Multimedia Tools and Applications (2023) 82:3745–3768
https://doi.org/10.1007/s11042-022-12943-8

Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks

Elahe Nasiri[1] · Kamal Berahmand[2] · Yuefeng Li[2]

Received: 19 April 2021 / Revised: 6 January 2022 / Accepted: 10 March 2022 /
Published online: 15 July 2022

Abstract

Link prediction is one of the most widely studied problems in the area of complex network analysis, in which machine learning techniques can be applied to deal with it. The biggest drawback of the existing methods, however, is that in most cases they only consider the topological structure of the network, and therefore completely miss out on the great potential that stems from the nodal attributes. Both topological structure and nodes' attributes are essential in predicting the evolution of attributed networks and can act as complements to each other. To bring out their full potential in solving the link prediction problem, a novel Robust Graph Regularization Nonnegative Matrix Factorization for Attributed Networks (RGNMF-AN) was proposed, which models not only the topology structure of networks but also their node attributes for direct link prediction. This model, in particular, combines two types of information, namely network topology, and nodal attributes information, and calculates high-order proximities between nodes using the Structure-Attribute Random Walk Similarity (SARWS) method. The SARWS score matrix is an indicator structural and attributed matrix that collects more useful attributed information in high-order proximities, whereas graph regularization technology combines the SARWS score matrix with topological and attribute information to collect more valuable attributed information in high-order proximities. Furthermore, the RGNMF-AN employs the $\ell_{2,1}$-norm to constrain the loss function and regularization terms, effectively removing random noise and spurious links. According to empirical findings on nine real-world complex network datasets, the use of a combination of

✉ Kamal Berahmand
kamal.berahmand@hdr.qut.edu.au

Elahe Nasiri
el.nasiri@azaruniv.ac.ir

Yuefeng Li
y2.li@qut.edu.au

[1] Department of Information Technology and Communications, Azarbaijan Shahid Madani University, Tabriz, Iran

[2] School of Computer Science, Faculty of Science, Queensland University of Technology (QUT), Brisbane, Australia

attributed and topological information in tandem enhances the prediction performance significantly compared to the baseline and other NMF-based algorithms.

Keywords Complex network · Link prediction · Nonnegative matrix factorization · Attributed network

1 Introduction

One of the most important tasks of complex network analysis is the task of identifying missing links or predicting future links in the network, using the information obtained from the current network, i.e., structural features and/or nodal attributes, commonly known as link prediction [32, 33]. The link prediction task has been applied to solve problems in various fields; therefore, it has been a hot topic for researchers from different disciplinary backgrounds. Examples include but are not limited to recommender systems in commercial applications [12], protein-protein interactions (PPI) in biological networks [48], collaboration prediction in co-authorship networks [50], etc.

Most link prediction methods are based on the topological structure of the networks. While the effectiveness of these methods has been established, they still suffer from neglecting a very important source of information, i.e., features or attributes of nodes in the network. In many real-world networks, each node is associated with an informative set of features. These types of complex networks are called attributed networks [4, 17]. A famous example of these networks would be collaboration networks, in which every researcher has a specific research background and profile that influences the interactions between them [67]. The importance of these features has been studied in the concept of homophily, and many types of research have emphasized the tendency of similar individuals to interact with each other [18, 19]. Hence, the process of link formation is highly affected by homophily [2, 71].

Generally, link prediction approaches are classified into two main categories: similarity-based and learning-based techniques. The first category calculates the probability of link existence based on a similarity score that is assigned to the node pairs. The similarity score is calculated based on the topological structure of the network or nodal attributes [32]. The learning-based methods are divided into three sub-categories: The first sub-category, i.e., feature-based classification, treats the link prediction task as a binary classification problem in which a pre-defined set of features are used as the input of the classifier, and the output of 1/ 0 indicates link presence/ absence [63]. The second sub-category, i.e., probabilistic models, assumes that the probability of a link existing between any pair of nodes depends on a set of parameters. Therefore, an optimization function is defined based on those parameters, and once the right set of values is learned for the parameters, the conditional probability of the link's existence is calculated for any pair of nodes [32]. The last sub-category belongs to the dimensionality reduction-based groups.

Most learning-based methods suffer from the high computational complexity that is due to the large size of the networks. Therefore, recently the dimensionality reduction-based methods have been the focus of many researchers. There are two main approaches to deal with the challenges of high dimensionality. The first approach is using embedding techniques which tries to map the nodes to a lower dimension space, such that their structural and non-structural properties are maximally preserved. Embedding techniques have been developed to be applicable for many network analysis tasks such as community detection [5], link prediction

[6, 26, 47], node classification [1, 8], and recommender systems [22]. Overall, the main approaches for learning network embedding are classified into Matrix Factorization, Deep Learning, Edge Reconstruction, Graph Kernel, and Generative Model [10]. Recently, many studies have been conducted to apply embedding approaches to attributed networks and take advantage of the node attributes to obtain more accurate embedding vectors [17, 36]. The second approach is called matrix factorization, which receives a matrix, e.g., the adjacency matrix of the current network, as input and decomposes it into two lower-dimension matrices, i.e., a base matrix and a coefficient matrix, such that their product is as close as possible to the original matrix [32]. To obtain the list of potential links, these matrices are used in a supervised or unsupervised framework.

Matrix factorization methods consist of various categories, such as singular value decomposition (SVD) [24], principal component analysis (PCA) [61], and independent component analysis (ICA) [68]. However, for the task of link prediction, the most commonly used matrix factorization based method is non-negative matrix factorization (NMF), which decomposes an original matrix ($X \in R^{n \times n}$) into the base matrix ($W \in R^{n \times k}$) and the coefficient matrix ($H \in R^{n \times k}$) such that all the elements in the three matrices are greater than or equal to zero [60], which leads to a situation also known as an additive parts-based representation of the original data. This term refers to the fact that only an additive combination of the original data is possible [21, 34].

NMF-based methods have been applied to many different problems, such as computer vision [56], dimensionality reduction [42, 45, 57, 58], and data mining [35]. In addition, NMF-based methods have also been applied to the link prediction problem in many different kinds of networks such as temporal networks [20], weighted networks [14], directed networks [15], and in order to improve their performance, they take advantage of the specific properties of those networks. For example, Chen et al. [14] utilized the edge weights to improve prediction accuracy, while in [15], the direction of links was used in the process of decomposition of the original matrix. Since the effectiveness of this method has been established, due to the high potential of NMF-based methods, in the present work, we decided to benefit from the NMF-based method's advantages in the attributed networks to solve the link prediction problem. After learning the base and coefficient matrix and multiplying them to find the reconstructed matrix, a score is obtained for every pair of nodes, which determines the probability of there being a link between the node pairs.

In this article, a new robust version of the graph regularization non-negative matrix factorization model combined with graph regularization and structure-attribute similarity is presented to capture semi-local topology structure and attribute information to solve the problem of link prediction in attributed networks. In order to obtain all of the link weight information from the original network, the Biased Local Rand Walk is used to measure the semi-local proximity and attribute similarity between local nodes; then, the graph regularization technology is combined with SARWS to explore topology information. In addition, the $\ell_{2,1}$-norm is used to remove random noise and spurious links. Ultimately, a unified link prediction model (GRNMF-AN) is suggested, and in order to learn the parameters of GRNMF-AN, multiplicative updating rules are used. The authors use nine real-world attributed networks and four evaluation metrics to assess the feasibility of the proposed model; the experimental findings indicate that our model outperforms conventional algorithms.

In Section 2, a brief overview of previous works in the field of link prediction will be provided. Then, in Section 3, some of the work's preliminaries are discussed, such as the definition of structural-attributed similarity, NMF variants, and the proposed algorithm

(GRNMF-AN). Section 4 covers the experimental analysis, which involves evaluating the performance of GRNMF-AN compared to state-of-the-art methods and analyzing the effect of the parameter, and Section 5 presents the effectiveness of the GRNMF-AN in link prediction performance. Finally, Section 6 presents the conclusion and future work and Section 5 presents the conclusion.

2 Related work

In this section, we will briefly introduce some of the most important methods that have been proposed to solve the problem of link prediction. Based on published surveys [32] on link prediction in complex networks, the main link prediction methods are classified into two main classes: similarity-based methods, which are the simplest way to predict missing or new links, and learning-based methods, which require more computational resources. In the following, we will discuss each class and provide a few examples of them.

Similarity-based methods calculate the probability of a link between a pair of nodes based on a pre-defined similarity measure and then pick the L links which have the highest similarities. The similarity score between a non-connected node pair is calculated using the topological structure of the network. In general, we can use local, global, and semi-local scores to calculate the similarity between pairs of nodes. Local-based scores use only the information from the local structure of the nodes to obtain the most similar node pairs. Global methods consider the whole structure of the network for calculating the similarity of a node pair. Therefore, they suffer from high computational complexity while benefiting from global information. Semi-local similarities have been able to achieve a trade-off between these two methods. They use more information compared to local indexes, and, unlike global indexes, they do not require high time and computational resources [7, 49]. To improve the performance of similarity-based methods, some researchers proposed benefiting from the side information associated with each node, also known as node attributes. For example, Muniz et al. [46] proposed a framework to combine structural similarity, obtained by common neighbors, node attributes, obtained by profile and gender information, and also temporal information, obtained by time attributes, and get the probability of a link existing between a pair of nodes.

Learning-based methods use structural and non-structural properties of networks as inputs for a machine learning framework to learn the probability of a link existing between a pair of nodes. These methods are divided into feature-based classification, probabilistic models, and dimensionality reduction methods [32]. The first class, i.e., feature-based classification, considers the link prediction problem as a binary classification task and applies a learning model such as a decision tree, neural network, and support vector machine to predict the labels for each pair of nodes. Structural and non-structural features can be used as input for classification [53]. The main challenge in this approach is how to define the right set of features to get the best results. Keikha et al. [30] proposed a link prediction framework based on deep learning, which extracts the best set of features from structural information and nodal attributes and eliminates the need for manual feature engineering. Structural information includes information from the local and global structure of nodes.

The second class, i.e., the probabilistic model, defines an objective function with a set of parameters and tries to obtain the current structure of the network-based optimization function with the right values for the parameters. When the right set of parameters are learned, the probability of a link existing between a pair of nodes is calculated using the conditional probability $P(A_{ij} = 1 | \theta)$, the probabilistic [62] and maximum likelihood [27] based methods have been widely studied, and several methods based on them have been proposed to solve the link prediction problem. Their most challenging aspect is that they require parameter tuning, which is computationally expensive and very time-consuming. Therefore, it is not applicable to large-scale networks. The last class, i.e., dimensionality reduction-based methods, has been the focus of attention by lots of researchers due to their applicability to large-scale networks. In general, they consist of two main approaches, i.e., embedding techniques and matrix factorization.

Embedding-based methods map the graph's information to a low-dimensional space in which the structural information of the graph and its components, e.g., nodes, edges, communities, etc., are preserved as much as possible. Some comprehensive surveys [10] provide a detailed study of graph embedding techniques, which are great references for embedding approaches and their applications. Here we summarize the three main approaches to graph embedding techniques. The first category is matrix factorization, in which the graph is represented in the form of a matrix, and then the matrix is factorized, and the embedding vectors for nodes are obtained. Menon and Elkan [43] are the pioneers of this technique by proposing a matrix factorization framework for graph structure and using it to solve the link prediction method for the first time. Other examples of this category include the Hope method [51], which is uses $|| \mathrm{S}-\mathrm{Y}_\mathrm{s}\ \mathrm{Y}_\mathrm{t}||_\mathrm{F}^2$ as an optimization function and tries to minimize it. S is the similarity matrix in this setting, which can be defined using several similarity measures, e.g., common neighbors, the Adamic-Adar index, etc.

Another approach to learning the embedding of graphs uses Deep Learning methods. Most of these methods use the autoencoder structure to achieve a dimensionality reduction on the graph structure. For example, SDNE [65] uses a deep autoencoder to maintain the graph's first and second-order proximity. DNGR [11] is another example of using auto-encoder networks, but its input consists of a positive pointwise mutual information (PPMI) matrix, similar to the similarity matrix used in Hope. GraphSAGE [29] is a method that uses convolutional neural networks (CNNs) to learn the embedding vectors for attributed graphs. The last approach to learning graph embedding is based on random walks. DeepWalk [54] and Node2Vec [26] were the two most well-known proposals in this category. First, these methods represent the graph as a node sequence generated by random walks. Then they apply a natural language processing method called SkipGram [44] to maximize the co-occurrence probability of the nodes that are located in the w-sized window in the random walk. The difference between DeepWalk and Node2Vec is in the generated random walks used to learn the embedding vectors. While DeepWalk uses a pure random walk, Node2Vec takes advantage of a biased random walk, providing a trade-off between structural equivalence and homophily.

Recently, some research has been conducted to incorporate node attributes in the representation learning process to improve the quality of obtained embeddings, which are then used for the prediction of new links. For instance, Yang et al. [70] proposed a matrix factorization-based method called TADW, which took advantage of node attributes to enhance the quality of node embeddings. Pen et al. [52] proposed a framework to combine the structural information, nodal attributes, and node labels to learn embedding vectors. Xu et al. [69] proposed a method called GANE to combine different types of biological information to learn protein representations, which then can be used to predict PPI and disease genes. Masrour et al. [41] studied the impact of three different approaches for incorporating node attribute information in the process of representation learning and how they can improve the accuracy of link prediction.

Matrix factorization-based methods take an original matrix X and try to learn two lower rank matrices W and H, such that the product of W and H is as close as possible to X. These methods preserve local and global properties of networks in the form of a matrix, e.g., adjacency matrix, similarity matrix, etc., and factorize it to perform the link prediction task [20]. Most authors proposed methods based on non-negative matrix factorization to apply for the link prediction problem. For example, Chen et al. [16] proposed an NMF-based framework for the link prediction problem by combining manifold regularization and sparse learning. Ma et al. [39] proposed an asymmetric NMF (SNMF) method be applied to the link prediction task in temporal networks, which consisted of three steps. First, it uses SNMF to discover features in each time step. Next, it combines the feature matrices from the previous step to obtain a unified feature matrix. Finally, that feature matrix is used to predict new links in the current time step. In another work, presented by Chen et al. [14], a graph regularized NMF-based framework was proposed to solve the link prediction problem by using link weights and capturing local topology information. Ma et al. [40] applied graph regularized NMF to temporal networks to predict new links. In the time step T, GrNMF factorized the matrix associated with G_T while considering the networks in previous time steps as regularization. Therefore, it captures the topological information of temporal networks.

NMF-based methods have also been applied to attributed networks. Chen et al. [13] proposed an NMF-based framework to solve the link prediction problem in attributed networks. The proposed method simultaneously decomposed the adjacency matrix A and the attribute matrix B to map them to a lower dimension space while keeping the representations at a minimum distance and similar to each other. Gao et al. [25] proposed an NMF-based model in which three types of information, i.e., local information of nodes, the global structure of the network, and nodal attributes, were combined into a unified optimization function.

3 Methodology

In this section, the authors include some notions and preliminary information concerning the definition of the variants NMF and the similarity matrix using a weighted-biased random walk and the proposed method.

3.1 Notions and notations

Given an attributed network $G = (V, E, A)$ with n nodes $V = \{1, 2, \ldots, n\}$, m links $E = \{e_1, e_2, \ldots., e_m\}$ and t node attributes $A_i = \{ a_1, a_2, \ldots., a_t\}$, the network is usually represented by an adjacent matrix $X = (X_{ij})$ n × n and an attribute matrix $A = (A_{ik})$ n × m, where $X_{ij} = 1$ if a link exists between nodes i and j, or 0 otherwise; $A_{ik} = 1$ if node i has the k_{th} attribute, or 0 otherwise. The authors also consider the attributed graphs as undirected, connected, simple graphs, and all the node attributes conform to a unique multi-dimensional schema, A. Other notations, which will be often used in this paper, are summarized in Table 1.

3.2 A brief review of various versions of NMF

In this section, Non-negative Matrix Factorization (NMF), Graph Regularized Non-Negative Matrix Factorization (GNMF), and weighted non-negative matrix factorization (WNMF) are briefly reviewed.

Table 1 Some often used parameters

Symbol	Definition
$n \in R$	The number of network nodes.
$X \in R_{+}^{n*n}$	The adjacent matrix of network nodes.
$A \in R^{n*t}$	The attributed matrix of network nodes.
$W \in R_{+}^{n*k}$	The base matrix of network nodes.
$H \in R_{+}^{k*n}$	The coefficient matrix of network nodes.
$L \in R^{n*n}$	The Laplacian matrix of network nodes.
$P \in R^{n*n}$	The Transition matrix network nodes.
$D \in R^{n*n}$	The diagonal matrix.
$S \in R^{n*n}$	The similarity matrix of network nodes.

Non-negative matrix factorization (NMF) is a computational method for reducing the linear dimensionality of a given data matrix X, and it can be used to solve complex data mining and machine learning challenges. An NMF decomposes an initial data matrix into two low-dimensional non-negative matrices. One of the decomposed matrices is a coefficient matrix, which is used to store a low-dimensional representation, while the other is a basic matrix, which may be considered as parts-based representations of the original data. In the following, a summary of the Non-negative Matrix Factorization problem is stated:

Given a non-negative matrix $X_{n\,*\,n}$, find two non-negative matrices $W_{n\,*\,k}$ and $H_{n\,*\,k}$ with the condition k << rank(X), that minimize $F(X, WH^T)$, where $F(A, B)$ is a loss function defining the "distance" between the matrices A and B. The choice of the loss function F affects the solution of the minimization problem. One popular choice is the Frobenius norm (or the Euclidean Distance).

$$min_{W,H\geq 0} = \left\|X - WH^T\right\|_F^2, \tag{1}$$

where $\|.\|_F$ indicates the Frobenius norm, constrain $W, H \geq 0$ requires that all the elements in matrices W and H are non-negative.

NMF optimization is a convex optimization problem [29]. According to the NP-hardness of the problem and the lack of suitable convex formulations, non-convex formulations with remarkably straightforward solvability are usually employed, and only local minima can be achieved in a reasonable time for computation. The multiplicative iterative updating of the Frobenius norm can be achieved as follows:

$$W_{ik} \leftarrow W_{ik} \frac{(XH)_{ik}}{\left(WH^T H\right)_{ik}}, \tag{2}$$

$$H_{jk} \leftarrow H_{ik} \frac{\left(X^T H\right)_{jk}}{\left(H^T H W^T\right)_{jk}}, \tag{3}$$

where at the very outset of the iterative update process, the two non-negative matrices W_0 and H_0 are initialized randomly. The iterative update process is performed until the given terminal condition is fulfilled, as presented in (2) and (3). Finally, the final W and H can be obtained.

However, NMF has two major shortcomings, one of which is that it only recognizes global data structures of X and ignores the local relationships between data. A graph regularization is applied to NMF to solve the first problem. Cai et al. [70] introduced the graph regularization non-negative matrix factorization (GNMF), which is a well-known effective method that incorporates a graph Laplacian term to consider the intrinsic geometrical structure. The objective function of GNMF, in particular, can be seen as follows:

$$min_{W,H\geq 0} = \left\|X - WH^T\right\|_F^2 + \alpha \, Tr\left(HLH^T\right), \tag{4}$$

where $\alpha \geq 0$ is the regularization parameter, $L = D - S$ is called the Laplacian matrix, which is based on spectral graph and manifold learning theories. D is a diagonal matrix with $D_i = \sum_j^n S_{ij}$, and S is a similarity matrix between nodes. The multiplicative update rules to solve (4) are given as the following.

$$W_{ik} \leftarrow W_{ik} \frac{(XH)_{ik}}{\left(WH^T H\right)_{ik}}, \tag{5}$$

$$H_{jk} \leftarrow H_{ik} \frac{\left(X^T W + \lambda SH\right)_{jk}}{\left(HW^T W + \lambda DH\right)_{jk}}. \tag{6}$$

Additionally, Cai et al. [9] proved that the objective function under these two update rules is convergent. In this way, the manifold learning theory can be combined with the NMF, which has an excellent performance.

Another problem of NMF is that it is extremely sensitive to outliers and noises because of applying the squared error function to measure the loss; this issue leads to the objective function being easily dominated by a few outliers with large errors. Kong et al. [5] proposed $\ell_{2,1}$-norm to enhance the robustness of NMF. It alleviates the impact of noises or outliers by utilizing $\iota_{2,\,1}$-norm to replace the Frobenius norm. The objective function of RNMF is defined as:

$$min_{W,H\geq 0} = \left\|X - WH^T\right\|_{2,1}^2, \tag{7}$$

where $\|.\|_{2,\,1}$ represents the $\ell_{2,1}$ norm.

Other modified versions of the negative matrix factorization algorithm include weighted non-negative matrix factorization (WNMF) [31], which is applied to emphasize the significance of important components; the element is more important in the case of higher weights. The objective function for general weighted non-negative matrix factorization can be formulated as follows:

$$min_{W,H\geq 0} = \left\|Yo\left(X - WH^T\right)\right\|_F^2, \tag{8}$$

where $\|.\|_F^2$ is the Frobenius norm, $Y \in R^{n\,*\,n}$ implies the weight matrix, and o denotes the Hadamard product. Since the objective function in the equation is not convex with W and H jointly, the purpose is to find a local minimum by iteratively updating W and H in a similar way with the unweight NMF in the eq. (1).

3.3 Constructing the similarity matrix based on the random walk

In the attributed network, two data sources can be applied to perform the link prediction task. The first source of data comes from the network and the set of connections between nodes; the second is the data regarding the nodes and their attributes. With the increase of rich graph attributes, including gene annotations in protein interaction networks and user profiles in social networks, it is more important than ever to consider both the structure and attribute information of graphs for high-quality link prediction.

According to the homophily property of social networks, the relationships between nodes with similar attributes tend to be stronger than the relationships between nodes with different attributes, and they will be more probable to connect in the network [4, 23]; therefore, attribute information can influence the existence of links in networks. In order to further improve the efficiency and accuracy of node similarity measurement in the attributed network, the Structural and Attribute Random Walk Similarity (SARWS) is presented to compute the similarity between nodes through fusing structure and attribute information.

The first step is embedding the information about vertex attribute similarity into a transformed weighted graph $G_0 = (V, E, W)$. In particular, for each edge e = $(u_i, u_j) \in E$, an edge weight $w(e)$ is assigned to quantify the vertex attribute similarity for u_i and u_j. As a result, the vertex attribute information of G is encoded into the weighted graph G_0 as edge weights. The well-known cosine similarity of the angle between two node vectors is used to measure the similarity between two nodes. The reason for choosing cosine similarity is its effectiveness for sparse vectors that consider only non-zero values. For two nodes u_i and u_j, whose attribute vector is $A_i =\{ a_{i1}, a_{i2}, \ldots., a_{it}\}$ and $A_j =\{ a_{j1}, a_{j2}, \ldots., a_{jt}\}$, respectively, the attribute similarity is expressed as follows:

$$ATSIM\left(u_i, u_j\right) = \frac{\sum_{d=1}^{t} A_{id} A_{jd}}{\sqrt{\sum_{d=1}^{t} \left(A_{id}\right)^2} \cdot \sqrt{\sum_{d=1}^{t} \left(A_{jd}\right)^2}}, \tag{9}$$

where t denotes the dimension of an attribute vector.

The second step is to run the biased random walk on the weighted graph to discover similarities between the nodes. Each node initially has a walker in random walk methods, so each walker would randomly select a neighbor of the node it currently stands on to localize. By a special rule of transition probability, the random walk similarity is constructed for a pair of nodes which could better capture both the information potential of topological and attribute relationships between nodes. A weight-biased random walk on a graph can be defined by a more general transition matrix, where the element p_{ij} gives again the probability that a walker on the node u_i of the graph will move to node u_j in a single step but depending on appropriate weights for each pair of vertex u_i and u_j. In the present article, the appropriate weight for each pair of nodes in the network is considered to be determined proportional to the attribute similarity ($ATSIM$) between the nodes obtained from the Eq. (9).

Therefore, a transition probability $p_{ij} = ATSIM_{ij}$ on each link (u_i, u_j) is assigned. The Local Random Walk (LRW) algorithm [37] uses semi-local information to obtain similarities between nodes. According to the Bias Local Random Walk model, the final formula is defined as follows:

$$S_{ij}^{BLRW}(\eta) = \sum_{l=1}^{\eta} \frac{d_i}{2|E|} \cdot \frac{ATSIM_{ij}}{\sum_{j\in\Gamma(i)} ATSIM_{ij}}(l) + \frac{d_j}{2|E|} \cdot \frac{ATSIM_{ij}}{\sum_{i\in\Gamma(j)} ATSIM_{ij}}(l), \tag{10}$$

in which η is the number of random walk steps. *d* and *E* are referred to as the degree of node and the number of existing links in the network, respectively. In the SARWS similarity matrix, each entity obtains the similarity between nodes using structural information and attributes by applying a formula. The obtained matrix captures nodal attribute similarity in addition to the higher-order structural proximity of node pairs. Therefore, it can be treated as the weight matrix of the graph. For the sake of simplicity, from now on, we refer to the SARWS matrix as S.

3.4 The unified model: RGNMF-AN

In this section, a new algorithm is proposed, which overcomes the deficiency of GNMF for the attributed network in the link prediction problem. Firstly, the formulation of the objective function is introduced; then, the optimization method is assigned. Under the NMF framework, the novel RGNMF-AN model is proposed, which considers topology structures and node attribute information simultaneously to optimize a unified objective function. By integrating topological and attribute information of network through matrix S in Section 3.3 and enabling $\ell_{2,1}$-norm to constrain the loss function and a regularization term, a unified model RGNMF-AN is proposed for link prediction in attributed networks. The optimized overall objective function is expressed as follows:

$$\mathrm{J}_{\mathrm{RGNMF-AN}} = \min_{W,H\geq 0}\left\|\mathrm{S} \text{ o } \left(X - WH^{T}\right)\right\|_{2,1}^{2} + \alpha \mathrm{Tr}\left(H \mathrm{L}_{\mathrm{S}} H^{\mathrm{T}}\right), \tag{11}$$

where *o* is the Hadamard product, $\|.\|_{2,\ 1}$ represents the $\ell_{2,1}$-norm, and α denotes the regularization parameter. $\mathrm{L_S}$ = D−S is called the Laplacian matrix, which is based on spectral graph theory and manifold learning theory. D is a diagonal matrix with $D_i = \sum_j^n \mathrm{S}_{ij}$, and S is the structure-attribute similarity between nodes in the objective function. In original GNMF algorithms, the similarity matrix is represented by the network's adjacency matrix; therefore, the attributed information of nodes cannot be expressed, and the network information is limited. However, the S similarity has been applied, which preserves a higher level of similarity between nodes in terms of structure topology and attribute information.

The link prediction similarity scores are determined in the first term. To tackle random noises in the observed network, it is important to apply $\ell_{2,1}$-norm to a loss function to achieve similarity score accuracy. The second term, $Tr(H^T L_S H)$, is a manifold regularization, which is a combination of local similarity, structure topology, and node attributed to obtain local topology information in a new latent space. In the original GNMF, the method utilizes the adjacency matrix to compute the Laplacian matrix. Despite all of its benefits, the adjacency matrix contains only 0/1 values for any pair of nodes, which results in not being able to distinguish between node pairs. While in real-world situations, each interaction has a specific value in the network. Therefore, we need to consider a new approach to obtain a more informative and accurate similarity matrix.

Since the objective function $\mathrm{J}_{\mathrm{RGNMF\ -\ AN}}$ is non-convex, it is extremely challenging to find the optimal global solution. The Lagrange function is used to update the objective function, which includes two variables W and H. It is convex for each matrix by fixing the other. According to matrix trace properties: $Tr(WH) = Tr(HW)$, $\|A\|_{2,1}$=*trace* (AUA^T), the objective function can be rewritten as follows:

$$\begin{aligned} \mathrm{J}_{\mathrm{RGNMF-AN}} &= \min_{\mathrm{U}\geq 0,\mathrm{W}\geq 0}(\mathrm{W},\mathrm{H}) \ \left\|\mathrm{S \ o} \ \left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right)\right\|_{2,1} + \alpha \ \mathrm{trace}\left(\mathrm{H}^{\mathrm{T}}\mathrm{LH}\right) \\ &= \mathrm{trace}\left(\left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right)\mathrm{U}\left(\mathrm{S \ o} \ \left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right)\right)^{\mathrm{T}}\right) \end{aligned} \tag{12}$$

in which the diagonal matrix U is defined as $U = [u_{ii}]_{n \times n}$ with the diagonal elements given by

$$\mathrm{u}_{\mathrm{ii}=}\left\|\left(\mathrm{S}^{1/2} \ \mathrm{o} \ \mathrm{X}\right)_{\mathrm{i}}-\left(\mathrm{S}^{1/2} \ \mathrm{o} \ \left(\mathrm{WH}^{\mathrm{T}}\right)\right)_{\mathrm{i}}\right\|_{2}, \tag{13}$$

where $(S^{1/2} \ o \ X)_i$ and $(S^{1/2} \ o \ (WH^T))_i$ is the ith column of $S^{1/2} \ o \ X$ and $S^{1/2} \ o \ (WH^T)$, respectively. In addition to this, please note that $S^{1/2} \ o \ S^{1/2}$. Hence, we have

$$\begin{aligned} \left\|\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right\|_{2,1} &= \mathrm{trace}\left(\left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right) \ \mathrm{U} \ \left(\mathrm{S \ o} \ \left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right)\right)^{\mathrm{T}}\right) \\ &= \mathrm{trace}\left(\left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right) \ \mathrm{U} \ \left(\mathrm{S \ o \ X}^{\mathrm{T}}-\mathrm{S \ o} \ \left(\mathrm{WH}^{\mathrm{T}}\right)\right)\right) \\ &= \mathrm{trace}\left(\left(\mathrm{X}-\mathrm{WH}^{\mathrm{T}}\right)\left(\ \mathrm{U} \ \left(\left(\mathrm{S \ o \ X}^{\mathrm{T}}\right)-\mathrm{U} \ \left(\mathrm{S \ o} \ \left(\mathrm{WH}^{\mathrm{T}}\right)\right)\right)\right)\right) \\ &= \mathrm{trace}\left(\left(\mathrm{XU}\left(\mathrm{S \ o \ X}^{\mathrm{T}} \ \right)\right)-\mathrm{XU}\left(\ \left(\mathrm{S \ o} \ \left(\mathrm{WH}^{\mathrm{T}}\right)\right)-\mathrm{WH}^{\mathrm{T}}\mathrm{U} \ \ \left(\left(\mathrm{S \ o \ X}^{\mathrm{T}}\right) + \mathrm{WH}^{\mathrm{T}}\mathrm{U} \ \left(\mathrm{S \ o} \ \left(\mathrm{WH}^{\mathrm{T}}\right)\right)\right)\right)\right) \end{aligned} \tag{14}$$

1. **First term:** $\|S \ o \ (X - \ WH^T)\|_{2,1}$

1.1. **Second part:** *trace* $((S \ o \ X^T)UHW^T)$

We have

$$\frac{\partial \mathrm{trace}\left((\mathrm{S \ o \ X})\mathrm{UHW}^{\mathrm{T}}\right)}{\partial \mathrm{W}} = (\mathrm{S \ o \ X})\mathrm{UH}, \tag{15}$$

$$\frac{\partial \mathrm{trace}\left((\mathrm{S \ o \ X})\mathrm{UHW}^{\mathrm{T}}\right)}{\partial \mathrm{H}} = \mathrm{U}\left(\mathrm{S \ o \ X}^{\mathrm{T}}\right)\mathrm{H}, \tag{16}$$

1.2. **Third part:** *trace* $(WH^TU(So(HW^T)))$

We have

$$\frac{\partial \mathrm{trace}\left(\mathrm{WH}^{\mathrm{T}}\mathrm{U}\left(\mathrm{S \ o} \ \left(\mathrm{HW}^{\mathrm{T}}\right)\right)\right)}{\partial \mathrm{W}} = 2\left(\mathrm{S \ o} \ \left(\mathrm{HW}^{\mathrm{T}}\right)\right)\mathrm{UH}, \tag{17}$$

$$\frac{\partial \mathrm{trace}\left(\mathrm{WH^TU}\left(\mathrm{S\ o\ }\left(\mathrm{HW^T}\right)\right)\right)}{\partial \mathrm{H}} = 2\mathrm{U}\left(\mathrm{S\ o\ }\left(\mathrm{HW^T}\right)\right)\mathrm{W}, \tag{18}$$

2. **Second term:** *trace*(H^TLH)

$$\frac{\partial trace\left(H^TLH\right)}{\partial H} = 2LH \tag{19}$$

3. Derivative of *J* with respect *W* and *H*

Considering the constraints *W*, *H*>≥0, we set Largrange multipliers A, B for them. Then, the Largrange function of the problem can be written as

$$J_{RGNMF-AN} = \left\| S\ o\ \left(X - WH^T\right)\right\|_{2,1} + \alpha\ trace\left(H^TLH\right) + trace\left(AW^T\right) + trace\left(BH^T\right) \tag{20}$$

According to (2), (3), (4), and (5), we have

$$\frac{\partial \mathrm{L}}{\partial \mathrm{W}} = -2(\mathrm{S\ o\ X})\mathrm{UH} + 2\left(\mathrm{S\ o\ }\left(\mathrm{WH^T}\right)\right)\mathrm{UH} + \mathrm{A}. \tag{21}$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{W}} = -2\mathrm{U}(\mathrm{S\ o\ X})\mathrm{W} + 2\mathrm{U}\left(\mathrm{S\ o\ }\left(\mathrm{WH^T}\right)\right)\mathrm{W} + 2\alpha\mathrm{LH} + \mathrm{B}. \tag{22}$$

According to the Kuhn-Tucker conditions, we have $\mathrm{A_{ij}W_{ij}} = 0$ and $\mathrm{B_{ij}H_{ij}} = 0$, and hence we get

$$-2((\mathrm{S\ o\ X})\mathrm{UH})_{\mathrm{ij}}\mathrm{W_{ij}} + 2\left(\left(\mathrm{S\ o\ }\left(\mathrm{WH^T}\right)\right)\mathrm{UH}\right)_{\mathrm{ij}}\mathrm{W_{ij}} + \mathrm{A_{ij}W_{ij}} = 0. \tag{23}$$

$$-2\ \left(\left(\mathrm{S\ o\ X^T}\right)\mathrm{W}\right)_{\mathrm{ij}}\mathrm{H_{ij}} + 2\left(\mathrm{U}\left(\mathrm{S\ o\ }\left(\mathrm{WH^T}\right)\right)\mathrm{W}\right)_{\mathrm{ij}}\mathrm{H_{ij}} + 2\alpha\ ((\mathrm{S-D})\mathrm{H})_{\mathrm{ij}}\mathrm{H_{ij}} + \mathrm{B_{ij}H_{ij}} = 0. \tag{24}$$

Therefore, we have

$$\mathrm{W_{ij}} \leftarrow \sqrt{\mathrm{W_{ij}}\frac{((\mathrm{S\ o\ X})\mathrm{UH})_{\mathrm{ij}}}{\left(\mathrm{S\ o\ }\left(\mathrm{WH^T}\right)\right)\mathrm{UH}\Big)_{\mathrm{ij}}}}. \tag{25}$$

$$\mathrm{H_{ij}} \leftarrow \sqrt{\mathrm{H_{ij}}\frac{\left(\mathrm{U}\left(\mathrm{S\ o\ X^T}\right)\mathrm{W} + \alpha\mathrm{DH}\right)_{\mathrm{ij}}}{\left(\mathrm{U}\left(\mathrm{S\ o\ HW^T}\right)\mathrm{W} + \alpha\mathrm{SH}\right)\Big)_{\mathrm{ij}}}}. \tag{26}$$

We obtain the new basic matrix W and the feature matrix H by minimizing Eqs. (25) and (26). Finally, we can reconstruct the similarity score of the original network with $\widehat{X} = WH^T$ to obtain the link prediction similarity. The RGNMF-AN algorithm is as follows:

Algorithm 1:RGNMF-AN

```
Input:
X : adjacency matrix of an undirected attributed network;
A : Attribute matrix of the network
S: the indicator similarity matrix;
K: a dimension of latent space;
max_iter: maximum number of iteration;
Parameter α;
Output:
X̂: similarity score matrix;
Initialization: $A \leftarrow A_0$, $L_A = Laplace(A)$ .
Begin algorithm
Divide $X$ into training set $G^T$ and probe set $G^P$
Randomly initialize $W$, $H$
$S = BLRW(G^T, A)$
Exploit node attribute and semi-local information using Eq (9)
Preserve local information using Eq (10)
FOR iter = 1: max_iter
    Update $W$ using Eq (24)
    Update $H$ using Eq (25)
    Get $W$ and $H$ after convergence
End of For
Return similarity score matrix $\hat{X} = WH^T$
```

***Algorithm* BLRW**

```
Input:
$G^T$: training graph
A : Attribute matrix of the network
Output:
S: node attribute and semi-local indicator similarity matrix
FOR each link (i,j) in $G^T$:
    Compute the Cosine similarity of the associated attributes of i and j
    Normalize the similarities and Obtain the transition probability matrix for $G^T$
FOR each non-connected pair of nodes (i,j) in $G^T$:
    Compute the similarity between i and j using Eq (10)
Return similarity matrix S
```

3.5 Proof of convergence

The RGNMF-AN model uses iterative updating rules to optimize the model. To be more specific, at each iteration, we update a variable e.g. W, while keeping fixing the other variable, e.g. H. The proof of convergence for Eqs. (25) and (26) is given in [64]. Therefore, the convergence of the RGNMF-AN under those updating rules to the local minimum can be easily proven.

3.6 Computational complexity analysis

The computational cost of the RGNMF-AN algorithm is analyzed, which consists of two key steps. The first step is to construct a similarity matrix, and the second step is to update the rules for optimizing the objective function. For the first key step, constructing a similarity (SARWS), one needs to $O(n^2d)$ to calculate a similar node clustering matrix, where d denotes the average node degree. For the second key step, in each iteration, updating W and H according to Eqs. (17) and (19) require $O(n^2 * k * iter)$. Since both $iter$ and k are constants, the overall time complexity of RGNMF-AN is $O(n^2)$.

4 Experiments

4.1 Datasets

To evaluate the proposed method, we conducted some experiments on 9 real-world datasets, including citation and biological networks. All the experiments were carried out on a desktop

Table 2 Details of datasets. |V|, |E|, and |Attr| are the number of nodes, links, and attributes of each node, respectively.CC is the average clustering coefficient, Max_Degree is the maximum degree of a node in the network, $\langle d \rangle$ is the average shortest path in the network and r is the assortative coefficient

Dataset	|V|	|E|	|Attr|	CC	Max_Degree	$\langle d \rangle$	r
Texas	187	328	1703	0.1937	104	3.0362	−0.2687
Cornell	195	304	1703	0.1568	94	3.2005	−0.2408
Washington	230	366	1703	0.1974	122	2.9946	−0.2226
Wisconsin	265	530	1703	0.2080	122	3.2599	−0.1882
Cora	2708	5278	1433	0.2407	168	6.3109	−0.0656
CiteSeer	3312	4536	3703	0.1425	99	9.3104	0.0480
E.coli	1505	5576	343	0.1033	152	3.5643	−0.1237
C.elegan	1607	2877	343	0.0285	151	4.4924	−0.1692
Drosophila	5511	19,712	343	0.0128	166	4.1488	−0.0495

PC equipped with a quad-core Intel i7 2.20GHz processor and 16GB RAM. Details of these networks can be found in Table 2.

- **Cora**[1] [59] is a citation network in which nodes present machine learning papers, and an edge between two papers is formed only if one of them is cited by the other. Keywords in papers are treated as node attributes. It consists of 2708 nodes and 5278 edges.
- **CiteSeer** is a citation network in which nodes present scientific papers, and an edge between two papers is formed only if one of them is cited by the other. Keywords in papers are treated as node attributes. It consists of 3312 nodes and 4732 edges.
- **Cornell**, **Texas**, **Washington,** and **Wisconsin**[2] are subnetworks drawn from the WebKB dataset. Each of them contains web pages as nodes and links connecting them. Cornell has 195 nodes and 304 edges. Texas consists of 187 nodes and 328 edges. Wisconsin contains 265 nodes and 530 edges.
- **Protein-Protein-Interaction Networks**

We also evaluated our proposed method, using three protein networks provided by Guo's dataset [28]. *E. coli* dataset containing 1834 proteins with 6954 interactions, Drosophila dataset containing 7059 proteins with 21,975 interactions, and C. elegan dataset containing 1607 proteins with 2877 interactions.

4.2 Comparing methods

To evaluate our proposed method, we compare it against some of the state-of-the-art methods for representation learning. The description of these methods can be found here:

- NMF is used directly and the reconstructed matrix is considered as the score matrix.

$$\min_{W\geq 0, H\geq 0} \left\| \left(X - WH^T\right) \right\|_F^2$$

[1] https://linqs.soe.ucsc.edu/data
[2] http://www.cs.cmu.edu/~webkb/

- GNMF [9] method combines the link structure with graph neighbor information of nodes.

$$\min_{U\geq 0, V\geq 0} \left\|X - WH^T\right\|_F^2 + \lambda Tr\left(H^T LH\right),$$

where λ is the parameter, L is the Laplacian matrix that is obtained via $L = D - S$, in which D is the diagonal matrix and S is the similarity matrix between node pairs.

- FSC-NMF [3] is an NMF-based method that learns node embeddings while taking the structure and contents of nodes into account. The model includes two optimization functions that iteratively optimize them.

$$\min_{B_1\geq 0, B_2\geq 0} \|A - B_1 B_2\|_F^2 + \alpha_1\|B_1 - U\|_F^2 + \alpha_2\|B_1\|_F^2 + \alpha_3\|B_2\|_F^2,$$

$$\min_{U\geq 0, V\geq 0} \left\|C - UV^T\right\|_F^2 + \beta_1\|U - B_1\|_F^2 + \beta_2\|U\|_F^2 + \beta_3\|V\|_F^2,$$

- TADW [70] incorporates the text features of each node into the embedding process under a framework of matrix factorization

$$\min_{W,H} \left\|X - W^T HT\right\|_F^2 + \frac{\lambda}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$$

- M-NMF [66] uses a matrix factorization approach to learn the representation of nodes while capturing the community structure of the network

$$\min_{M\geq 0, U\geq 0, H\geq 0, C\geq 0} \left\|S - MU^T\right\|_F^2 + \alpha\left\|H - UC^T\right\|_F^2 - \beta Tr\left(H^T BH\right),$$

where α and β are non-negative parameters.

- GraphSAGE [29] is a representation learning method based on CNN, which uses the nodes' attributes of the neighborhood of each node to learn the embedding vector of that node.

4.3 Evaluation metrics

AUC [38]: The AUC metric evaluates link prediction methods based on the scores, they give to non-observed links. The AUC calculation is done through the following procedure: First, we compare the scores of randomly selected missing links vs. randomly selected non-existent links. Then, among n comparisons, assume the score of the missing link has been higher than the score of the non-existent link for n' times and also for n", the scores were equal. Then AUC can be defined as:

$$AUC = \frac{n' + 0.5\, n''}{n}$$

F-measure: F1-measure can be interpreted as the weighted average of two other evaluation metrics, i.e. precision and recall. The F1-measure can be calculated as the following:

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)}$$

In which, the precision and recall can be calculated as follows:

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive},$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

ROC curve [55]: this curve can compare the capability of different models in identifying positive and negative samples, in varying decision thresholds. If model A has a higher true positive rate with respect to a false-positive rate, in all the thresholds, compared to model B, then model A is outperforming model B in the classification task. This curve is appropriate when we are dealing with a highly imbalanced dataset, like complex networks, in which the percentage of the negative class is significantly higher than the positive class.

RMSE and **PCC** are the standard deviations of the differences between the vectors of predicted and actual values of node pairs.

$$RMSE = \sqrt{\frac{\sum_{i,j}\left(r_{ij}-A_{ij}\right)^2}{n}},$$

$$PCC = \frac{\sum_{i,j}\left(A_{ij}-average(A)\right)\left(r_{ij}-average(r)\right)}{\sqrt{\sum_{i,j}\left(A_{ij}-average(A)\right)^2}\sqrt{\sum_{i,j}\left(r_{ij}-average(r)\right)^2}},$$

in which, A_{ij} and r_{ij} are the actual value and the predicted value, respectively.

4.3.1 Parameter settings

To evaluate the proposed method against other methods, as usual, we need to split our data into training and testing sets. To get the positive samples for the test set, we randomly delete 10% of edges from the original network while making sure that the residual network is obtained after the edge removals are connected. And add them to the test set. The remaining edges belong to the training set. To obtain negative samples for training and testing, we randomly select an equal number of node pairs that have no link connecting them. Other parameters were set empirically. In particular, $\alpha = 0.1$, max_iter = 40, and the dimension of latent space to 70.

4.4 Experiment analysis

The obtained results for the AUC measure are summarized in Table 3. The best-obtained result for each dataset is shown, highlighted in bold. It is shown that in all datasets, the proposed method has outperformed other methods. RGNMF-AN and RGNMF are abbreviations for the proposed methods with and without considering node attributes in the random walk generation process. In particular, except for CiteSeer and C.elegans, the proposed method has achieved the highest AUC in all the networks. To be more specific, the obtained results show that adding structure and content information can significantly improve the prediction accuracy compared to regular GNMF. For example, in Wisconsin, Cora, and CiteSeer networks, RGNMF-AN achieved 11%, 34%, and 29% improvement and RGNMF achieved 10%, 33%, and 27% of AUC, compared to GNMF. Also, it is shown that by using both structural

Table 3 AUC results for link prediction

Algorithms	Dataset								
	Texas	Cornell	Washington	Wisconsin	Cora	CiteSeer	E.coli	C.elegans	Drosophila
RGNMF-AN	**0.8222**	**0.7737**	**0.8051**	**0.8537**	**0.9232**	0.9428	**0.9583**	0.8091	**0.7897**
RGNMF	0.7896	0.7403	0.7901	0.8482	0.9187	0.9230	0.9582	0.8060	0.7855
NMF	0.5476	0.5826	0.6715	0.7176	0.8627	0.8861	0.8959	0.7293	0.7672
GNMF	0.7852	0.7258	0.7555	0.7417	0.5806	0.6504	0.9303	0.7635	0.7630
FSCNMF	0.7736	0.7691	0.7716	0.7851	0.8356	0.9292	0.9318	0.7701	0.7161
TADW	0.7461	0.6543	0.7060	0.5728	0.5748	0.6217	0.7226	0.6919	0.5467
MNMF	0.7576	0.7015	0.5408	0.7239	0.9020	**0.9643**	0.9131	**0.8525**	0.7531
GraphSAGE	0.6364	0.6423	0.7438	0.7367	0.7584	0.8190	0.6269	0.5799	0.5538

and content information, we can obtain better results compared to using only structural information.

The micro-F1 scores that resulted from all the algorithms are reported in Table 4. The best-obtained result for each dataset is shown, highlighted in bold. Results show that in all the datasets, the proposed methods have a better or competitive advantage over NMF and GNMF. In addition to that, in all networks, the proposed methods significantly outperformed embedding-based methods. For instance, the RGNMF-AN and RGNMF have achieved 39%, 30%, and 29% higher F-measure in Cornell, Washington, and Wisconsin compared to the best results obtained by embedding methods.

Tables 5 and 6 report the RMSE and PCC results for the proposed methods vs. comparing methods. As Table 5 shows, RGNMF-AN and RGNMF have achieved the lowest RMSE, compared to NMF and GNMF in all the networks except for CiteSeer. Compared to the embedding-based methods, i.e., FSCNMF, TADW, and MNMF, the proposed methods, achieved remarkable improvement in all networks. From Table 6, it can be concluded that overall, the proposed methods do not perform well in terms of PCC, compared to embedding-based methods, while compared with NMF and GNMF, they achieved considerable improvement.

To investigate the effect of content and structure on the performance of link prediction in NMF-based methods, we used Fig. 1, to compare obtained AUCs for different networks. It is obvious that taking advantage of higher-order similarity, obtained by LRW, can increase the prediction accuracy compared to regular GNMF. Furthermore, in addition to structural information, taking advantage of the content of nodes in computing the similarity matrix can improve the obtained results.

Table 4 F-measure results for link prediction

Algorithms	Dataset								
	Texas	Cornell	Washington	Wisconsin	Cora	CiteSeer	E.coli	C.elegans	Drosophila
RGNMF-AN	**0.9983**	**0.9985**	**0.9986**	**0.9986**	**0.9999**	**0.9999**	**0.9994**	**0.9997**	**0.9998**
RGNMF	**0.9983**	**0.9985**	**0.9986**	**0.9986**	0.9998	0.9997	**0.9994**	**0.9997**	**0.9998**
NMF	0.9954	0.9983	0.9935	0.9952	0.9996	0.9996	**0.9994**	0.9993	0.9994
GNMF	0.9972	0.9936	0.9945	0.9953	0.9994	0.9996	0.9993	0.9995	0.9994
FSCNMF	0.9819	0.6071	0.6944	0.7000	0.6508	0.7549	0.8536	0.6898	0.6270
TADW	0.6428	0.6250	0.6944	0.6111	0.5569	0.5894	0.6579	0.6271	0.5256
MNMF	0.6250	0.5714	0.4583	0.6000	0.7817	0.9514	0.8563	0.7491	0.7681
GraphSAGE	0.5780	0.6856	0.6111	0.6256	0.6325	0.7329	0.5435	0.5121	0.5072

Table 5 RMSE results for link prediction

Algorithms	Dataset								
	Texas	Cornell	Washington	Wisconsin	Cora	CiteSeer	E.coli	C.elegans	Drosophila
RGNMF-AN	**0.0018**	**0.0016**	**0.0016**	**0.0014**	**0.0001**	0.0001	**0.0006**	**0.0002**	**0.0001**
RGNMF	**0.0018**	**0.0016**	**0.0016**	**0.0014**	**0.0001**	0.0001	**0.0006**	**0.0002**	**0.0001**
NMF	0.0020	0.0018	0.0019	0.0016	0.0002	0.0001	0.0007	0.0003	0.0002
GNMF	0.0022	0.0022	0.0017	0.0016	0.0002	**0.00009**	0.0008	0.0003	0.0002
FSCNMF	0.1758	0.2085	0.1980	0.2004	0.2088	0.1805	0.1123	0.2153	0.2359
TADW	0.2138	0.2316	0.2073	0.2575	0.2445	0.2396	0.2238	0.2259	0.2492
MNMF	0.2190	0.2574	0.2736	0.2284	0.1602	0.0465	0.1174	0.1486	0.1617
GraphSAGE	0.2012	0.2232	0.2455	0.2154	0.2342	0.1923	0.2852	0.3337	0.3010

4.4.1 Parameter sensitivity

The performance of RGNMF-AN depends on various factors and parameters. Those parameters are the alpha, the dimensionality of latent space, and the maximum iteration number. To investigate the impact of these parameters, we performed experiments such that two out of three parameters are fixed and the third one is varied over a range of values. The alpha parameter can take a value between $\{10^5,\ 10^3,\ 10^1,\ 10^{-1},\ \ 10^{-3},\ 10^{-5}\}$ while the latent space dimension can have a value between $\{10,\ 20,\ \ldots,\ 90\}$ and the maximum number of iterations can be in the range of $\{10,\ 20,\ \ldots,\ 60\}$.

4.4.2 Impact of alpha

Figure 2 demonstrates the effect of parameter alpha on the four evaluation metrics. It is shown that by decreasing the value of alpha from 10^5 to 10^1, the performance does not improve, but when the value of alpha is equal to 10^{-1}, the performance increases significantly and from that point, varying the alpha value does not produce any significant change. Therefore, we can see that the optimal value of alpha should be set to 10^{-1}.

4.4.3 Impact of iteration

Figure 3 illustrates the effect of the maximum number of iterations on the performance of RGNMF-AN using the four evaluation metrics. From observing Fig. 3, we can see that with

Table 6 PCC results for link prediction

Algorithms	Texas	Cornell	Washington	Wisconsin	Cora	CiteSeer	E.coli	C.elegans	Drosophila
RGNMF-AN	0.0887	0.1323	0.0636	0.1588	0.0858	0.1202	0.1977	0.0229	0.0489
RGNMF	0.0845	0.1337	0.0752	0.1644	0.0877	0.1187	0.2016	0.0286	0.0401
NMF	0.0002	0.0187	0.0279	0.0836	0.0710	0.0961	0.1461	0.0292	0.0449
GNMF	0.1530	0.1109	0.1083	0.1140	0.0101	0.0205	0.2011	0.0469	0.0339
FSCNMF	0.1798	**0.4350**	**0.4829**	**0.4850**	0.4231	0.5551	**0.7488**	0.4204	0.2681
TADW	**0.4043**	0.3241	0.4150	0.1129	0.1503	0.2204	0.3335	0.3252	0.0582
MNMF	0.3780	0.1919	−0.0134	0.3244	**0.6455**	**0.9070**	0.7283	**0.6427**	**0.5943**
GraphSAGE	0.3554	0.2565	0.1123	0.1067	0.1121	0.1764	0.4615	0.1345	0.1132

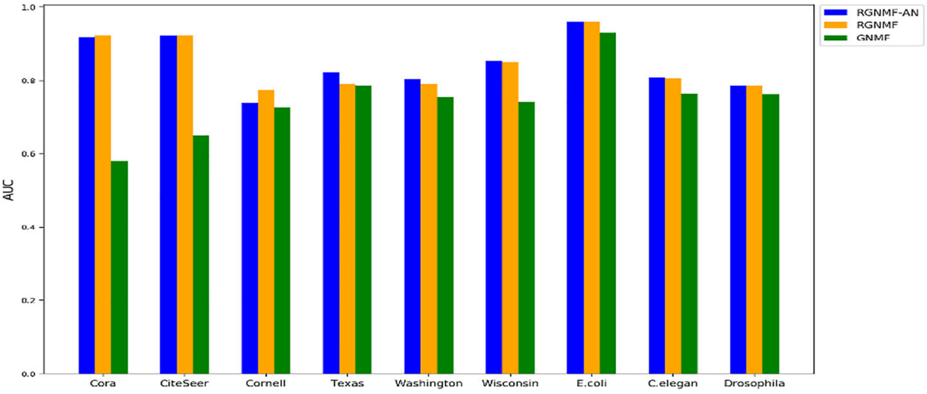

Fig. 1 Comparing the effect of structural and attributes information on GNMF

the increase of iteration number, the model shows unstable results until we reach the point where the maximum number of iterations is equal to 40. At that point, the performance is stabilized which proves that the model is converged.

4.4.4 Impact of latent space dimension

Figure 4 shows the effect of the value of the latent space dimension on the performance of the model. It is very critical to find the optimal value of a dimension since it has a direct impact on both performance and complexity. From Fig. 4, we can see that for most datasets, the best performance is achieved when we set the values of dimensions 70 and 80. Therefore, we chose to consider 70 as the optimal value for the latent space dimension.

To make the evaluation more reliable, we use ROC curves to compare the proposed methods, i.e., RGNMF-AN and RGNMF, against other algorithms. The ROC curve for each algorithm illustrates the ability to distinguish between the positive and negative classes for that particular algorithm. In the worst-case scenario, an algorithm predicts the labels, randomly and as a result, the curve would be in the form of $x = y$. Since ROC curves are suitable for experiments in highly imbalanced datasets, they are perfect for the evaluation of link prediction methods. Figure 5 compares the obtained ROC curves for each method in all networks. Overall, RGNMF-AN and RGNMF obtained better results in almost all the thresholds and therefore achieved the highest area under the curve.

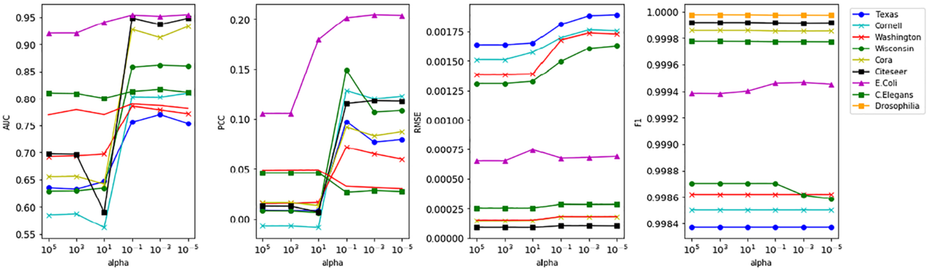

Fig. 2 Impact of alpha on the performance of the proposed method

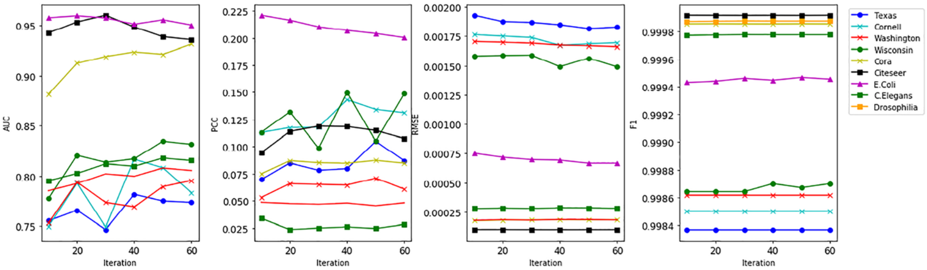

Fig. 3 Impact of iteration number on the performance of the proposed method

5 Discussion

Most of the existing methods in the area of link prediction concentrate on the information provided by the topological structure of the networks. Although these methods have some benefits, they neglect a crucial information source, i.e. nodal attributes, which negatively impacts their performance. In this work, we proposed a method that considers both structural and non-structural information, that is present in the network. The nodes' attributes are integrated with topological information via multiplying a Hadamard matrix. This approach was used to solve the link prediction problem in weighted and directed networks, but to the best of our knowledge, this is the first time that it has been applied to attributed graphs to deal with link prediction. Our proposed similarity matrix, which is the combination of structural and non-structural information has high accuracy in capturing the proximity between node pairs.

For evaluation purposes, we have performed various experiments on different datasets. The comparison methods are some of the most well-known NMF-based and deep learning-based methods. In particular, we have compared the performance of the proposed method, i.e., RGNMF-AN to six other methods, i.e., NMF, GNMF, FSC-NMF, TADW, M-NMF, and GraphSAGE. The obtained results in Table 3, illustrate the superiority of our algorithm in terms of AUC. In particular, in all the networks, except for CiteSeer, the proposed method has achieved better results. Table 4 shows the obtained F-scores for all methods. Clearly, RGNMF-AN has outperformed all the other methods and achieved better results. Tables 5 and 6 summarize the performance of RGNMF-AN, against the comparing methods. It is obvious that overall, the proposed methods have achieved the least RMSE among other methods. More specifically, except for CiteSeer, in all the other networks, RGNMF-AN

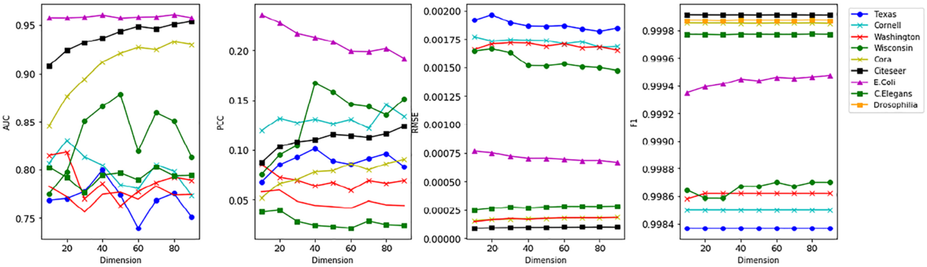

Fig. 4 Impact of latent space dimension on the performance of the proposed method

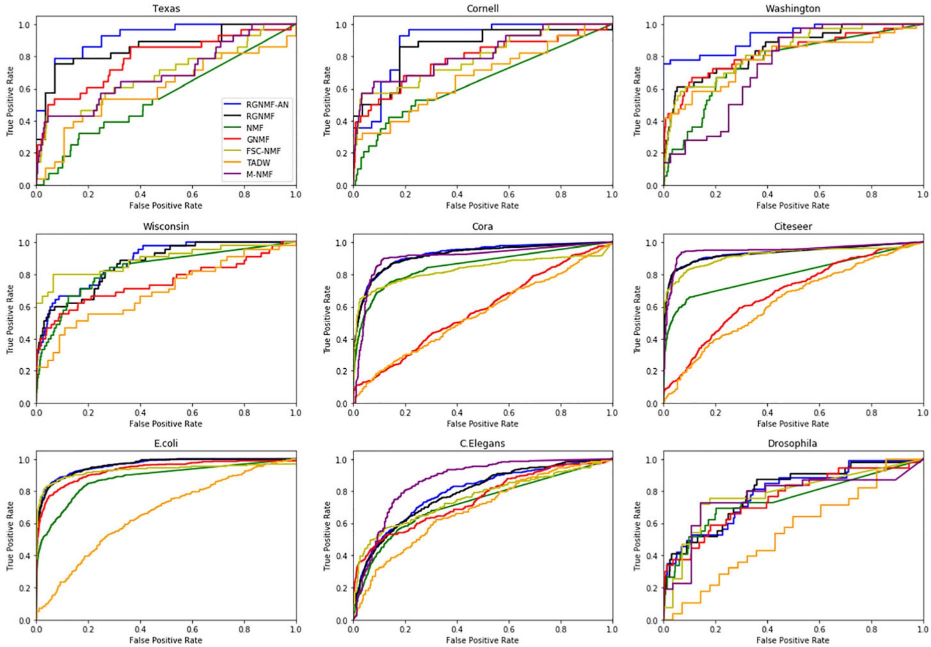

Fig. 5 ROC Curve for the proposed method vs. comparing methods

achieved the best RMSE, whereas, according to Table 6, it was not able to do well, compared to other methods.

To further evaluate other aspects of our algorithm, we performed some experiments to investigate the effect of various parameters, i.e., alpha, number of iteration, and latent space dimensions on the performance of RGNMF-AN. The obtained results are illustrated in Figs. 2, 3, and 4. We have also, plotted the ROC curve for comparing the proposed method, vs. comparing methods that illustrate the better performance of RGNMF-AN.

6 Conclusion

In this paper, the authors focused on the problem of link prediction in attributed networks. The main contribution of the present paper was to propose a method, called Robust Graph Regularization Nonnegative Matrix Factorization (RGNMF-AN), that simultaneously considers topological and non-topological information about networks, to capture the semi-local proximity between a pair of nodes and present it as a weight between node pairs. Furthermore, $\ell_{2,1}$-norm was also used to constrain the objective function, to minimize the influence of the random noises and spurious links, in the step of reconstructing the original matrix. In addition, multiplicative updating rules are applied to learn and optimize the model parameters. According to the experimental analysis performed on various datasets, the proposed method has outperformed the other NMF-based methods and achieved better results, in terms of predicting new links. In particular, an extensive experimental analysis was performed on nine real-world datasets, using AUC, F-measure, RMSE, and PCC metrics to compare the obtained results of the RGNMF-AN against other methods, and the superiority of the RGNMF-AN was proved.

In future studies, the proposed method would have the option to be applied to multilayer, signed, and bipartite networks to deal with the link prediction problem. Furthermore, suggesting an approach to specify a high-order relationship among nodes by applying hypergraph form in the present study could be an excellent topic for future studies.

Declarations

Ethics approval No animals or human participants are involved in this research work.

Conflict of interest I confirm that this work is original and has either not been published elsewhere or is currently under consideration for publication elsewhere. None of the authors have any competing interests in the manuscript.

References

1. Aggarwal CC, Li N (2011) On node classification in dynamic content-based networks. In: Proceedings of the 2011 SIAM international conference on data mining. SIAM
2. Aiello LM et al (2012) Friendship prediction and homophily in social media. ACM Trans Web 6(2):9
3. Bandyopadhyay S et al (2018) Fscnmf: fusing structure and content via non-negative matrix factorization for embedding information networks. arXiv preprint arXiv:1804.05313
4. Berahmand K et al (2020) A new attributed graph clustering by using label propagation in complex networks. J King Saud Univ-Comput Inf Sci
5. Berahmand K, Nasiri E, Li Y (2021) Spectral clustering on protein-protein interaction networks via constructing affinity matrix using attributed graph embedding. Comput Biol Med 138:104933
6. Berahmand K et al (2021) A modified DeepWalk method for link prediction in attributed social network. Computing:1–23
7. Berahmand K et al (2021) A preference random walk algorithm for link prediction through mutual influence nodes in complex networks. J King Saud Univ – Comput Inf Sci
8. Bhagat S, Cormode G, Muthukrishnan S (2011) Node classification in social networks. In: Social network data analytics. Springer, pp 115–148
9. Cai D, He X, Han J, Huang TS (2010) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
10. Cai H, Zheng VW, Chang KC-C (2018) A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans Knowl Data Eng 30(9):1616–1637
11. Cao S, Lu W, Xu Q (2016) Deep neural networks for learning graph representations. In: AAAI
12. Chen H, Li X, Huang Z (2005) Link prediction approach to collaborative filtering. In: Proceedings of the 5th ACM/IEEE-CS joint conference on digital libraries (JCDL'05). IEEE
13. Chen B, Li F, Chen S, Hu R, Chen L (2017) Link prediction based on non-negative matrix factorization. PLoS One 12(8):e0182968
14. Chen G, Xu C, Wang J, Feng J, Feng J (2019) Graph regularization weighted nonnegative matrix factorization for link prediction in weighted complex network. Neurocomputing 369:50–60
15. Chen G, Xu C, Wang J, Feng J, Feng J (2020) Nonnegative matrix factorization for link prediction in directed complex networks using PageRank and asymmetric link clustering information. Expert Syst Appl 148:113290
16. Chen G et al (2020) Robust non-negative matrix factorization for link prediction in complex networks using manifold regularization and sparse learning. Phys A: Stat Mech Appl 539:122882
17. Chunaev P (2020) Community detection in node-attributed social networks: a survey. Comput Sci Rev 37: 100286
18. Currarini S, Matheson J, Vega-Redondo F (2016) A simple model of homophily in social networks. Eur Econ Rev 90:18–39
19. Dev P (2016) Homophily and community structure in networks. J Public Econ Theory 18(2):268–290
20. Divakaran A, Mohan A (2019) Temporal link prediction: a survey. N Gener Comput:1–46
21. Esmaeili M, Saad HM, Nosratinia A (2021) Semidefinite programming for community detection with side information. IEEE Trans Netw Sci Eng

22. Forouzandeh S, Berahmand K, Rostami M (2020) Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. Multimed Tools Appl 80:1–28
23. Forouzandeh S, Rostami M, Berahmand K (2021) Presentation a trust Walker for rating prediction in recommender system with biased random walk: effects of H-index centrality, similarity in items and friends. Eng Appl Artif Intell 104:104325
24. Franceschini A, Lin J, von Mering C, Jensen LJ (2016) SVD-phy: improved prediction of protein functional associations through singular value decomposition of phylogenetic profiles. Bioinformatics 32(7):1085–1087
25. Gao S, Denoyer L, Gallinari P (2011) Temporal link prediction by integrating content and structure information. In: Proceedings of the 20th ACM international conference on Information and knowledge management
26. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks. Kdd 2016:855–864
27. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. Proc Natl Acad Sci 106(52):22073–22078
28. Guo Y, Li M, Pu X, Li G, Guang X, Xiong W, Li J (2010) PRED_PPI: a server for predicting protein-protein interactions based on sequence data with probability assignment. BMC Res Notes 3(1):1–7
29. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems
30. Keikha MM, Rahgozar M, Asadpour M (2019) DeepLink: a novel link prediction framework based on deep learning. J Inf Sci:0165551519891345
31. Kim Y-D, Choi S (2009) Weighted nonnegative matrix factorization. In: 2009 IEEE international conference on acoustics, speech and signal processing. IEEE
32. Kumar A, Singh SS, Singh K, Biswas B (2020) Link prediction techniques, applications, and performance: a survey. Phys A: Stat Mech Appl 553:124289
33. Kumar A, Mishra S, Singh SS, Singh K, Biswas B (2020) Link prediction in complex networks based on significance of higher-order path index (SHOPI). Phys A: Stat Mech Appl 545:123790
34. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788–791
35. Li Y, Ngom A (2013) The non-negative matrix factorization toolbox for biological data mining. Source Code Biol Med 8(1):1–15
36. Li J et al (2017) Attributed network embedding for learning in a dynamic environment. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management
37. Liu W, Lü L (2010) Link prediction based on local random walk. EPL (Europhysics Letters) 89(5):58007
38. Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Phys A: Stat Mech Appl 390(6):1150–1170
39. Ma X, Sun P, Qin G (2017) Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability. Pattern Recogn 71:361–374
40. Ma X, Sun P, Wang Y (2018) Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks. Phys A: Stat Mech Appl 496:121–136
41. Masrour F et al (2018) Attributed network representation learning approaches for link prediction. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE
42. Mehrpooya A et al (2021) High dimensionality reduction by matrix factorization for systems pharmacology. Brief Bioinform
43. Menon AK, Elkan C (2011) Link prediction via matrix factorization. In: Joint european conference on machine learning and knowledge discovery in databases. Springer
44. Mikolov T et al (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems
45. Mokhtia M, Eftekhari M, Saberi-Movahed F (2020) Feature selection based on regularization of sparsity based regression models by hesitant fuzzy correlation. Appl Soft Comput 91:106255
46. Muniz CP, Goldschmidt R, Choren R (2018) Combining contextual, temporal and topological information for unsupervised link prediction in social networks. Knowl-Based Syst 156:129–137
47. Nasiri E, Bouyer A, Nourani E (2019) A node representation learning approach for link prediction in social networks using game theory and K-core decomposition. Eur Phys J B 92(10):228
48. Nasiri E, Berahmand K, Rostami M, Dabiri M (2021) A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. Comput Biol Med 137:104772
49. Nasiri E, Berahmand K, Li Y (2021) A new link prediction in multiplex networks using topologically biased random walks. Chaos Soliton Fract 151:111230
50. Newman ME (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64(2):025102

51. Ou M et al (2016) Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining
52. Pan S et al (2016) Tri-party deep network representation. Network 11(9):12
53. Pavlov M, Ichise R (2007) Finding experts by link prediction in co-authorship networks. FEWS 290:42–55
54. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM
55. Provost F, Fawcett T (2001) Robust classification for imprecise environments. Mach Learn 42(3):203–231
56. Qian B et al (2016) Double constrained NMF for partial multi-view clustering. In: 2016 international conference on digital image computing: techniques and applications (DICTA). IEEE
57. Saberi-Movahed F, Eftekhari M, Mohtashami M (2019) Supervised feature selection by constituting a basis for the original space of features and matrix factorization. Int J Mach Learn Cybern:1–17
58. Saberi-Movahed F et al (2021) Decoding clinical biomarker space of covid-19: exploring matrix factorization-based feature selection methods. medRxiv
59. Sen P, Namata G, Bilgic M, Getoor L, Galligher B, Eliassi-Rad T (2008) Collective classification in network data. AI Mag 29(3):93–93
60. Tropp JA (2003) Literature survey: nonnegative matrix factorization. University of Texas at Asutin, p 26
61. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). IEEE Trans Pattern Anal Mach Intell 27(12):1945–1959
62. Wang C, Satuluri V, Parthasarathy S (2007) Local probabilistic models for link prediction. In: Seventh IEEE international conference on data mining (ICDM 2007). IEEE
63. Wang P, Xu BW, Wu YR, Zhou XY (2015) Link prediction in social networks: the state-of-the-art. SCIENCE CHINA Inf Sci 58(1):1–38
64. Wang D, Liu JX, Gao YL, Zheng CH, Xu Y (2015) Characteristic gene selection based on robust graph regularized non-negative matrix factorization. IEEE/ACM Trans Comput Biol Bioinform 13(6):1059–1067
65. Wang D, Cui P, Zhu W (2016) Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining
66. Wang X et al (2017) Community preserving network embedding. In: Thirty-First AAAI Conference on Artificial Intelligence
67. Wang W et al (2020) Attributed collaboration network embedding for academic relationship mining. ACM Trans Web 15(1):1–20
68. Xie J, Douglas PK, Wu YN, Brody AL, Anderson AE (2017) 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. J Neurosci Methods 282:81–94
69. Xu B, Li K, Zheng W, Liu X, Zhang Y, Zhao Z, He Z (2018) Protein complexes identification based on go attributed network embedding. BMC Bioinforma 19(1):1–10
70. Yang C et al (2015) Network representation learning with rich text information. In: IJCAI
71. Yuan G et al (2014) Exploiting sentiment homophily for link prediction. In: Proceedings of the 8th ACM conference on recommender systems. ACM

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.