1, 1.2, 1.3, \ldots, 4.9, 5\}$, inner and outer iterations as $\{20, 40, \ldots, 120\}$. In the experiment, two parameters are fixed, and the effects of the other four are examined. For example, we fixed inner-iteration=80 and outer-iteration=20 and checked the effects of other parameters.

4.5.1. Impact of dimension of latent space k

We focus on how to specify the dimension of latent space k in an undirected network, The k has a direct impact on the precision and computational cost of our proposed model. If k is too small, it will reduce the algorithm prediction accuracy for some datasets; if k is too large, it will increase the time complexity of the algorithm. Therefore, it is very important to select the optimal k value. Fig. 2 reports the experimental results on nine datasets. To study the effectiveness of the dimension of latent space k uniquely, we ignored the other components by setting α and β parameters to zero and λ=1000. So that according to formula (21), if the value of λ is high, attacker effect is very low and almost zero.

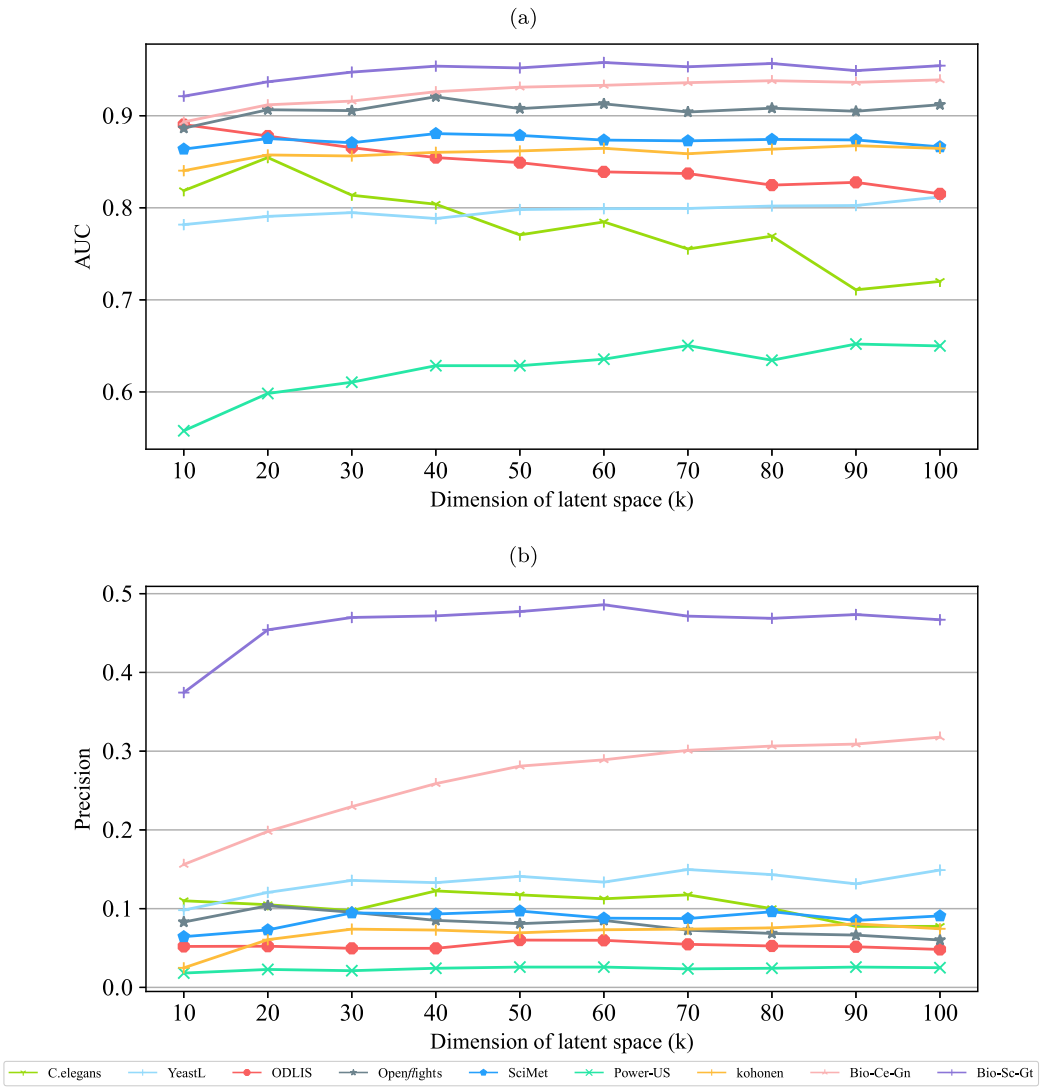


Fig. 2. The performances on nine real-world networks using different dimensions of latent space k. The k value ranges from 10 to 100. (a) Results based on AUC measure (b) Results based on Precision measure.

It is clear that, the performance changes gradually with the change of k. For some complex datasets, it is necessary to increase the capacity of the model by increasing the value of k to obtain a more accurate approximation. However, in some datasets, k must be limited to the rank of the input matrix. For example, for *C.elegans*, the value of k=20 and for *Power-Us*, the value of k=100 have the best AUCs.

4.5.2. Impact of attack parameter λ

In this subsection, to study the effectiveness of the adversarial attacking uniquely, we ignored the other two components by setting α and β parameters to zero. From Fig. 3, LPANMF switches back to conventional NMF when λ becomes large; because the strength of the attack is diminished. In fact, for high values of $lambda$, LPANMF performs similarly to basic NMF. According to Fig. 3(a), the AUC of LPANMF converges to a constant value as $\lambda \to \infty$. Similar behavior can be seen in the Precision measure (Fig. 3(b)). According to our observations, the optimal choice of $lambda$ is influenced by the size of the input matrix $\boldsymbol{A}$. For AUC measure, values $\lambda < 1.8$ are considered a heavy attack, so the results are not suitable; it can be seen that the values $1.9 < \lambda < 2.4$ is ideal for different datasets. For Precision measure, the values $2.8 < \lambda < 4.2$ is suitable.

4.5.3. Impact of α and β

This section analyzes the influence of two remain parameters by fixing the optimal k and λ parameters derived from Sections 4.5.1 and 4.5.2. According to (17), LPANMF has two main parameters, namely α and β, which correspond to the similarity and the Frobenius regularization terms, respectively. Each graph has 441 assessments corresponding to 21 unique αs and 21 distinct βs. In Fig. 4, each graph has a bar that shows the best and lowest values found on the relevant dataset for each measure, which is represented by a spectrum of colors. Subsequently, warm colors (e.g., *orange* to *yellow*) represent high AUC amounts and cold colors (e.g., *blue* to *dark-blue*) represent low amounts for the measure.

4.6. Robustness evaluation

In this section, we evaluate the robustness of LPANMF so that by randomly eliminating a portion of the links in the original network, we change the network sparsity. Therefore, we repeat the evaluation on the training sets with 30% to 90% observation rates. The results of the experiment are reported in Fig. 5. Some analyzes are listed below:

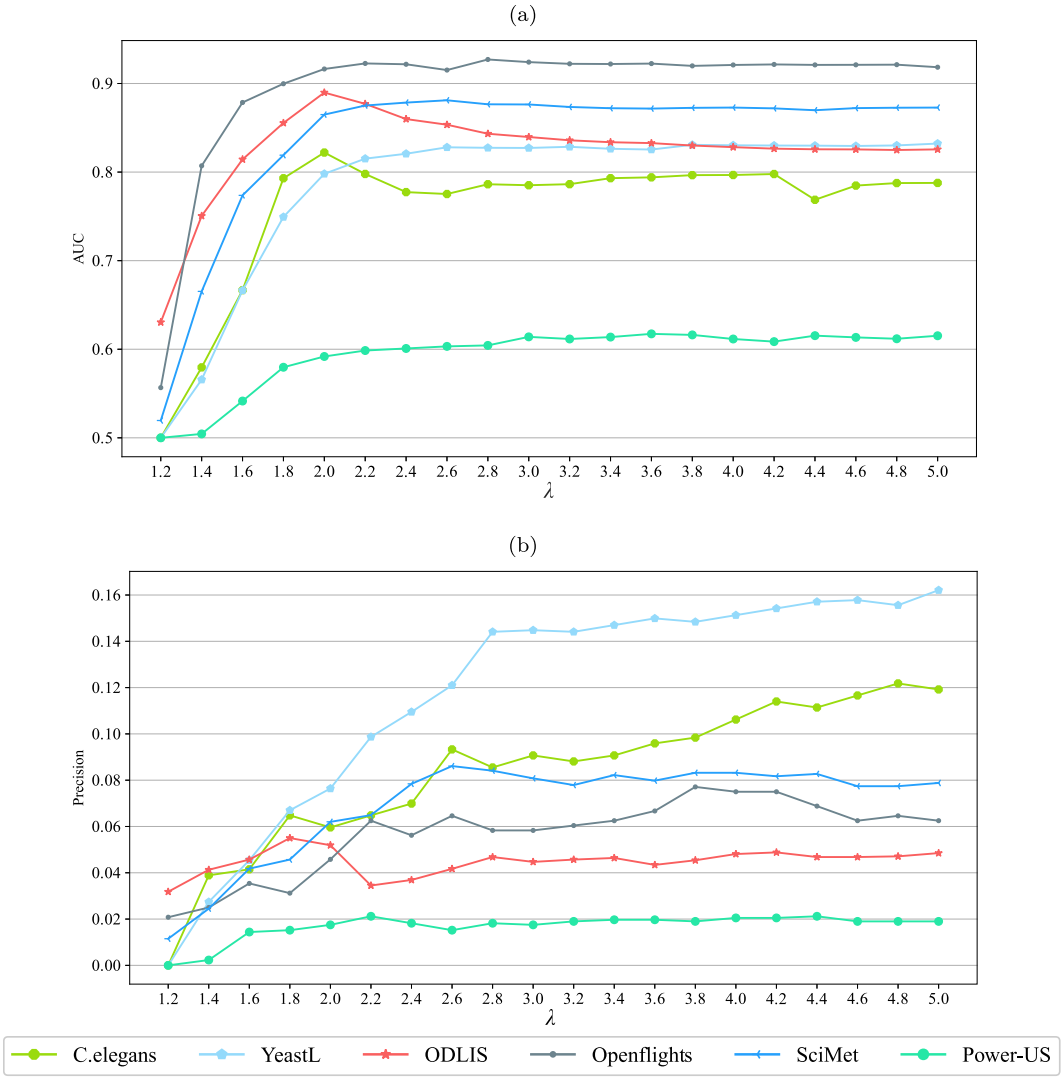


Fig. 3. Plots of (a) AUC and (b) Precision against different attacks (λ) on the six datasets when $\alpha, \beta = 0$.

- In Fig. 5, the curve of LPANMF is placed above the curve of compared methods in most cases, which indicates the robustness of the proposed method is better than other predictors under different ratios of the training set.
- In Fig. 5, the number of missing links rises as the percentage of the training set reduces from 90% to 30%. In this case, LPANMF continues to perform better than other methods when there is a limited number of seen links. Given the sparse nature of networks in the real world, this advantage is crucial.

4.7. Convergence analysis

Fig. 6 illustrates the Convergence of the objective function of LPANMF with train error on the *C.elegans* and *Power-Us* datasets. After every 80 minimizing iterations, an attack is applied to the model. We can see a long-term non-monotonically behavior for train loss because the Majorization-Minimization property is present in LPANMF, over $\boldsymbol{R}$ and factors. As you can see, slight fluctuations are observed in the train loss that indicates the effect of the attack on the model, which can increase the robustness of the model. It is worth noting that despite gaining a higher loss value on the train set, the prediction on the probe set is superior. This superiority is owing to the Intrinsic generalization of the proposed adversarial training in LPANMF.

5. Conclusion

This paper proposes a novel adversarial graph embedding framework based on Nonnegative Matrix Factorization for link prediction problem. We argue that most of the current embedding methods overlook the data distributions of the latent representation, which causes them to suffer from sparsity in real-world networks. We proposed an adversarial training algorithm to enhance generalization capabilities and increase model robustness by adding a bounded adversary matrix to the input matrix. We also use an additional term that regularizes the latent representations and enforces the latent embedding to be matched with the local topological structure. The adversarial procedure is jointly optimized with a graph regularization to extract a more robust representation. Experimental results show that the proposed method outperforms baselines in the link prediction task in terms of AUC and Precision.

There are several directions for future work. The first direction would be to extend our model to a deep matrix factorization for extracting multi-layer structures from complex networks. We will study

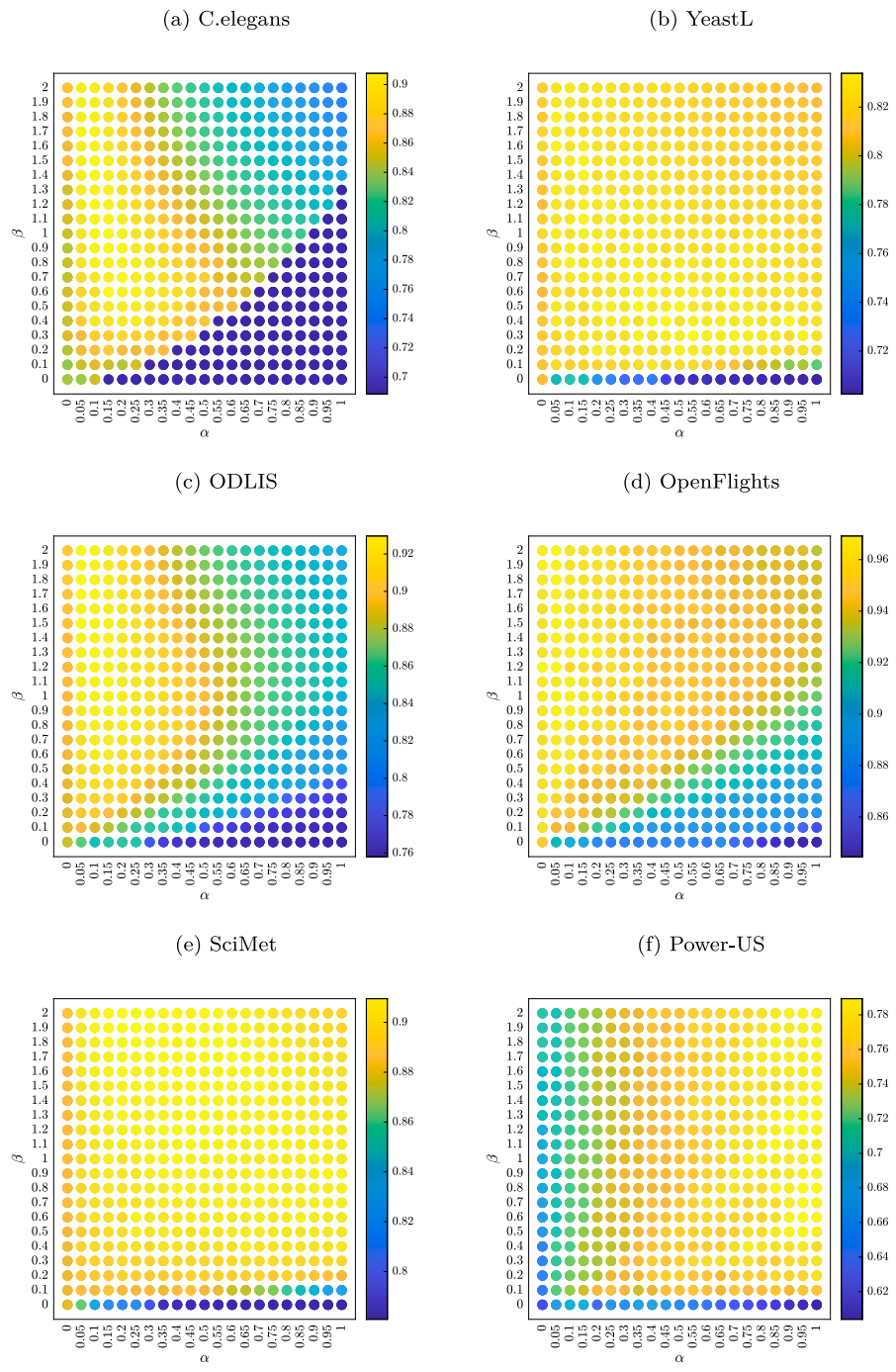


Fig. 4. Parameter analysis of Adversarial NMF (ADNMF) on six datasets for AUC measure. α is the similarity parameter and β is the frobenius norm parameter.

the effect of other types of adversarial attacks to obtain more powerful adversarial examples. We will also investigate how to incorporate context information, such as node attributes, into the proposed model to enhance the network reconstruction.

CRediT authorship contribution statement

Reza Mahmoodi: Data curation, Investigation, Methodology, Software, Visualization, Writing – original draft. **Seyed Amjad Seyedi:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – review & editing. **Fardin Akhlaghian Tab:** Conceptualization, Formal analysis, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft. **Alireza Abdollahpouri:** Investigation, Project administration, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

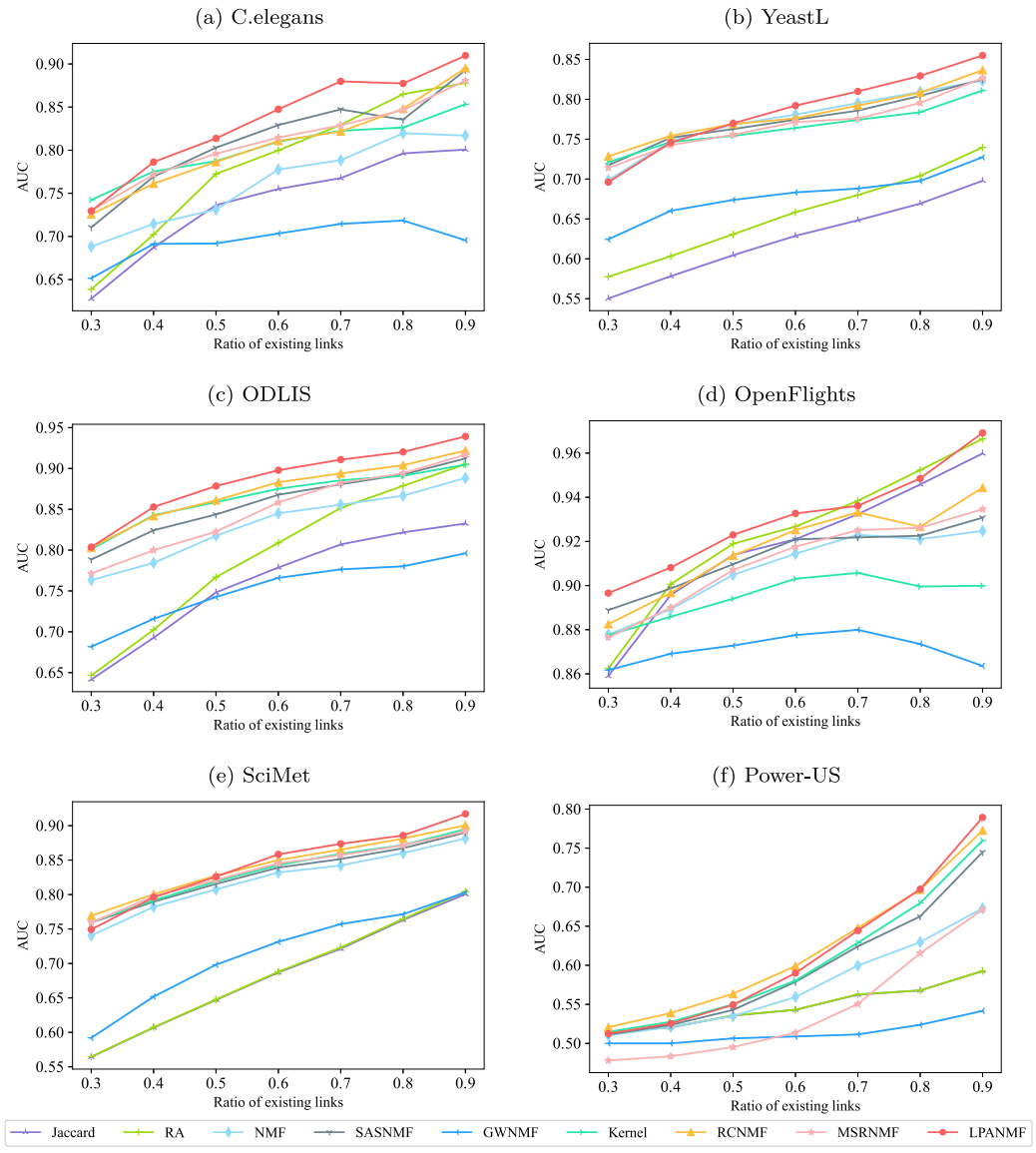


Fig. 5. AUC of six graphs under different ratio of existing links (ranges from 30% to 90%). (a)–(f) show different graphs: *C.elegans*, *YeastL*, *ODLIS*, *Openflights*, *SciMet*, *Power-US*. Besides our method (LPANMF), we further compare proposed model with eight methods (Jaccard, AA, NMF, SASNMF, GWNMF, NMFKL, RCNMF, MSRNMF). Each data point is the maximum values obtained from the runs.

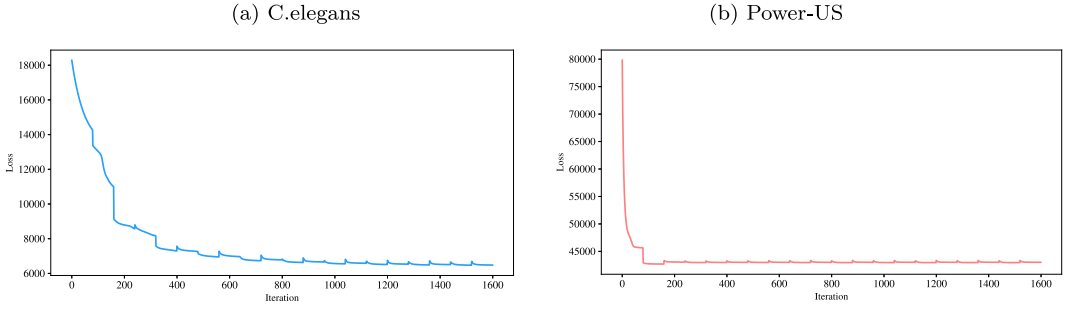


Fig. 6. Convergence of Adversarial NMF (LPANMF) on the (a) *C.elegans* and (b) *Power-US* datasets. After every 80 minimizing iterations, an attack is applied to the model.

References

[1] L. Amaral, J. Ottino, Complex networks. Augmenting the framework for the study of complex systems, Eur. Phys. J. B 38 (2004) 147–162.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: Structure and dynamics, Phys. Rep. 424 (4) (2006) 175–308.

[3] L. Getoor, Link mining: A new data mining challenge, SIGKDD Explor. Newsl. 5 (1) (2003) 84—89.

[4] F. Xie, Z. Chen, J. Shang, X. Feng, J. Li, A link prediction approach for item recommendation with complex number, Knowl.-Based Syst. 81 (2015) 148–158.

[5] E. Bütün, M. Kaya, Predicting citation count of scientists as a link prediction problem, IEEE Trans. Cybern. 50 (10) (2020) 4518–4529.

[6] C.V. Cannistraci, G. Alanis-Lobato, T. Ravasi, From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks, Sci. Rep. 3 (1) (2013) 1–14.

[7] G. Chen, Z.-P. Liu, Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data, Bioinformatics 38 (19) (2022) 4522–4529.

[8] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, C. Piccardi, Link prediction in criminal networks: A tool for criminal intelligence analysis, PLoS One 11 (4) (2016) e0154244.

[9] B. Taskar, M.-F. Wong, P. Abbeel, D. Koller, Link prediction in relational data, in: Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS '03, MIT Press, Cambridge, MA, USA, 2003, pp. 659—666.

[10] N.N. Daud, S.H. Ab Hamid, M. Saadoon, F. Sahran, N.B. Anuar, Applications of link prediction in social networks: A review, J. Netw. Comput. Appl. 166 (2020) 102716.

[11] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (6) (2011) 1150–1170.

[12] M.E.J. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2001) 025102.

[13] L.A. Adamic, E. Adar, Friends and neighbors on the web, Social Networks 25 (3) (2003) 211–230.

[14] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.

[15] K. Yu, W. Chu, S. Yu, V. Tresp, Z. Xu, Stochastic relational models for discriminative link prediction, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, Vol. 19, MIT Press, 2006.

[16] L. Getoor, B. Taskar, Probabilistic entity-relationship models, PRMs, and plate models, in: Introduction to Statistical Relational Learning, 2007, pp. 200–238.

[17] L. Getoor, N. Friedman, D. Koller, A. Pfeffer, Learning probabilistic relational models, in: Relational Data Mining, Springer, Berlin, Germany, 2001, pp. 307–335,

[18] R. Guimerá, M. Sales-Pardo, L.A.N. Amaral, Classes of complex networks defined by role-to-role connectivity profiles, Nat. Phys. 3 (2007) 63–69.

[19] A. Clauset, C. Moore, M.E. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (7191) (2008) 98–101.

[20] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[21] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[22] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[23] W. Wang, Y. Feng, P. Jiao, W. Yu, Kernel framework based on non-negative matrix factorization for networks reconstruction and link prediction, Knowl.-Based Syst. 137 (2017) 104–114.

[24] Z. Shajarian, S.A. Seyedi, P. Moradi, A clustering-based matrix factorization method to improve the accuracy of recommendation systems, in: 2017 Iranian Conference on Electrical Engineering, (ICEE), 2017, pp. 2241–2246.

[25] R. Abdollahi, S. Amjad Seyedi, M. Reza Noorimehr, Asymmetric semi-nonnegative matrix factorization for directed graph clustering, in: 2020 10th International Conference on Computer and Knowledge Engineering, (ICCKE), 2020, pp. 323–328.

[26] S.A. Seyedi, P. Moradi, F.A. Tab, A weakly-supervised factorization method with dynamic graph embedding, in: 2017 Artificial Intelligence and Signal Processing Conference, (AISP), 2017, pp. 213–218.

[27] N. Gillis, Nonnegative Matrix Factorization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020.

[28] N. Salahian, F.A. Tab, S.A. Seyedi, J. Chavoshinejad, Deep autoencoder-like nmf with contrastive regularization and feature relationship preservation, Expert Systems with Applications 214 (2023) 119051.

[29] J. Chavoshinejad, S.A. Seyedi, F. Akhlaghian Tab, N. Salahian, Self-supervised semi-supervised nonnegative matrix factorization for data clustering, Pattern Recognition 137 (2023) 109282.

[30] A.K. Menon, C. Elkan, Link prediction via matrix factorization, in: D. Gunopulos, T. Hofmann, D. Malerba, M. Vazirgiannis (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 437–452.

[31] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572.

[32] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network, 2016, arXiv preprint arXiv:1609.03126.

[33] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015, arXiv preprint arXiv:1511.06434.

[34] A. Sinha, H. Namkoong, R. Volpi, J. Duchi, Certifying some distributional robustness with principled adversarial training, 2017, arXiv preprint arXiv:1710.10571.

[35] F. Farnia, J.M. Zhang, D. Tse, Generalizable adversarial training via spectral normalization, 2018, arXiv preprint arXiv:1811.07457.

[36] T. Cai, V.Y. Tan, C. Févotte, Adversarially-trained nonnegative matrix factorization, IEEE Signal Process. Lett. 28 (2021) 1415–1419.

[37] G. Salton, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[38] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, B.-Q. Yin, Power-law strength-degree correlation from resource-allocation dynamics on weighted networks, Phys. Rev. E 75 (2007) 021102.

[39] H. Gao, J. Huang, Q. Cheng, H. Sun, B. Wang, H. Li, Link prediction based on linear dynamical response, Physica A 527 (2019) 121397.

[40] T. Zhao, G. Liu, D. Wang, W. Yu, M. Jiang, Learning from counterfactual links for link prediction, in: International Conference on Machine Learning, PMLR, 2022, pp. 26911–26926.

[41] X. Wu, Q. Cheng, Stabilizing and enhancing link prediction through deepened graph auto-encoders, in: IJCAI: Proceedings of the Conference, Vol. 2022, NIH Public Access, 2022, pp. 3587–3593.

[42] Y. Zhao, S. Li, C. Zhao, W. Jiang, Link prediction via a neighborhood-based nonnegative matrix factorization model, in: J. Mu, Q. Liang, W. Wang, B. Zhang, Y. Pi (Eds.), The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems, Springer International Publishing, Cham, 2015, pp. 603–611.

[43] B. Chen, F. Li, S. Chen, R. Hu, L. Chen, Link prediction based on non-negative matrix factorization, PLOS ONE 12 (8) (2017) 1–18.

[44] W. Wang, M. Tang, P. Jiao, A unified framework for link prediction based on non-negative matrix factorization with coupling multivariate information, PLOS ONE 13 (11) (2018) 1–22.

[45] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Graph regularization weighted non-negative matrix factorization for link prediction in weighted complex network, Neurocomputing 369 (2019) 50–60.

[46] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Robust non-negative matrix factorization for link prediction in complex networks using manifold regularization and sparse learning, Physica A 539 (2020) 122882.

[47] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Nonnegative matrix factorization for link prediction in directed complex networks using PageRank and asymmetric link clustering information, Expert Syst. Appl. 148 (2020) 113–290.

[48] J. Wang, R. Mu, A regularized convex nonnegative matrix factorization model for signed network analysis, Soc. Netw. Anal. Min. 11 (1) (2021) 1–12.

[49] G. Chen, H. Wang, Y. Fang, L. Jiang, Link prediction by deep non-negative matrix factorization, Expert Syst. Appl. 188 (2022) 115991.

[50] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.

[51] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. (TOIS) 22 (1) (2004) 5–53.

[52] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[53] S.A. Seyedi, F.A. Tab, A. Lotfi, N. Salahian, J. Chavoshinejad, Elastic adversarial deep nonnegative matrix factorization for matrix completion, Inform. Sci. 621 (2023) 562–579.

[54] D.R. Hunter, K. Lange, Quantile regression via an MM algorithm, J. Comput. Graph. Statist. 9 (1) (2000) 60–77.

[55] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, Vol. 20, Curran Associates, Inc., 2007.

[56] R.A. Rossi, N.K. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 4292–4293.

[57] J. Duch, A. Arenas, Community identification using extremal optimization, Phys. Rev. E 72 (2005) 027104.

[58] V. Colizza, R. Pastor-Satorras, A. Vespignani, Reaction–diffusion processes and metapopulation models in heterogeneous networks, Nat. Phys. 3 (4) (2007) 276–282.

[59] A. Cho, J. Shin, S. Hwang, C. Kim, H. Shim, H. Kim, H. Kim, I. Lee, WormNet v3: A network-assisted hypothesis-generating server for caenorhabditis elegans, Nucleic Acids Res. 42 (W1) (2014) W76–W82.

[60] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, et al., Topological structure analysis of the protein–protein interaction network in budding yeast, Nucleic Acids Res. 31 (9) (2003) 2443–2450.

[61] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world'networks, Nature 393 (6684) (1998) 440–442.

[62] L. Chai, L. Tu, X. Yu, X. Wang, J. Chen, Link prediction and its optimization based on low-rank representation of network structures, Expert Syst. Appl. 219 (2023) 119680.

[63] O.F. Robledo, X.-X. Zhan, A. Hanjalic, H. Wang, Influence of clustering coefficient on network embedding in link prediction, Appl. Netw. Sci. 7 (1) (2022) 1–20.