Knowledge-Based Systems 196 (2020) 105800

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

NetSRE: Link predictability measuring and regulating

Xingping Xian [a], Tao Wu [b,*], Shaojie Qiao [c,*], Xi-Zhao Wang [d], Wei Wang [e], Yanbing Liu [f]

[a] Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
[b] School of Cybersecurity and Information Law, Chongqing University of Posts and Telecommunications, Chongqing, China
[c] School of Software Engineering, Chengdu University of Information Technology, Chengdu, China
[d] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[e] Institution of Cybersecurity, Sichuan University, Chengdu, China
[f] Chongqing Engineering Laboratory of Internet and Information Security, Chongqing University of Posts and Telecommunications, Chongqing, China

ARTICLE INFO

Article history:
Received 18 August 2019
Received in revised form 6 February 2020
Accepted 20 March 2020
Available online 27 March 2020

Keywords:
Network data
Link prediction
Link predictability
Structural patterns
Low-rank coding
Structure perturbation

ABSTRACT

Link prediction is an elemental issue for network-structured data mining, which has already found a wide range of applications. The organization of real-world networks usually embodies both regularities and irregularities, and the precision of link prediction algorithms coincides with the portion of a network being categorized as regular. Quantifying and controlling how well an unobserved link can be predicted is a fundamental problem in link prediction. This paper proposes a structural regularity-exploring architecture, called NetSRE, for measuring and regulating link predictability of networks. The proposed NetSRE assumes that there are consistent interaction patterns across the local subgraphs of networks and one of them can be represented by a linear summation of the others, and thus, link predictability can be characterized by the self-representation degree of network structures. Specifically, NetSRE includes (1) a low Frobenius norm pursuit-based self-representation network model for predicting the "true" underlying networks, (2) a "structural regularity" index for measuring the link predictability of networks, i.e., the inherent difficulty of link prediction independent of specific algorithms, and (3) an importance measuring method for structural role exploration of network links and a link-based structure perturbation algorithm for link predictability regulation. Experimental results on real-world networks validate the performance of our method. It is found that real-world networks have various structural regularities and link predictability can be estimated based on structure mining directly. We show that network heterogeneity provides a way to intrinsically segregate network links into qualitatively distinct groups, which have different influences on the link predictability of networks.

1. Introduction

Networks have been proved to be an effective abstraction for representing real-world complex systems [1]. With network models, the various complex systems, ranging from the Internet and the World Wide Web to biological and social networks, are all considered as a collection of discrete units that interact through a set of connections. Driven by the increasing availability of network data [2,3], network science has seen a surge of interest in the last twenty years, and the research focus has been transferred from statistical analysis-based empirical studies [4–6] to practical structure mining. Most recently, many structure mining works have been developed, including community detection [7–9], influential node ranking [10–12], graph classification [13,14], and graph summarization [15].

In network science, link prediction [16–18] is a fundamental notion, which attempts to uncover missing links or detect spurious links using features intrinsic to the network topology itself. In the last decade, the link prediction problem has received increased attention and a growing number of methods have been proposed for link prediction. These methods can be roughly divided into three classes [19,20]: similarity-based methods, maximum likelihood methods, and matrix decomposition-based methods. Link prediction can benefit a wide range of real-world applications. For instance, in biological networks, our knowledge of biological interactions is highly limited; using the predicted results for guiding the design of experiments, rather than blindly checking all possible interactions, can sharply reduce the experimental costs [21]. In online social networks, the potential commercial interests have led to the creation and proliferation of fake accounts, and link prediction can help to find the fake accounts by detecting abnormal social relations [22]. In e-commerce websites, link prediction can be used for recommending products to target users [23]. In the security domain,

* Corresponding authors.
E-mail addresses: wutaoadeny@gmail.com (T. Wu), sjqiao@cuit.edu.cn (S. Qiao).

https://doi.org/10.1016/j.knosys.2020.105800
0950-7051/

with the availability of network data related to terrorist activities, link prediction can be used to reveal some hidden relationships to discover the potential terrorists [24].

Just as a popular saying goes, "every coin has two sides". Recently, link prediction has raised privacy concerns in the case where the predicted link is between users who would like to keep the relationship private. Specifically, in the real world, many types of information, such as sexual contacts, purchase records, and financial relationships, are considered highly sensitive and anonymized for privacy preserving. However, based on link prediction, many privacy inference attack methods have been proposed. For example, Zheleva and Getoor [25] conducted a preliminary study on sensitive relationship inference from anonymized graphs. Ying and Wu [26] investigated how well a graph randomization approach can protect sensitive links and showed that similarity measures can be exploited by attackers to significantly improve their confidence and accuracy of the predicted sensitive links. Yang et al. [27] identified a fundamental weakness of link-based graph anonymization mechanisms and exploited it to recover most of the original graph structure. Michael et al. [28] presented a "link reconstruction attack method, which can infer connections that a user wants to hide to preserve his privacy. Moreover, link prediction-based de-anonymization methods are defined to match the accounts across networks for user identification [29–31].

Motivation. As discussed above, link prediction can be applied to predict the potential relationship between two individuals. To reveal the structure of various networks accurately, more robust and sophisticated link prediction methods are required. From another perspective, link prediction may increase the risk of information leakage. Even if the data publishers remove sensitive information before network datasets are released, it may still be inferred by link prediction, thereby encroaching user privacy. Naturally, considering the interests of all parties, the problem of **link predictability measuring and regulating (LPMR)**, which characterizes the inherent difficulty of link prediction and explores the potential influence of network links on the accuracy of link prediction methods.

Based on considerable literature on link prediction, researchers have started realizing the significance of structural features of networks. Besides relying on specific algorithms, the accuracy of link prediction methods depends on the network structure itself. Especially, no algorithm can achieve satisfactory performance in random networks, while high level of prediction accuracy can be achieved readily in regular networks. In fact, real-world networks usually embody both regular components and irregular components, where only the former can be modeled and explained. Consequently, the accuracy of link prediction depends on the regularity level of networks, i.e., the proportion of the regular components. Therefore, the intrinsic regularity of networks is the fundamental factor influencing the accuracy of link prediction.

Link predictability denotes the inherent difficulty of link prediction in networks independent of specific algorithms, which can be calculated by estimating their regularity level. By measuring the predictability of a network, we can determine whether the deficient performance of link prediction is caused by an inappropriate algorithm or is due to the irregularity of the network itself, and then estimate how a large space remains for performance improvement. Furthermore, by regulating the link predictability of networks, the risks arising from link prediction, such as privacy disclosure, can be reduced directly. However, despite its practical importance, so far, the problem of LPMR has not been fully investigated.

Contributions. This paper proposes a network structural regularity exploring architecture, called NetSRE. NetSRE measures the link predictability of networks by exploring their organization principles, which indicate the upper bound of link prediction accuracy and provide guidance for algorithm optimization. NetSRE assumes that links play different roles in network organization, where some of them have disproportionate influence on network regularity, and then, link predictability can be regulated based on a limited number of links. By analyzing the organizational relationships in network self-representation, the potential links can be predicted based on the learned structure patterns of networks. Along this line, the distribution of the representative subgraphs in network self-representation indicates the link predictability of networks, and the links with various substitutabilities in network self-representation have different influences on link predictability regulating. The main contributions of this paper are summarized as follows:

- First, we model a network structure from the perspective of self-representation and formalize the question as an optimization problem. Using the self-representation model, the network structure can be decomposed into a set of representative subgraphs and the combination relationships between them. By applying the model on link prediction, i.e., **L**ow **F**robenius norm-based **L**ink **P**rediction (**LFLP**), the expressive power of the self-representation model is proved.
- Second, based on the assumption that real-world networks have a certain degree of regularity and the data matrices are approximately low rank, we define a low-rank pursuit-based self-representation model to uncover the common representative subgraphs. According to the learned representation matrix, we define a **S**tructural **R**egularity **I**ndex (**SRI**) to measure the link predictability of networks.
- Third, according to the usage of links in the network self-representation model, we define a novel importance metric for network links to indicate their regularity level. Based on the link selection mechanism, the structure perturbation-based **L**ink **P**redictability **R**egulation (**LPR**) algorithm is proposed to control the networks' potentiality for link prediction.

The remainder of this paper is organized as follows. Section 2 surveys the background and related work. Section 3 introduces the problem definition and evaluation mechanism. Section 4 introduces our proposed method. Section 5 shows the experiments conducted, and Section 6 concludes the paper.

2. Background and related work

2.1. Link prediction and network modeling

The most generic framework used for link prediction is similarity-based methods [32], including local indices Common Neighbors (CN), Adamic–Adar (AA) [33], Resource Allocation (RA) [34], etc., global indices Katz [35], SimRank [36], etc., and quasi-local indices Local Path Index (LP) [34], Local Random Walk (LRW) [37], etc. Recently, some novel similarity-based methods, including dynamical response-based method [38], neighborhood-based method [39], etc., have also been developed. The methods always assume that two nodes are more likely to be linked if they have a higher heuristic node similarity. However, an important limitation of these methods is that they lack universal applicability to different types of networks and do not achieve good consistency across all networks [40]. To solve the link prediction problem sophisticatedly, many network modeling based methods have been proposed. Specifically, maximum likelihood methods, including stochastic block model (SBM) [41], hierarchical structure model (HSM) [42], LOOP model [43], etc., presuppose the organizing principles of networks and learn the model parameters by maximizing the likelihood of the observed structure and

then calculate the likelihood of any non-observed link according to the rules and parameters. However, the methods are highly time consuming and can always only handle networks with a few thousands of nodes, while real social networks scale from millions to more than a billion nodes. Matrix-based approaches, including LO [19], NMF [44,45], RPCA [46], fusion models [47], kernel framework based on NMF [48] etc., model network structure with matrix theory, such as matrix factorization theory and low rank and sparse theory, and they predict missing links via solving the formulated optimization problem. Compared with the maximum likelihood methods, matrix-based approaches have better computational efficiency and are applicable to large-scale networks. However, the performance of them depends on a number of conditions, including information about node attributes [44], multiple networks ensemble [45], and dense network topology [46], which highly limits their applications. Motivated by the recent success of deep learning, some deep learning methods for link prediction have been proposed, including Graph Neural Networks (GNN) for link prediction [49], dual convolutional neural network for link prediction [50], deep dynamic network embedding for link prediction [51], etc., which extract structure feature from networks automatically and improve prediction performance significantly.

Different from these works, this paper aims to explore the organization principle of real-world networks at sufficient depth. We try to reveal the relationships between the substructures of networks and analyze their roles in network organization. Then we measure and regulate the link predictability of networks under linear summation assumption. To the best of our knowledge, this is one of the first several linear coding based network modeling methods.

2.2. Structural regularity exploration and link perturbation

To explore the structural regularity of networks, based on graphlets [52] and motifs [53], Rossi et al. [54,55] proposed graphlet counting methods according to the finding that graphlet frequencies often carry significant information about the local network structure. Benson et al. [56] considered network motifs as fundamental building blocks for complex networks and found different higher-order organizational patterns at the level of network motifs. Koutra et al. [57] summarized real-world networks to provide the most succinct description in terms of local graph structures, including stars, bipartite cores, cliques, and chains. Shen et al. [58] proposed a general stochastic block model for detecting multiple structure types, including community structure and multipartite structure. Rather than finding specific structural features or patterns of networks, Zhou et al. [59] focused on the structural consistency of networks and developed an eigenvalue perturbation method to estimate the consistency level of networks.

To promote information diffusion, Cheng et al. [60] proposed a users activity frequency-based similarity measure to optimize the connections of online social networks. To improve the robustness of large-scale infrastructure networks, Ash et al. [61] developed an evolutionary algorithm to generate evolved networks that are resilient to cascading failure. To prevent the propagation of viruses, Wang et al. [62] proposed an immunization strategy to influence the effective structure of networks. Liu et al. [63] defined a measure of link diffusion importance to identify redundant links, thereby enhancing the performance of node ranking. However, although a few structure perturbation-based methods have been proposed, the link predictability regulation problem has attracted little attention and is still challenging.

2.3. Low-rank learning method

For processing big data in complex networks, a fundamental task is to find a low-dimensional representation of the high-dimensional data. To handle the problem, principal component analysis (PCA) [64] was proposed to be one of the most common approaches in recovering the best low-rank representation. However, the PCA method works well for data with Gaussian noise, and its performance degrades for data with gross errors. Then, a more robust method, robust principal component analysis (RPCA) [65], was proposed, which can be formulated as follows.

$$\min_{\mathbf{A},\mathbf{E}} rank(\mathbf{A}) + \|\mathbf{E}\|_1,\ s.t., \mathbf{X} = \mathbf{A} + \mathbf{E} \tag{1}$$

The PCA and RPCA methods assume that the data are distributed in a single space. Real-world data, however, often originate from a set of multiple subspaces. To correctly partition the data into different subspaces, the sparse subspace clustering (SSC) [66] and low rank representation (LRR) [67] approaches were proposed. Formally, the SSC algorithm solves the following problem:

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_1 + \lambda\|\mathbf{E}\|_1\ s.t.\ \mathbf{X} = \mathbf{XZ} + \mathbf{E} \text{ and } \mathrm{diag}(\mathbf{Z}) = \mathbf{0}. \tag{2}$$

LRR is similar to SSC, except that it aims to find a low rank representation instead of a sparse representation. The objective function of LRR can be formulated as follows:

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{\mathbf{2,1}}\ \mathbf{s.t.}\ \mathbf{X} = \mathbf{XZ} + \mathbf{E}. \tag{3}$$

Note that SSC determines the sparse representation of each data vector individually, and may not capture the global structure of **X**. In contrast, LRR finds the lowest rank representation of the entire data jointly.

3. Problem definition and evaluation mechanism

3.1. Link prediction

To clearly illustrate the LPMR problem, a working flowchart of network data analysis with structure perturbation is illustrated in Fig. 1. The working flowchart contains the following parts: first, the datasets about real-world complex systems are collected; then, on the basis of the datasets, networks are constructed to characterize the interactive relationships of the objects in complex systems; next, because networks always contain sensitive links or noisy links that can be identified by link prediction, link predictability measuring and regulating methods are conducted to protect sensitive information or improve data quality; finally, according to the goal of link predictability regulating, the anti-inference networks for privacy-preserving and the enhanced networks for data mining can be obtained, and downstream network analysis tasks utilize the resulted networks to get insights about the complex systems.

Given an undirected network $G = (V, E)$, where V is a set of $|V|$ nodes and $E \subseteq V \times V$ is a set of links, link prediction aims to generate a predicted network based on the observed network $G^T = (V, E^T)$ to approximate the "true" underlying network G. For performance evaluation, the observed network G^T is constructed through adding and deleting links. All links of G^T are denoted as training set E^T, and the difference between the observed network G^T and the underlying network G is defined as testing set E^P.

To evaluate the influence of link predictability regulation on the performance of link prediction algorithms, we adopt two standard metrics, area under the receiver operating characteristic curve (AUC) and precision, to estimate link prediction accuracy.

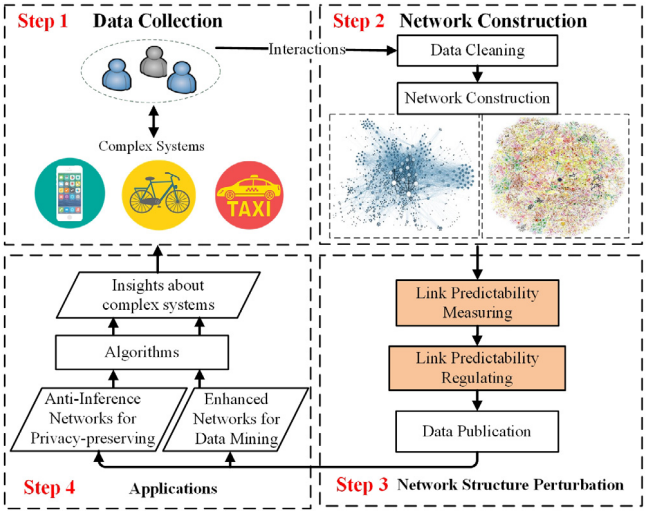


Fig. 1. Working flowchart of network data analysis with structure perturbation.

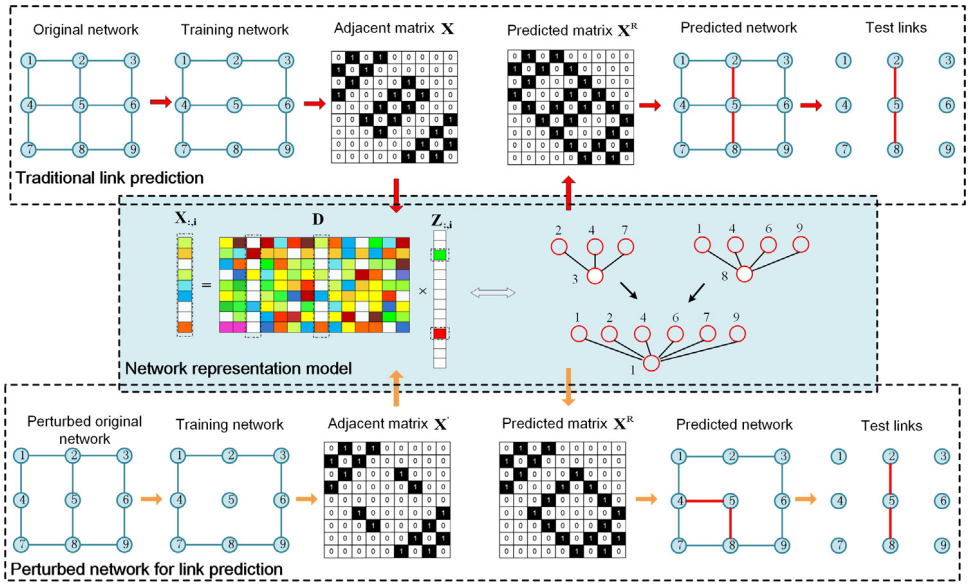


Fig. 2. Illustration of structure perturbation-based link predictability regulation.

- **AUC**. Among n times of independent comparisons, if there are n' times in which the score of the missing (spurious) link is higher (or less) than that of the non-existent (existent) link and n'' times in which the two have the same score, then AUC can be calculated as

$$\mathbf{AUC} = (\mathbf{n}' + \mathbf{0.5n}'')/\mathbf{n} \tag{4}$$

  If all scores are generated from an independent and identical distribution, AUC will be approximated to 0.5. Therefore, the extent to which AUC exceeds 0.5 indicates how much better the algorithm performs than the pure chance.
- **Precision**. Precision is defined as the ratio of the relevant links to the number of selected links. If L_p links among the top-L links are accurately predicted, then

$$\mathbf{Precision} = \mathbf{L_p/L} \tag{5}$$

3.2. Network data preparation

To evaluate the performance of link prediction methods, based on a "true" underlying network, various observed networks are generated by structure modification. Here, we consider the typical modification techniques that have been widely used in the area of network data analysis [43,68–70]. Specifically, the selected techniques are introduced as follows:

- **Adding**. This method generates the observed network $G^T = (V, E^T)$ by only adding $k|E|$ links randomly, where k is the modification coefficient.
- **Deleting**. This method generates the observed network $G^T = (V, E^T)$ by randomly eliminating $k|E|$ links.

3.3. Link predictability regulation

Definition 1 (*Link Predictability*)**.** Link predictability characterizes the inherent difficulty of link prediction independent of specific algorithms. It can be quantified by the portion of regular components, i.e., the subgraphs that obey the organization principle of networks. Networks have less predictability if they tend to be random and more predictability if their structures are of high regularity.

Table 1
Notations used in the paper.

Notations	Descriptions
G	The "true" underlying network
G^T	The observed network corresponding to G
E^T	The link set of G^T used as training set
E^P	The difference between G and G^T
$\mathbf{X}$	The adjacency matrix of network
$\mathbf{Z}$	The representation matrix corresponding to $\mathbf{X}$
$\mathbf{E}$	The sparse matrix denoting the noise in $\mathbf{X}$
λ	The trade-off parameter to balance different terms
$\mathbf{SM}$	The existence likelihoods matrix of network links
σ_r	Structural regularity index
$RD(k)$	Regularity of the subgraph centered node k
$U_{i,j}$	The importance of link (i, j)

Definition 2 (*Link Importance*)**.** Link importance measures the change in the portion of the regular components caused by the deletion or addition of a specified link. For an existing link, the more the subgraph containing the link is used for network representation, the higher is substitutability of the link in network organization and the lesser the impact on regular components.

Definition 3 (*Link Predictability Regulation*)**.** Given a network G, we learn the important link set E^R based on network modeling, and then, perturb as few network links as possible, guided by their importance to change the network's regularity level. By link perturbation, the resulting network G^R has different link predictability from that of network G.

Note that, as the major goal of publishing network data is to pursue useful and worthy research, one needs to limit the level of link perturbation to maintain data utility. An illustrative example of link perturbation-based predictability regulation is given in Fig. 2. According to the figure, the upmost plot is the paradigm of traditional link prediction, where the missing links can be predicted based on the network representation model. By comparing the predicted links with the test links (denoted as red solid lines), we can find that all missing links in the training network are identified correctly. As the regularity level of networks has a direct impact on the accuracy of link prediction, the predictability of networks can be regulated by important link-based structure perturbation. Consequently, in the downmost plot of Fig. 2, in the perturbed network, the accuracy of the link prediction task is reduced.

The notations used in this paper are described in Table 1.

4. Our method

4.1. Network representation modeling

Empirical studies on complex networks have indicated that most real-world networks possess some common topological characteristics, such as small-world, scale-free, and core–periphery features, which can be modeled effectively based on the presupposed organization principles [1,42]. Moreover, from the perspective of network summarization, Koutra et al. [15] found that network structures are composed of an enriched set of representative subgraphs, including cliques, stars, chains, and bipartite cores. Inspired by these studies, in this study, networks are viewed as a linear summation of a set of elemental subgraphs with specific interaction patterns. That is, networks can be represented using the elemental subgraphs as structural bases. Specifically, let $\mathbf{X} \in R^{n\times n}$ denote the adjacency matrix of network G. Each column of matrix $\mathbf{X}$ is viewed as a local structure $\mathbf{X}_{:,i}$; thus, $\mathbf{X}$ contains n local structures, i.e., $[\mathbf{X}_{:,1}, \mathbf{X}_{:,2}, \ldots, \mathbf{X}_{:,n}]$. Given a complete basis matrix $\mathbf{D} = [\mathbf{D}_{:,1}, \mathbf{D}_{:,2}, \ldots, \mathbf{D}_{:,n}] \in R^{n\times n}$, i.e., a collection of structural bases, each local structure $\mathbf{X}_{:,i}$ can be represented by a linear combination of bases, which is given as follows:

$$\mathbf{X}_{:,i} = [\mathbf{D}_{1,:}\mathbf{Z}_{:,i}, \mathbf{D}_{2,:}\mathbf{Z}_{:,i}, \ldots, \mathbf{D}_{n,:}\mathbf{Z}_{:,i}]^{\mathrm{T}} = \sum_{k=1}^{n} \mathbf{D}_{:,k} Z_{k,i} \tag{6}$$

where $Z_{k,i}$ corresponds to the weight of the base $\mathbf{D}_{:,k}$. That is, $\mathbf{X}_{:,i}$ is actually a linear combination of matrix $\mathbf{D}$'s columns weighted by the elements of $\mathbf{Z}_{:,i}$. Thus, the adjacency matrix $\mathbf{X} \in R^{n\times n}$ of network G can be represented by $\mathbf{X} = \mathbf{DZ}$, where $\mathbf{Z} \in R^{n\times n}$ is a representation matrix capturing the organization principle of the network. Thus, the network modeling can be simply transferred to an optimization problem:

$$\min_{\mathbf{D},\mathbf{Z}} \|\mathbf{X}-\mathbf{DZ}\| + \|\mathbf{Z}\| \tag{7}$$

where $\|\cdot\|$ denotes a certain matrix norm.

According to [71], the learned atoms of the basis matrix $\mathbf{D}$ almost never coincide with the original data, and hence, cannot be considered as good representatives of the data. To recognize the organization principle of networks and find representative subgraphs from the actual subgraphs, the best candidate for the basis matrix $\mathbf{D}$ is the adjacency matrix $\mathbf{X}$. Thus, the optimization problem can be reformulated as

$$\min_{\mathbf{Z}} \|\mathbf{X}-\mathbf{XZ}\| + \|\mathbf{Z}\| \tag{8}$$

where each local structure $\mathbf{X}_{:,i}$ can be represented as a combination of the others.

4.2. Self-representation model-based link prediction

Link prediction is a widely accepted means of assessing the validity of network models. Here we apply the self-representation model on link prediction to prove its expressive power firstly. Although the adjacency matrix $\mathbf{X}$ is adopted as the basis matrix for network representation, the network structures are still not guaranteed to be fully represented by the product of the basis matrix $\mathbf{X}$ and the representation matrix $\mathbf{Z}$. Therefore, in this paper, we define matrix $\mathbf{E}$ to denote the difference matrix between the adjacency matrix $\mathbf{X}$ and the graph representation $\mathbf{XZ}$, i.e., $\mathbf{X} = \mathbf{XZ} + \mathbf{E}$. Because individuals may have different interaction patterns in reality, the modeling of real-world networks should be node-oriented. Thus, because each column of the adjacency matrix $\mathbf{X}$ represents the interactions between a node and the remaining nodes, to characterize the node-specific corruptions in networks, $\ell_{2,1}$ norm, i.e., $\|\cdot\|_{2,1}$, is adopted in our model to constrain the matrix $\mathbf{E}$ in terms of a graph node. The goal of link prediction is to infer the "true" underlying network by finding the structural patterns of the observed network. By applying the proposed self-representation network model defined in Eq. (8) to the observed network G^T, the learned representation matrix $\mathbf{Z}^*$ reveals the organization principle of the network; thus, the unknown structure can be inferred based on it. To avoid overfitting, we adopt here the Frobenius norm to constrain the magnitude of the representation matrix $\mathbf{Z}$.

Based on the above discussion, the objective function for network link prediction can be formulated as

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_{\mathcal{F}}^2 + \lambda\|\mathbf{E}\|_{2,1}, \ s.t., \mathbf{X} = \mathbf{XZ} + \mathbf{E} \tag{9}$$

where $\|\mathbf{Z}\|_{\mathcal{F}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}{}^2$, and $\|\mathbf{E}\|_{21} = \sum_{j=1}^{n}\sqrt{\sum_{i=1}^{n} e_{ij}{}^2}$, λ is the trade-off parameter to balance different terms.

Lemma 1. *For the matrix* $\mathbf{X}$, $\mathbf{Z}$ *and* $\mathbf{E}$, *the augmented item* $\mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]$ *and error item* $\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2$ *can be merged, i.e., we have the following equation:*

$$\begin{aligned}&\arg\min_{\mathbf{E}} \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]+\frac{\mu}{2}\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2\\&=\arg\min_{\mathbf{E}} \frac{\mu}{2}\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}+\frac{\mathbf{Y}_1}{\mu}\|_{\mathcal{F}}^2\end{aligned} \tag{10}$$

Proof. According to the definition of the interior product and the Frobenius norm of the matrix, we have

$$\begin{aligned}&\arg\min_{\mathbf{E}} \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]+\frac{\mu}{2}\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2\\&=\arg\min_{\mathbf{E}} \frac{\mu}{2}(\langle\frac{2}{\mu}\mathbf{Y}_1,\mathbf{X}-\mathbf{XZ}-\mathbf{E}\rangle+\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2)\\&=\arg\min_{\mathbf{E}} \frac{\mu}{2}(\langle\frac{\mathbf{Y}_1}{\mu},\frac{\mathbf{Y}_1}{\mu}\rangle+2\langle\frac{\mathbf{Y}_1}{\mu},\mathbf{X}-\mathbf{XZ}-\mathbf{E}\rangle+\\&\quad\langle\mathbf{X}-\mathbf{XZ}-\mathbf{E},\mathbf{X}-\mathbf{XZ}-\mathbf{E}\rangle-\langle\frac{\mathbf{Y}_1}{\mu},\frac{\mathbf{Y}_1}{\mu}\rangle)\\&=\arg\min_{\mathbf{E}} \frac{\mu}{2}(\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}+\frac{\mathbf{Y}_1}{\mu}\|_{\mathcal{F}}^2-\|\frac{\mathbf{Y}_1}{\mu}\|_{\mathcal{F}}^2)\end{aligned} \tag{11}$$

By removing the terms irrelevant to $\mathbf{E}$, it is converted to

$$\arg\min_{\mathbf{E}} \frac{\mu}{2}\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}+\frac{\mathbf{Y}_1}{\mu}\|_{\mathcal{F}}^2 \tag{12}$$

Consequently, the equality holds.

Optimization Solution. To solve (9), we employ an efficient optimization technique, the augmented Lagrange multiplier (ALM) algorithm [72]. First, we introduce an auxiliary variable $\mathbf{J}$ to make the objective function separable:

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{J}\|_{\mathcal{F}}^2+\lambda\|\mathbf{E}\|_{2,1},\ s.t.,\mathbf{X}=\mathbf{XZ}+\mathbf{E},\mathbf{Z}=\mathbf{J} \tag{13}$$

This problem can be solved by the inexact ALM method, which minimizes the following augmented Lagrange function:

$$\begin{aligned}&L(\mathbf{J},\mathbf{Z},\mathbf{E})=\|\mathbf{J}\|_{\mathcal{F}}^2+\lambda\|\mathbf{E}\|_{2,1}+\mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]\\&+\mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z}-\mathbf{J})]+\tfrac{\mu}{2}(\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2+\|\mathbf{Z}-\mathbf{J}\|_{\mathcal{F}}^2)\end{aligned} \tag{14}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are the Lagrange multipliers and $\mu>0$ is the penalty parameter. Each variable in optimization (14) can be addressed iteratively by updating $\mathbf{J}$, $\mathbf{Z}$, and $\mathbf{E}$ one-by-one. To solve this problem, we update each variable while fixing the others.

Update J. To update variable $\mathbf{J}$, by ignoring the irrelevant terms w.r.t. $\mathbf{J}$ in (14), we have the following objective:

$$\arg\min_{\mathbf{J}} \|\mathbf{J}\|_{\mathcal{F}}^2+\mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z}-\mathbf{J})]+\frac{\mu}{2}\|\mathbf{Z}-\mathbf{J}\|_{\mathcal{F}}^2 \tag{15}$$

According to Lemma 1, we can combine $\mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z}-\mathbf{J})]$ and $\frac{\mu}{2}\|\mathbf{Z}-\mathbf{J}\|_{\mathcal{F}}^2$, and (15) can be converted into

$$\begin{aligned}&\arg\min_{\mathbf{J}} \|\mathbf{J}\|_{\mathcal{F}}^2+\frac{\mu}{2}\|\mathbf{Z}-\mathbf{J}+\frac{\mathbf{Y}_2}{\mu}\|_{\mathcal{F}}^2\\&=\arg\min_{\mathbf{J}} \mathbf{J}^{\mathrm{T}}\mathbf{J}+\frac{\mu}{2}\langle\mathbf{J}-(\mathbf{Z}+\frac{\mathbf{Y}_2}{\mu}),\mathbf{J}-(\mathbf{Z}+\frac{\mathbf{Y}_2}{\mu})\rangle\end{aligned} \tag{16}$$

By specifying the derivative w.r.t. $\mathbf{J}$ to zero, we obtain

$$\mathbf{J}=\frac{\mu}{\mu+2}(\mathbf{Z}+\frac{\mathbf{Y}_2}{\mu}) \tag{17}$$

Update Z. To update variable $\mathbf{Z}$, by ignoring the irrelevant terms w.r.t. $\mathbf{Z}$ in (14), we have the following objective:

$$\begin{aligned}&\arg\min_{\mathbf{Z}} \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]+\mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z}-\mathbf{J})]\\&+\frac{\mu}{2}(\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2+\|\mathbf{Z}-\mathbf{J}\|_{\mathcal{F}}^2)\end{aligned} \tag{18}$$

By setting the partial derivative of (18) w.r.t. $\mathbf{Z}$ equal to zero, we obtain

$$\mathbf{Z}=(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{X}-\mathbf{E})+\mathbf{J}+(\mathbf{X}^{\mathrm{T}}\mathbf{Y}_1-\mathbf{Y}_2)/\mu) \tag{19}$$

Update E. To update variable $\mathbf{E}$, by ignoring the irrelevant terms w.r.t. $\mathbf{E}$ in (14), we have the following objective:

$$\begin{aligned}&\arg\min_{\mathbf{E}} \lambda\|\mathbf{E}\|_{2,1}+\mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})]+\frac{\mu}{2}(\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_{\mathcal{F}}^2)\\&=\arg\min_{\mathbf{E}} \frac{\lambda}{\mu}\|\mathbf{E}\|_{2,1}+\frac{1}{2}\|\mathbf{E}-(\mathbf{X}-\mathbf{XZ}+\frac{\mathbf{Y}_1}{\mu})\|_{\mathcal{F}}^2\end{aligned} \tag{20}$$

The solution to the problem is presented in [67]. Specifically, let us assume $\Psi=\mathbf{X}-\mathbf{XZ}+\frac{\mathbf{Y}_1}{\mu}$, where the kth column of $\mathbf{E}$ is given as

$$\mathbf{E}(:,\mathbf{k})=\begin{cases}\frac{\|\Psi_k\|-\frac{\lambda}{\mu}}{\|\Psi_k\|}\Psi_k, \text{if}\frac{\lambda}{\mu}<\|\Psi_k\|,\\0, \text{otherwise.}\end{cases} \tag{21}$$

Algorithm 1 Solving problem (14) by using the inexact ALM method.

Input: adjacency matrix of observed network $\mathbf{X}$, trade-off parameter λ.
Output: representation matrix $\mathbf{Z}$, error matrix $\mathbf{E}$.

Initial $\mathbf{Z}=\mathbf{J}=\mathbf{E}=\mathbf{0}$, $\mathbf{Y_1}=\mathbf{Y_2}=\mathbf{0}$, $\mu=10^{-6}$, $\max_\mu=10^6$, $\rho=1.1$, $\varepsilon=10^{-8}$;
while not converged **do**
  Fix the others and update $\mathbf{J}$ by (17);
  Fix the others and update $\mathbf{Z}$ by (19);
  Fix the others and update $\mathbf{E}$ by (20);
  // Update the multipliers as follows
  $\mathbf{Y_1}=\mathbf{Y_1}+\mu(\mathbf{X}-\mathbf{XZ}-\mathbf{E})$;
  $\mathbf{Y_2}=\mathbf{Y_2}+\mu(\mathbf{Z}-\mathbf{J})$;
  Update the parameter μ by $\mu=\min(\rho\mu,\max_\mu)$;
  // Check the convergence conditions
  $||\mathbf{X}-\mathbf{XZ}-\mathbf{E}||_\infty<\varepsilon$ and $|\mathbf{Z}-\mathbf{J})||_\infty<\varepsilon$;
end while

Algorithm 2 Link prediction algorithm LFLP.

Input: adjacency matrix $\mathbf{X}$ of observed network.
Output: detected missing link set $\mathbf{M}^+$ and detected spurious link set $\mathbf{M}^-$.

Obtain the optimal representation matrix $\mathbf{Z}^*$ via Algorithm 1;
Construct the adjacency matrix $\mathbf{SM}$ using (22);
Separate $\mathbf{SM}$ into the positive component $\mathbf{SM}^+$ and the negative component $\mathbf{SM}^-$ according to the entries' sign;
Remove the existing entries of $\mathbf{X}$ in $\mathbf{SM}^+$, and the remaining entries with the higher scores are more likely to be the missing links, which are saved in $\mathbf{M}^+$;
Sort the existing entries of $\mathbf{SM}^-$ by comparing it with $\mathbf{X}$, and the ones with the lower scores are more likely to be the spurious links, which are saved in $\mathbf{M}^-$.

The process of solving (14) is summarized in Algorithm 1.

Link Prediction. By learning the optimal representation matrix $\mathbf{Z}^*$ of the observed networks, the existence likelihoods of network links can be inferred by matrices $\mathbf{Z}^*$ and $\mathbf{X}$, i.e.,

$$\mathbf{SM} = \mathbf{XZ}^* + (\mathbf{XZ}^*)^{\mathrm{T}}. \tag{22}$$

Actually, the feasibility of the proposed link prediction method is based on the consistent patterns across subgraphs, where the corrupted local structure can be rectified based on the feature of the similar ones. All non-observed links are ranked according to their likelihoods, where the links with greater scores have a higher possibility to be missing links. Similarly, all observed links are ranked and the links with lower scores are more likely to be spurious links. The entire LFLP algorithm is presented in Algorithm 2.

4.3. Link predictability measuring and regulating

Because real-world networks always have certain regularity and their local structures, i.e., subgraphs, may have similar interaction patterns, we assume that the networks can be represented based on a subset of the subgraphs. That is, we assume that there is a subset of subgraphs, called representative subgraphs, such that each subgraph in the network can be described as a linear combination of the representative subgraphs. To explain the idea, Fig. 3 shows an example. To the network G, the node 1-centered subgraph can be represented by node 5- and node 9-centered subgraphs. That is, the column of matrix $\mathbf{X}$ corresponding to node 1 is a linear combination of the columns of $\mathbf{X}$ weighted by the entries of the first column of the representation matrix $\mathbf{Z}$, where only the entries corresponding to node 5 and node 9 are nonzero, as shown in the middle of Fig. 3. Similarly, the node 2-centered subgraph can be represented by node 4-, 6-, and 8-centered subgraphs. Based on the above assumption, the structural role of each subgraph for network representation can be captured by the rows of the learned representation matrix $\mathbf{Z}^*$, as shown in the top right corner of Fig. 3. Specifically, the entries of the nonzero rows of $\mathbf{Z}^*$ provide information about the relative importance of the subgraphs for network representation. A subgraph that is more representative takes part in the representation of many subgraphs in the network, and hence, its corresponding row in the optimal representation matrix $\mathbf{Z}^*$ has many nonzero elements. Meanwhile, a subgraph that is less representative takes part in the representation of fewer subgraphs in the network.

In real-world networks, individuals may have similar personal hobbies or political preferences, thereby generating multiple nodes with the same substructures. Because the substructures have the same structural roles for network representation, the potential representative subgraph can be any one of them. Thus, the more regular the networks are, the more redundant subgraphs are included, as shown by the zero rows in Fig. 4. Therefore, to represent a network structure with as few representative subgraphs as possible, the representation matrix $\mathbf{Z}$ must be low-rank. Similarly, the more regular the substructures are, the fewer representative subgraphs are needed to represent them, which corresponds to the number of nonzero entries of the representation matrix. Thus, to represent each substructure with as few representative subgraphs as possible, the representation matrix $\mathbf{Z}$ must be sparse.

Based on the aforementioned discussion, networks can be modeled via a low-rank and sparse representation as follows:

$$\min_{\mathbf{Z},\mathbf{E}} rank(\mathbf{Z}) + \alpha\|\mathbf{Z}\|_{\mathbf{0}} + \beta\|\mathbf{E}\|_{\mathbf{2,1}} \ \mathbf{s.t.}\ \mathbf{X} = \mathbf{XZ} + \mathbf{E}. \tag{23}$$

As this problem is NP-hard, as suggested by [65,73], we can solve the following relaxed convex program instead

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_* + \alpha\|\mathbf{Z}\|_{\mathbf{1}} + \beta\|\mathbf{E}\|_{\mathbf{2,1}} \ \mathbf{s.t.}\ \mathbf{X} = \mathbf{XZ} + \mathbf{E}. \tag{24}$$

where α and β are the trade-off parameters.

Optimization Solution. To solve the optimization problem, we introduce auxiliary variables $\mathbf{J}$ and $\mathbf{Q}$ to make the objective function separable. This problem can be converted as

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{J}\|_* + \alpha\|\mathbf{Q}\|_1 + \beta\|\mathbf{E}\|_{2,1},\ s.t., \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \mathbf{Z} = \mathbf{J}, \mathbf{Z} = \mathbf{Q} \tag{25}$$

which can be handled by solving the following ALM problem:

$$\begin{aligned} L(\mathbf{J},\mathbf{Z},\mathbf{E}) &= \|\mathbf{J}\|_* + \alpha\|\mathbf{Q}\|_1 + \beta\|\mathbf{E}\|_{2,1} + \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X} - \mathbf{XZ} - \mathbf{E})] \\ &+ \mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z} - \mathbf{J})] + \mathrm{tr}[\mathbf{Y}_3^{\mathrm{T}}(\mathbf{Z} - \mathbf{Q})] + \frac{\mu}{2}(\|\mathbf{X} - \mathbf{XZ} - \mathbf{E}\|_{\mathcal{F}}^2 \\ &+ \|\mathbf{Z} - \mathbf{J}\|_{\mathcal{F}}^2 + \|\mathbf{Z} - \mathbf{Q}\|_{\mathcal{F}}^2) \end{aligned} \tag{26}$$

where $\mathbf{Y}_1$, $\mathbf{Y}_2$, and $\mathbf{Y}_3$ are Lagrange multipliers and $\mu > 0$ is a penalty parameter. This problem can be solved by minimizing $\mathbf{J}$, $\mathbf{Q}$, $\mathbf{Z}$, and $\mathbf{E}$. By considering the efficiency, we choose the inexact ALM method.

Update J. To update variable $\mathbf{J}$, by ignoring the irrelevant terms w.r.t. $\mathbf{J}$ in (26), we have the following objective:

$$\begin{aligned} &\arg\min_{\mathbf{J}} \|\mathbf{J}\|_* + \mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z} - \mathbf{J})] + \frac{\mu}{2}\|\mathbf{Z} - \mathbf{J}\|_{\mathcal{F}}^2 \\ &= \arg\min_{\mathbf{J}} \|\mathbf{J}\|_* + \frac{\mu}{2}\|\mathbf{J} - (\mathbf{Z} + \frac{\mathbf{Y}_2}{\mu})\|_{\mathcal{F}}^2 \end{aligned} \tag{27}$$

This problem can be effectively solved by using the singular value thresholding (SVT) operator [74].

Update Q. To update variable $\mathbf{Q}$, by ignoring the irrelevant terms w.r.t. $\mathbf{Q}$ in (26), we have the following objective:

$$\begin{aligned} &\arg\min_{\mathbf{Q}} \alpha\|\mathbf{Q}\|_1 + \mathrm{tr}[\mathbf{Y}_3^{\mathrm{T}}(\mathbf{Z} - \mathbf{Q})] + \frac{\mu}{2}\|\mathbf{Z} - \mathbf{Q}\|_{\mathcal{F}}^2 \\ &= \arg\min_{\mathbf{Q}} \alpha\|\mathbf{Q}\|_1 + \frac{\mu}{2}\|\mathbf{Q} - (\mathbf{Z} + \frac{\mathbf{Y}_3}{\mu})\|_{\mathcal{F}}^2 \end{aligned} \tag{28}$$

This problem can be effectively solved by using the shrinkage operator [72].

Update Z. To update variable $\mathbf{Z}$, by ignoring the irrelevant terms w.r.t. $\mathbf{Z}$ in (26), we have the following objective:

$$\begin{aligned} &\arg\min_{\mathbf{Z}} \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X} - \mathbf{XZ} - \mathbf{E})] + \mathrm{tr}[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z} - \mathbf{J})] + \mathrm{tr}[\mathbf{Y}_3^{\mathrm{T}}(\mathbf{Z} - \mathbf{Q})] \\ &+ \frac{\mu}{2}(\|\mathbf{X} - \mathbf{XZ} - \mathbf{E}\|_{\mathcal{F}}^2 + \|\mathbf{Z} - \mathbf{J}\|_{\mathcal{F}}^2 + \|\mathbf{Z} - \mathbf{Q}\|_{\mathcal{F}}^2) \end{aligned} \tag{29}$$

By setting the partial derivative of (29) with respect to $\mathbf{Z}$ equal to zero,

$$\begin{aligned} \mathbf{Z} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + 2\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{X} - \mathbf{E}) + \ \mathbf{J} + \ \mathbf{Q} + (\mathbf{X}^{\mathrm{T}}\mathbf{Y}_1 \\ -\mathbf{Y}_2 - \mathbf{Y}_3)/\mu) \end{aligned} \tag{30}$$

Update E. Similar to the update operation in Algorithm 1, the objective about variable $\mathbf{E}$ is expressed as follows:

$$\begin{aligned} &\arg\min_{\mathbf{E}} \beta\|\mathbf{E}\|_{2,1} + \mathrm{tr}[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X} - \mathbf{XZ} - \mathbf{E})] + \frac{\mu}{2}(\|\mathbf{X} - \mathbf{XZ} - \mathbf{E}\|_{\mathcal{F}}^2) \\ &= \arg\min_{\mathbf{E}} \beta\|\mathbf{E}\|_{2,1} + \frac{\mu}{2}\|\mathbf{E} - (\mathbf{X} - \mathbf{XZ} + \frac{\mathbf{Y}_1}{\mu})\|_{\mathcal{F}}^2 \end{aligned} \tag{31}$$

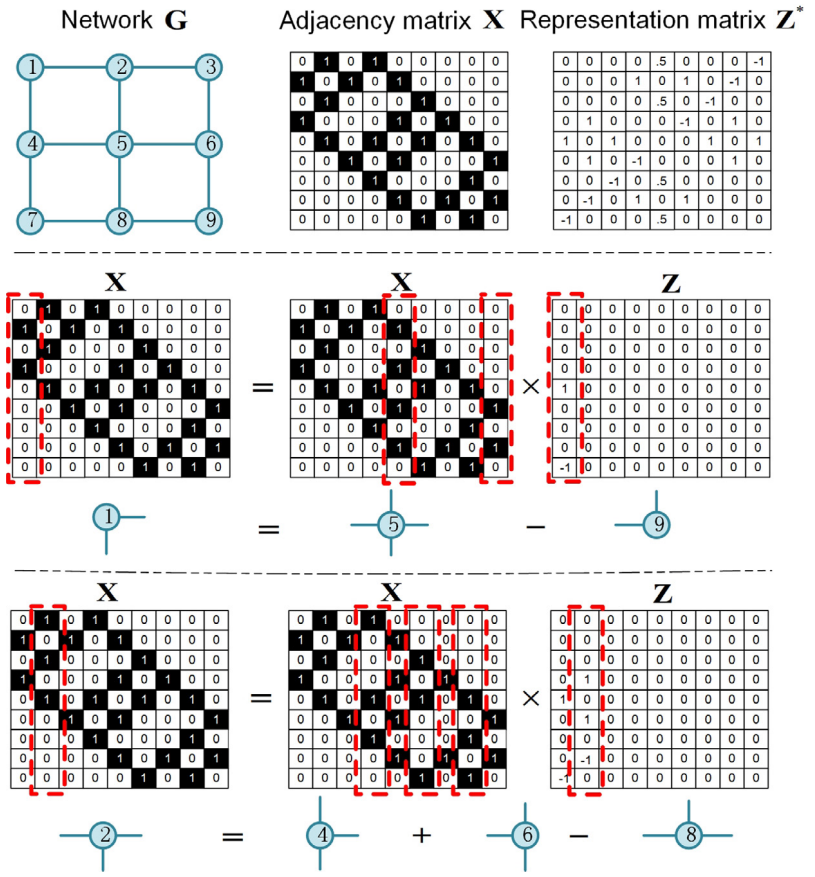


Fig. 3. Illustration of the self-representation network model.

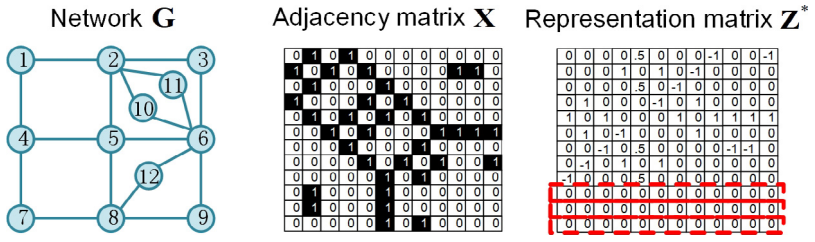


Fig. 4. Illustration of low-rank pursuit of the representation matrix.

The complete algorithm for solving problem (26) is outlined in Algorithm 3.

Link predictability measurement. Link predictability aims to quantify the extent to which a network can be modeled and predicted, which depends on the regularity level of networks. Based on the proposed self-representation model, the subgraphs of regular networks tend to be fully represented by some of the representative subgraphs. That is, the learned representation matrix $\mathbf{Z}^*$ captures the regularity of networks. First, the more the number of identical subgraphs included in the network, i.e., the higher the proportion of the regular components, the lower is the rank of the representation matrix $\mathbf{Z}^*$. Second, according to the self-representation network model, the more regular the networks, the fewer representative subgraphs are needed to represent them. For the representation matrix $\mathbf{Z}^*$, we can rank k subgraphs $\mathbf{X}_{:,i_1}, \mathbf{X}_{:,i_2}, \ldots, \mathbf{X}_{:,i_k}$ as $\mathbf{X}_{:,i_1} \geq \mathbf{X}_{:,i_2} \geq, \ldots, \geq \mathbf{X}_{:,i_k}$, i.e., the subgraph $\mathbf{X}_{:,i_1}$ is the most representative and $\mathbf{X}_{:,i_k}$ is the least representative, whenever for the corresponding rows of $\mathbf{Z}^*$, we have

$$\|\mathbf{Z}_{i_1,:}\|_1 \geq \|\mathbf{Z}_{i_2,:}\|_1 \geq, \ldots, \geq \|\mathbf{Z}_{i_k,:}\|_1 \tag{32}$$

where $\|\cdot\|_1$ indicates the ℓ_1 norm of vectors. Thus, the nonzero rows of matrix $\mathbf{Z}^*$ indicate the number of possible representative subgraphs. Third, the more regular the subgraphs, the fewer other subgraphs are needed to represent them, which can be characterized by the number of non-zero entries in $\mathbf{Z}^*$. Based on the aforementioned discussion, we define the structural regularity index σ_r for link predictability measurement as follows:

$$\sigma_r = \frac{1}{\sqrt{(n-r)/n}\sqrt{\tau/(n \cdot r)}} \tag{33}$$

where r is the rank of $\mathbf{Z}^*$, τ is the number of zero entries in $\mathbf{Z}^*$, $(n-r)/n$ denotes the proportion of identical subgraphs in the network, and $\tau/(n \cdot r)$ characterizes the density of zero entries of the reduced echelon form of matrix $\mathbf{Z}^*$.

Link predictability regulation. According to the learned representation matrix $\mathbf{Z}^*$, we can find that there are some links that frequently participate in the network self-representation and others that are rarely used. That is, the links play different structural roles in the network organization and have various influences on network regularity. Thus, the link predictability can

Algorithm 3 Solving Formula (26) by using the inexact ALM method.

Input: adjacency matrix of observed network $\mathbf{X}$, trade-off parameter λ.
Output: representation matrix $\mathbf{Z}$, error matrix $\mathbf{E}$.

Initial $\mathbf{Z} = \mathbf{J} = \mathbf{E} = \mathbf{0}$,$\mathbf{Y_1} = \mathbf{Y_2} = \mathbf{0}$,$\mu = 10^{-6}$,$\max_\mu = 10^6$, ρ = 1.1, $\varepsilon = 10^{-8}$;
while not converged **do**
  Fix the others and update $\mathbf{J}$ by (27);
  Fix the others and update $\mathbf{Q}$ by (28);
  Fix the others and update $\mathbf{Z}$ by (30);
  Fix the others and update $\mathbf{E}$ by (31);
  // Update the multipliers as follows
  $\mathbf{Y_1} = \mathbf{Y_1} + \mu(\mathbf{X} - \mathbf{XZ} - \mathbf{E})$;
  $\mathbf{Y_2} = \mathbf{Y_2} + \mu(\mathbf{Z} - \mathbf{J})$;
  $\mathbf{Y_3} = \mathbf{Y_3} + \mu(\mathbf{Z} - \mathbf{Q})$;
  Update the parameter μ by $\mu = \min(\rho\mu, \max_\mu)$;
  // Check the convergence conditions
  $||\mathbf{X} - \mathbf{XZ} - \mathbf{E}||_\infty < \varepsilon$ and $||\mathbf{Z} - \mathbf{J}||_\infty < \varepsilon$ and $||\mathbf{Z} - \mathbf{Q}||_\infty < \varepsilon$;
end while

be regulated based on important link-based structural perturbation. Thus, how to identify the roles of network links and measure their importance in terms of network regularity becomes a key problem.

For a representative subgraph, more subgraphs in regular networks can be represented by them than that in irregular networks. That is, there is a large degree of commonality between the subgraphs in regular networks. Thus, based on the learned representation matrix $\mathbf{Z}^*$, we can define the regularity degree of the subgraph-centered node k as follows:

$$RD(k) = \sum_{i=1}^{n} \frac{\mathbf{Z}_{i,k}}{\|\mathbf{Z}_{:,k}\|_1} \tag{34}$$

Thus, the importance of links for network self-representation can be estimated by the related subgraphs, i.e.,

$$U_{ij} = RD(i) \times RD(j) \tag{35}$$

Algorithm 4 LPR algorithm.

Input: network $G = (V, E)$ and its adjacency matrix $\mathbf{X}$, the number of perturbed links w.
Output: the regulated network G^R.

Learn the representation matrix $\mathbf{Z}^*$ using Algorithm 3;
for each network node $k \in V$ **do**
  Calculate the regularity level $RD(k)$ of the subgraph-centered node k using (34);
end for
for each network link $(i, j) \in E$ **do**
  Calculate the importance U_{ij} of link (i, j) using (35);
end for
Sort the importance scores $\{U_{ij}\}$ and obtain the ranked list $\mathbf{U}$;
$i = 0$;
while $i < w$ **do**
  Remove the regular network link $\mathbf{U}[i]$;
  $i = i + 1$;
end while

The metric quantifies the potential influence of link (i, j) in both directions. The links with a smaller value of U_{ij} are more likely to be regular links. Otherwise, they are more likely to be irregular links. By employing the importance of network links, link predictability can be regulated by important link-based structure perturbation. The whole structure perturbation-based LPR algorithm is given in Algorithm 4.

4.4. Theoretical analysis and convergence analysis

The proposed self-representation-based link prediction algorithm involves the following: (1) solving the representation matrix $\mathbf{Z}^*$ with the inexact ALM method in Algorithm 1 and (2) predicting the network links with matrix multiplication in Algorithm 2. Specifically, Algorithm 1 performs matrix addition when solving $\mathbf{J}$. Moreover, Algorithm 1 performs matrix inversion, addition, and multiplication in solving $\mathbf{Z}^*$ and $\mathbf{E}$. We can observe that the most time-consuming components of Algorithm 1 are the matrix multiplications and inverse in Steps 4 and 5. Each matrix multiplication costs close to $\mathcal{O}(n^3)$ and the matrix inverse takes $\mathcal{O}(n^3)$ for $n \times n$ matrixes. Therefore, Algorithm 1 costs nearly $(k+1)\mathcal{O}(n^3)$ in total, when there are k multiplication operations. In Algorithm 2, the time cost mainly comes from constructing the adjacency matrix in step 2 and identifying the missing and spurious links in steps 4 and 5. We can find that the most time-consuming component of Algorithm 2 is the matrix multiplication in step 2. Because the numbers of missing and spurious links are very limited, the complexity of the link identification can be ignored. Thus, the complexity of Algorithm 2 is $\mathcal{O}(n^3)$.

For LPMR, on the basis of Algorithm 3, the structure perturbation algorithm is proposed in Algorithm 4. Specifically, the time-consuming components concentrate on the following steps:

- The trace norm computation in step 3 of Algorithm 3.
- The matrix inversion and multiplication in steps 5 and 6 of Algorithm 3.
- The sorting operation about the link importance scores in step 6 of Algorithm 4.

The conventional SVD of an $n \times n$ matrix has time complexity $\mathcal{O}(n^3)$. It will be time-consuming if n is large, i.e., the number of data samples is large. Fortunately, the SVD of an $n \times n$ matrix can be accelerated to $\mathcal{O}(rn^2)$, where r is the rank of the matrix, by using the recent fast low-rank method [75]. In addition, the computation complexity of matrix inversion and multiplication computation costs nearly $(k+1)\mathcal{O}(n^3)$ in total, where k is the number of matrix multiplication operations. Finally, the time complexity of the sorting operation is $\mathcal{O}(m\log m)$, where m is the number of network links.

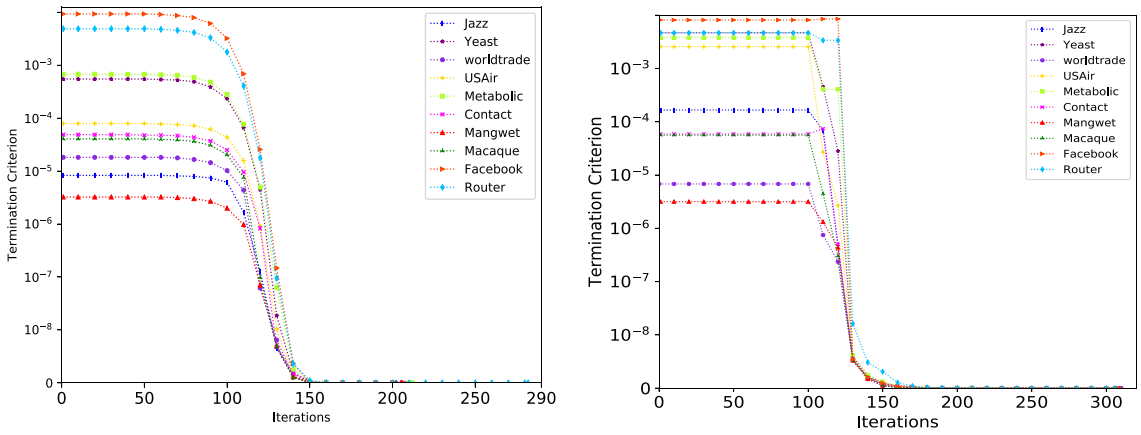
To experimentally show the convergence behavior of Algorithms 1 and 3, for simplicity, we provide an intuitive curve illustration of the convergence with respect to the iteration number on 10 datasets, as shown in Fig. 5. The results show that Algorithms 1 and 3 converge in around 220 and 310 steps, respectively, indicating that our optimization methods exhibit a good convergence property.

5. Experiments

We conduct an experimental study of the proposed algorithm based on real-world networks. Three sets of experiments are conducted to evaluate the performance of the proposed methods, including the link prediction algorithm, link predictability measure, and link predictability regulation algorithm.

5.1. Experimental setup

We consider the following 10 real-world networks drawn from disparate fields: (i) Jazz [76], a collaboration network of jazz musicians; (ii) Worldtrade [77], a network of miscellaneous



(a) Convergence curve of Algorithm 1.

(b) Convergence curve of Algorithm 3.

Fig. 5. Convergence curve of the proposed algorithms. For Algorithm 1, the value of the termination criterion is $\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_\infty < \varepsilon$ and $\|(\mathbf{Z}-\mathbf{J})\|_\infty < \varepsilon$. For Algorithm 3, the value of the termination criterion is $\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_\infty < \varepsilon$ and $\|\mathbf{Z}-\mathbf{J}\|_\infty < \varepsilon$ and $\|\mathbf{Z}-\mathbf{Q}\|_\infty < \varepsilon$.

Table 2
Statistical features of networks. The features include network size N, link number M, average node degree $\langle k\rangle$, maximum node degree $k_{\max}$, clustering coefficient C, assortative coefficient r, and degree heterogeneity H, $H = \frac{\langle k^2\rangle}{\langle k\rangle^2}$. For the sampled subnetwork, N of the original network is shown in the bracket.

Networks	N	M	$<k>$	k_{max}	C	r	H
Jazz	198	2742	27.69	100	0.617	0.020	1.395
Worldtrade	80	788	19.70	72	0.700	-0.391	1.640
Contact	264	2108	15.96	101	0.657	-0.479	3.546
Metabolic	453	2040	9.006	239	0.646	-0.219	4.478
Mangwet	97	1446	29.81	90	0.468	-0.150	1.265
Macaque	94	1515	32.23	74	0.773	-0.150	1.238
USAir	332	2126	12.80	139	0.625	-0.207	3.464
Facebook	1000(56952)	25459	50.92	922	0.587	-0.020	2.040
Router	1000(5022)	1346	2.692	106	0.039	-0.412	8.409
Yeast	1000(2361)	5578	11.15	90	0.357	0.430	2.664

manufactures of metal among 80 countries in 1994; (iii) Contact [78], a contact network between people measured by the carried wireless devices; (iv) Metabolic [79], a metabolic network of C.elegans; (v) Mangwet [80], the food web in Mangrove Estuary during the wet season; (vi) Macaque [81], the cortical networks of the macaque monkey; (vii) USAir [82], the US Air transportation network; (viii) Facebook [83], a directed network of a small subset of posts to other user's wall on Facebook. Here, we treat it as a simple graph by ignoring the directions and weights; (ix) Router [84], a symmetrized snapshot of the structure of the Internet at the level of autonomous systems; (x) Yeast [85], a protein–protein interaction network in budding yeast. The statistical features of the networks are summarized in Table 2.

5.2. Link prediction evaluation

To evaluate the performance of the proposed link prediction algorithm, we introduce five link prediction methods for comparison. The simplest is the common neighbor (CN) [16], where two nodes have a higher connecting probability if they have more common neighbors. An improved method based on CN is resource allocation (RA) [34], which assigns more weight to less-connected neighbors. In addition, we compare the proposed prediction algorithm LFLP with three global link prediction methods, including SPM [59], NMF [45], RPCA [46], and LO [19]. Their details are described as follows:

(1) Common Neighbor (CN): The CN metric is one of the most widespread measurements used in the link prediction problem. For two nodes, x and y, CN is defined as the number of nodes that both x and y have a direct interaction with. A greater number of the common neighbors makes it easier to create a link between x and y. This measure is defined as

$$\mathrm{CN}(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{36}$$

(2) Resource Allocation (RA): This metric is proposed by Zhou et al. [34], and is motivated by the physical processes of resource allocation. The RA metric suppresses the contribution of the high-degree common neighbors, and is defined as

$$\mathrm{RA}(x,y) = \sum_{z\in|\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{37}$$

(3) Non-negative Matrix Factorization (NMF) [45]: a method that learns the latent features from graphs for structure prediction.

$$A^* = \frac{1}{R}\sum\nolimits_{r=1}^{r=R} W^{(r)}H^{(r)} \tag{38}$$

where A^* is the reconstruction matrix of the original graph based on the basis matrix W^r and coefficient matrix H^r.

(4) Structural Perturbation Method (SPM) [59]: a global algorithm based on eigen decomposition.

$$s^{SPM} = \sum\nolimits_{k=1}^{N}(\lambda_k + \Delta\lambda_k)x_k x_k^T \tag{39}$$

where λ_k, x_k, and $\Delta\lambda_k$ are the eigenvalue, eigenvector, and disturbing quantity of the eigenvalue, respectively.

(5) Robust Principal Component Analysis (RPCA) [46]: an RPCA-based structure prediction method.

$$\arg\min_{\mathbf{X}^*,\mathbf{E}} rank(\mathbf{X}^*) + \gamma\|E\|_0 \; s.t. \; \mathbf{X}^* = \mathbf{A} - \mathbf{E} \tag{40}$$

where $rank(\mathbf{X}^*)$ denotes the rank of matrix $\mathbf{X}^*$, and the operator $\|\cdot\|_0$ is the ℓ_0 norm.

(6) Linear Optimization (LO) [19]: assumes that the likelihood of the existence of an unobserved link between node i to node j, denoted by f_{ij}, can be unfolded by a linear summation of contributions from i's neighbors, namely

$$f_{ij} = \sum_k a_{ik}z_{kj} \tag{41}$$

where z_{kj} is the contribution from node k to node j.

Table 3
AUC (top half) and Precision (bottom half) of the link prediction algorithms for missing link prediction. Each value is averaged over 20 independent runs with 10% random links as a probe set. The parameters of the methods are tuned to their optimal values. The best results are emphasized in bold, and the values in bracket are the standard deviation.

Algorithm	Jazz	World trade	Contact	Metabolic	Mangwet	Macaque	USAir	Facebook	Router	Yeast
CN	0.951(0.010)	0.875(0.032)	0.937(0.010)	0.913(0.017)	0.712(0.032)	0.945(0.008)	0.952(0.017)	0.932(0.016)	0.585(0.032)	0.895(0.015)
RA	0.966(0.013)	0.887(0.028)	0.927(0.024)	0.904(0.014)	0.714(0.039)	0.932(0.024)	0.895(0.016)	**0.943(0.018)**	0.590(0.021)	0.893(0.010)
NMF	0.951(0.017)	0.902(0.018)	0.938(0.020)	**0.944(0.010)**	0.863(0.025)	0.958(0.019)	0.968(0.017)	0.899(0.029)	0.700(0.015)	0.920(0.024)
SPM	**0.970(0.014)**	0.915(0.031)	0.923(0.022)	0.864(0.020)	0.908(0.026)	0.980(0.007)	0.968(0.016)	0.902(0.032)	0.619(0.074)	0.929(0.019)
RPCA	0.861(0.026)	0.858(0.044)	0.877(0.035)	0.586(0.049)	0.881(0.025)	0.959(0.018)	0.860(0.041)	0.723(0.030)	0.591(0.069)	0.800(0.023)
LO	0.945(0.020)	0.903(0.024)	0.930(0.018)	0.787(0.028)	0.914(0.030)	0.980(0.006)	0.926(0.032)	0.905(0.020)	0.721(0.053)	0.944(0.019)
LFLP	0.964(0.009)	**0.941(0.021)**	**0.949(0.016)**	0.845(0.035)	**0.948(0.019)**	**0.987(0.008)**	**0.973(0.020)**	0.908(0.025)	**0.740(0.051)**	**0.946(0.010)**
CN	0.531(0.021)	0.397(0.031)	0.552(0.022)	0.133(0.032)	0.120(0.023)	0.536(0.023)	0.376(0.018)	0.287(0.011)	0.010(0.001)	0.302(0.007)
RA	0.514(0.019)	0.409(0.032)	0.560(0.014)	0.131(0.032)	0.135(0.023)	0.509(0.023)	0.436(0.017)	0.410(0.012)	0.029(0.028)	0.335(0.007)
NMF	0.452(0.020)	0.415(0.023)	0.510(0.027)	0.235(0.022)	0.444(0.029)	0.652(0.024)	0.406(0.034)	0.405(0.079)	0.219(0.023)	0.473(0.017)
SPM	0.579(0.024)	0.420(0.021)	0.578(0.021)	0.217(0.056)	0.450(0.037)	0.700(0.030)	0.405(0.025)	0.415(0.059)	0.190(0.022)	0.507(0.013)
RPCA	**0.612(0.025)**	0.425(0.025)	**0.607(0.016)**	0.223(0.020)	0.549(0.039)	0.755(0.024)	0.420(0.003)	0.372(0.010)	0.104(0.017)	0.541(0.015)
LO	0.547(0.024)	0.385(0.026)	0.323(0.015)	0.237(0.027)	0.514(0.032)	0.745(0.017)	0.291(0.018)	0.297(0.053)	0.210(0.018)	0.596(0.014)
LFLP	0.592(0.022)	**0.474(0.021)**	0.604(0.018)	**0.324(0.026)**	**0.565(0.034)**	**0.768(0.020)**	**0.448(0.021)**	**0.419(0.062)**	**0.251(0.021)**	**0.603(0.022)**

Table 4
AUC (top half) and Precision (bottom half) of the link prediction algorithms for spurious link identification. Each value is averaged over 20 independent runs. The parameters of the methods are tuned to their optimal values. The best results are emphasized in bold. The values in bracket are the standard deviation.

Algorithm	Jazz	World trade	Contact	Metabolic	Mangwet	Macaque	USAir	Facebook	Router	Yeast
CN	0.956(0.011)	0.882(0.034)	0.929(0.019)	0.885(0.027)	0.685(0.021)	0.899(0.044)	0.941(0.020)	**0.972(0.024)**	0.584(0.016)	0.915(0.011)
RA	**0.971(0.021)**	0.877(0.021)	0.924(0.022)	**0.904(0.013)**	0.668(0.030)	0.892(0.034)	0.935(0.016)	0.953(0.018)	0.557(0.017)	0.912(0.022)
NMF	0.917(0.020)	0.913(0.015)	0.971(0.013)	0.829(0.012)	0.903(0.018)	0.947(0.029)	0.934(0.011)	0.965(0.029)	0.920(0.027)	0.823(0.024)
SPM	0.886(0.019)	0.804(0.027)	0.654(0.041)	0.628(0.036)	0.725(0.024)	0.864(0.023)	0.657(0.036)	0.077(0.022)	0.353(0.042)	0.783(0.034)
RPCA	0.919(0.015)	0.892(0.025)	0.979(0.007)	0.685(0.034)	0.902(0.020)	0.935(0.018)	0.896(0.016)	0.910(0.020)	0.581(0.019)	0.913(0.017)
LO	0.944(0.019)	0.906(0.027)	0.977(0.008)	0.846(0.025)	0.905(0.025)	**0.953(0.017)**	0.935(0.015)	0.936(0.024)	0.905(0.010)	0.938(0.007)
LFLP	0.947(0.014)	**0.928(0.026)**	**0.987(0.008)**	0.857(0.023)	**0.906(0.018)**	0.942(0.019)	**0.947(0.006)**	0.940(0.028)	**0.951(0.010)**	**0.943(0.017)**
CN	0.569(0.027)	0.548(0.055)	0.233(0.018)	0.167(0.023)	0.288(0.030)	0.571(0.024)	0.248(0.024)	0.776(0.015)	0.022(0.002)	0.073(0.007)
RA	0.541(0.018)	0.539(0.038)	0.245(0.014)	0.140(0.027)	0.138(0.026)	0.549(0.023)	0.265(0.032)	**0.693(0.012)**	0.177(0.008)	0.111(0.011)
NMF	0.651(0.023)	0.588(0.043)	0.686(0.022)	**0.243(0.028)**	0.566(0.029)	0.809(0.022)	0.482(0.026)	0.636(0.039)	0.589(0.004)	0.233(0.017)
SPM	0.358(0.029)	0.260(0.031)	0.116(0.020)	0.160(0.020)	0.206(0.027)	0.476(0.032)	0.166(0.021)	0.263(0.023)	0.069(0.001)	0.252(0.017)
RPCA	0.592(0.021)	0.535(0.032)	0.754(0.025)	0.074(0.025)	0.518(0.035)	0.810(0.022)	0.443(0.022)	0.466(0.012)	0.037(0.007)	0.425(0.023)
LO	0.602(0.010)	0.582(0.051)	0.800(0.023)	0.169(0.019)	0.567(0.034)	**0.879(0.025)**	0.549(0.020)	0.533(0.023)	0.398(0.018)	0.547(0.008)
LFLP	**0.663(0.020)**	**0.621(0.034)**	**0.810(0.024)**	0.221(0.021)	**0.583(0.036)**	0.818(0.022)	**0.597(0.019)**	0.682(0.022)	**0.618(0.026)**	**0.568(0.009)**

To test the validity of the link prediction algorithms, we first select 10% of the network links as the missing link set E^M (the probe set) and use the remaining 90% as the training set E^T. The results of the missing link prediction measured by AUC and the Precision are shown in Table 3. All data points are obtained by averaging over 20 implementations with an independently random division of the training set and missing link set, and the values in bracket are the standard deviation. For each network, the bold number in the corresponding column emphasizes the highest accuracy. The results in Table 3 indicate that the proposed LFLP method generally performs the best among the state-of-the-art algorithms. To evaluate the effectiveness of network reconstruction algorithms for spurious link identification, 10% spurious links (the probe set) are added randomly to each real network to construct the observed network. The results for spurious link identification measured by AUC and Precision are shown in Table 4. For all networks, our method LFLP performs the best among the state-of-the-art algorithms, usually remarkably better than the second best. A possible reason is that the low Frobenius norm-constrained model adopted by our work has a greater expressive capability than the other methods. For the network Metabolic and Facebook, LFLP does not perform very well in the experiments. The possible reason behind the results is that the self-representation level of the networks is poor. According to Table 2, we can find that the average node degree of Metabolic and Facebook and their maximum node degree are quite different, which indicates that the networks have strong structural heterogeneity and one of the local substructures of the networks cannot be represented effectively by the others.

5.3. Link predictability measuring evaluation

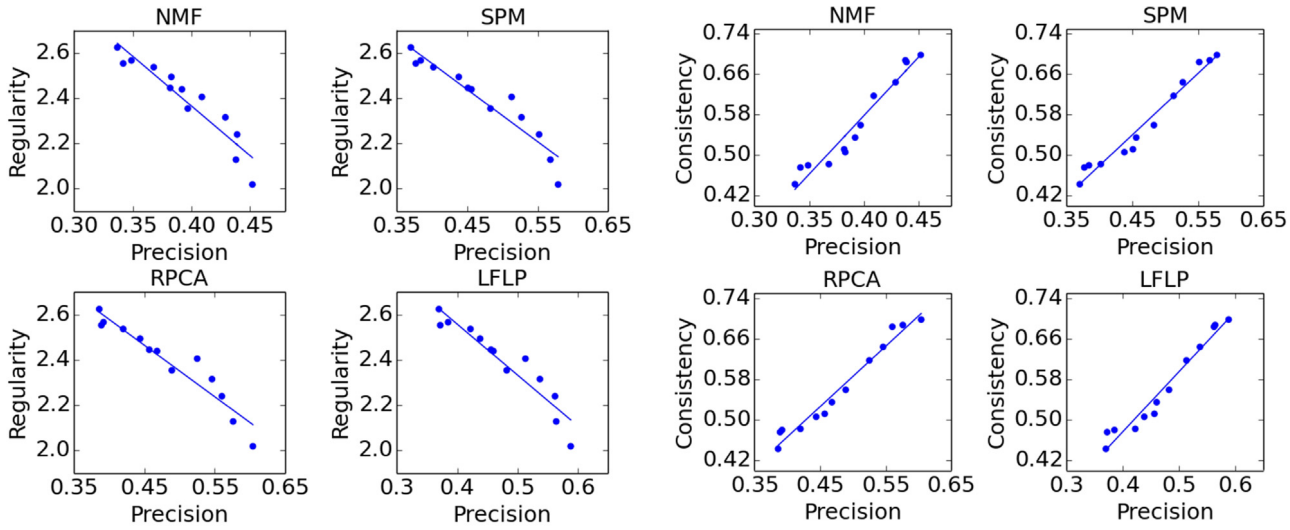
According to the results in Tables 3 and 4, a specified link prediction algorithm performs differently across the networks. The reason behind the different prediction accuracy on the networks is that the networks possess diversified link predictability. Thus, link predictability becomes a critical property for network analysis. To evaluate the effectiveness of the defined structural regularity index for link predictability measurement, we compare the values of the structural regularity index with the precisions of the four representative link prediction algorithms. Moreover, for comparison, the structural consistency [59] is introduced:

$$\sigma_c = \frac{|E^L \cap \Delta E|}{|\Delta E|} \tag{42}$$

That is, structural consistency σ_c is defined as the fraction of common links between ΔE and E^L. ΔE is the perturbation link set, and E^L is the set of top-L-ranked links.

First, we perturb the Jazz network by removing different percentages of links randomly, from 1% to 12%, to generate multiple regulated networks with different levels of structural regularity. For each regulated network, we estimate its regularity level and calculate the precision of the four representative link prediction algorithms. The scatter plot between the structural regularity values and the prediction precision of the Jazz network is shown in Fig. 6(a). We can find that the higher the link prediction precision, the smaller is the regularity value (a higher level of structural regularity). The structural consistency of the Jazz network under different degrees of perturbation is also presented in Fig. 6(b). We can observe a positive correlation between the structural consistency and precision of link prediction. Therefore, both network regularity and structural consistency can be used to indicate the link predictability of networks. However, there is actually a phase of link prediction in the calculation of the structural consistency index, while the structural regularity index is calculated by mining the network structure directly.

To evaluate the proposed structural regularity index adequately, we apply it to all real-world networks. The experimental results are shown in Fig. 7, where each value is the average over the results generated from NMF, SPM, RPCA, and LFLP. In general, the value of structural regularity is correlated with the average precision of link prediction in the networks, indicating that a higher regularity level will result in greater link predictability



(a) Structural regularity for link predictability measurement. (b) Structural consistency for link predictability measurement.

Fig. 6. Scatter plot between link predictability and prediction precision of Jazz network. For (a), the y-axis indicates the structural regularity of the network, for (b), the y-axis indicates the network consistency and the x-axis indicates the precision of link prediction methods in networks perturbed with varied percent. The solid lines indicate the linear fittings of the results. The smaller the values of structural regularity, the more regular are the networks.

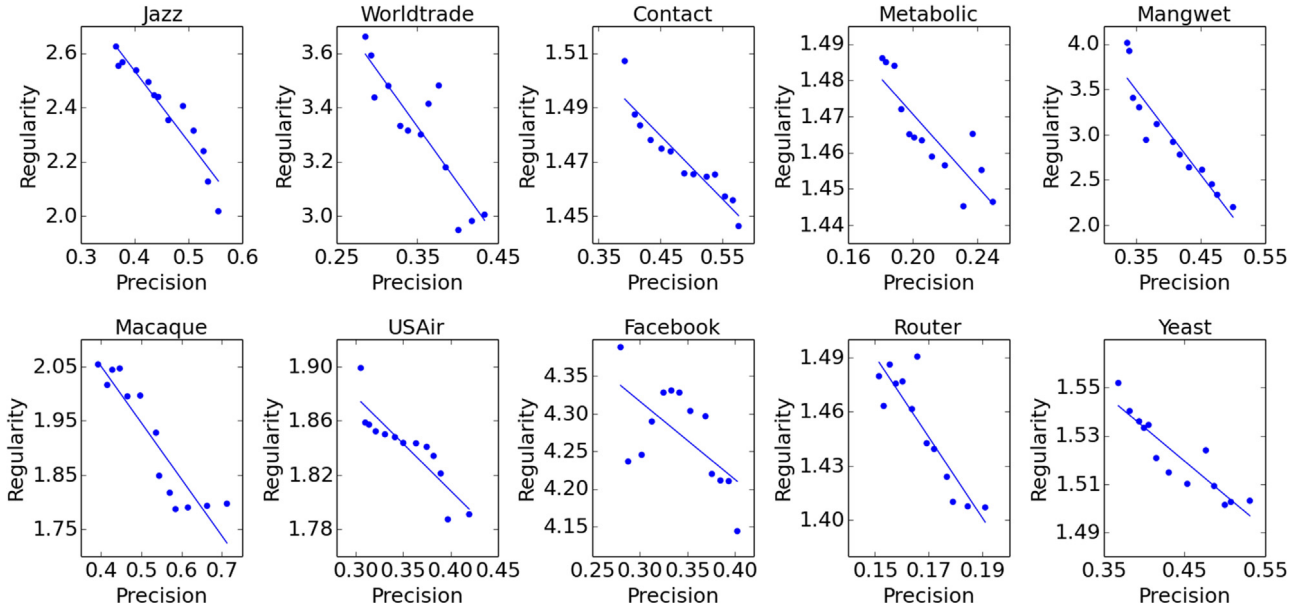


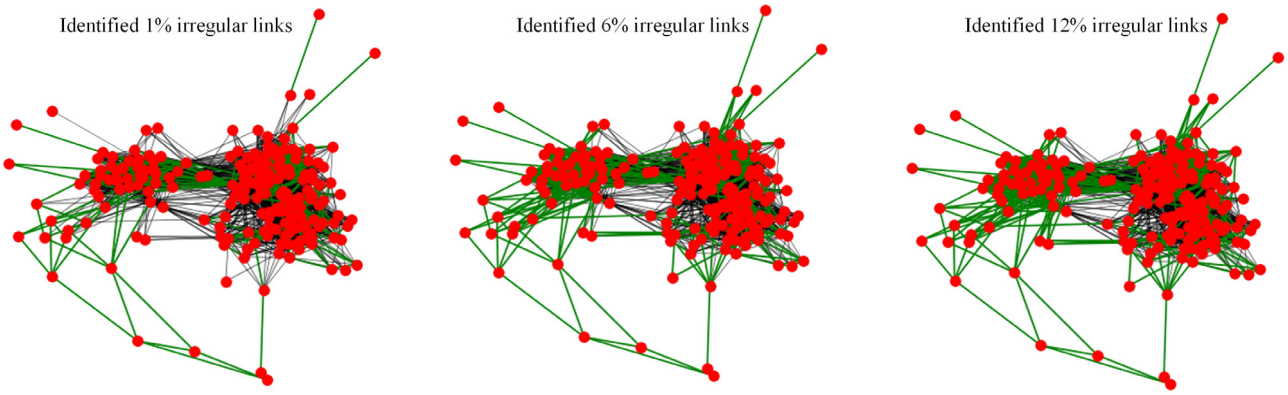
Fig. 7. Scatter plot between the defined structural regularity index and the link prediction precision in real-world networks. Each precision value is the average over the algorithms NMF, SPM, RPCA, and LFLP. The solid lines are the linear fittings of the results. The smaller the values of structural regularity, the more regular are the networks.

of networks. Thus, the results on the networks verify the effectiveness of the proposed structural regularity index σ_r for link predictability measurement.

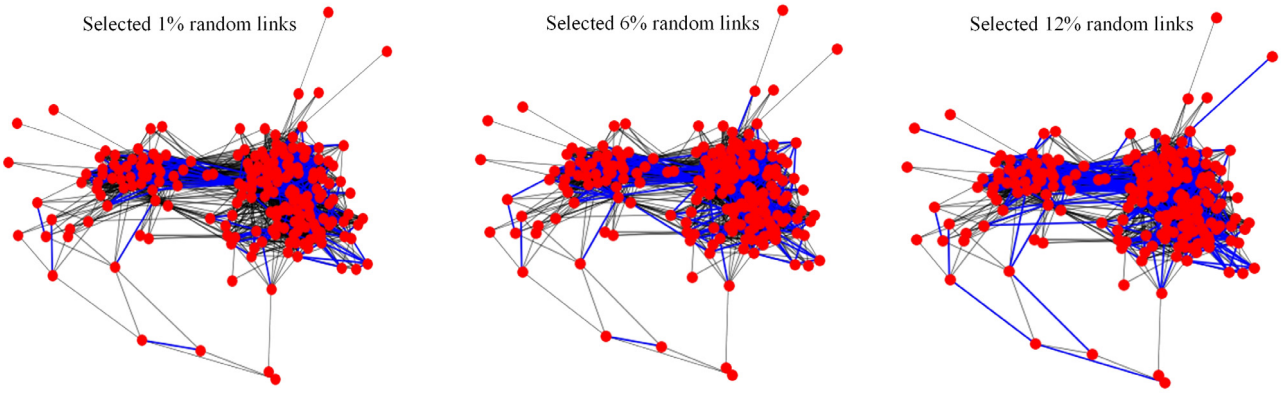
5.4. Link predictability regulation evaluation

To explore the influence of network links on link predictability, we first identify the regular and irregular links based on the proposed link importance metric. Generally, the regular links have high substitutability in network self-representation, while the irregular links have few equivalent links in network self-representation. Moreover, we adopt here a random mechanism for link selection to regulate the networks' link predictability.

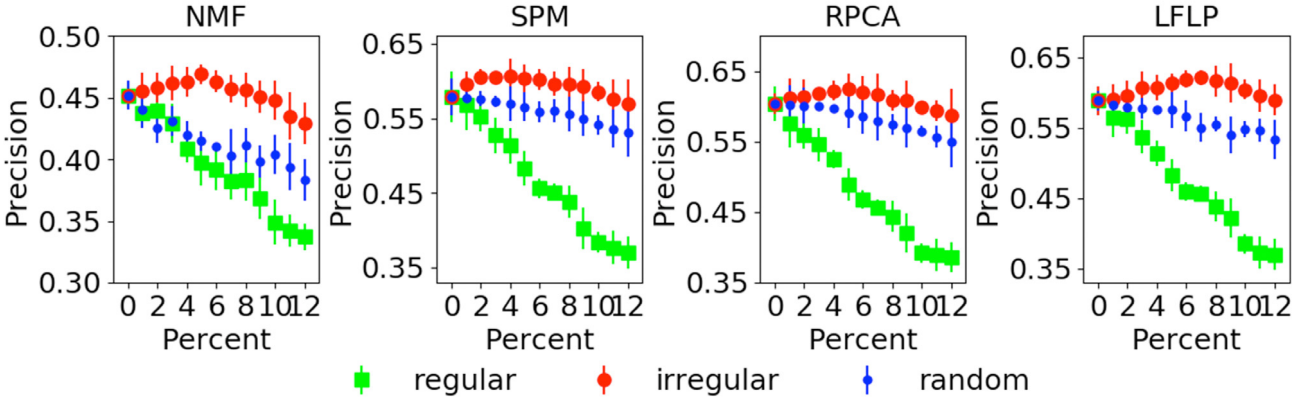
To understand the structural roles of network links deeply, we apply the three link selection mechanisms to the Jazz network and analyze their influence in detail. Fig. 8(a) presents the identified irregular network links (in green color) for percentages 1, 6, and 12. The black links shown in Fig. 8(a) have a higher level of regularity compared to the irregular links in green. Fig. 8(b) shows the links selected randomly (in blue color) for percentages 1, 6, and 12. Compared to the links selected randomly in Fig. 8(b), the identified irregular links in Fig. 8(a) are more likely to be the weak links between the periphery nodes of the network and are difficult to be modeled with link prediction methods. One explanation for the preference is that the excessive sparsity of the subgraphs of the periphery nodes makes the related links unable to form regular structural patterns. Moreover, the links with high regularity level gather at the network core and are more likely to generate regular structures. Fig. 8(c) shows the link prediction precision of NMF, SPM, RPCA, and LFLP on the



(a) Identified irregular links for link predictability regulation.



(b) The links selected randomly for link predictability regulation.



(c) Link predictability regulation of Jazz network.

Fig. 8. Link predictability regulation in network Jazz. The links with green color in (a) are the irregular links selected based on the proposed link importance index, and those in black have a higher regularity level. The blue links in (b) are the randomly selected ones. The result in (c) shows the precision of link prediction algorithms under various percentages of the removed network links.

regulated networks with varied perturbation ratio. Fig. 8(c) shows that, for link predictability regulation, neither the irregular links nor the regular links work as effectively as the links selected randomly. More interestingly, there is a range in which the link prediction precision can be improved via removal of irregular links.

To probe into the problem of link predictability regulation, all real-world networks are regulated with varied perturbation ratios and various links. In each regulated network, NMF, SPM, RPCA, and LFLP are adopted for link prediction, and the average prediction precision of the algorithms is shown in Fig. 9. As shown in the figure, the prediction precision can be improved by removing irregular links. This improvement implies that the structural regularity of networks can be strengthened by irregular link-based structure perturbation. As the number of removed links continues to increase, the network sparsity would increase and lead to the reduction of the prediction precision. By applying the regular link-based structure perturbation mechanism, the precision of link prediction degrades continuously with an increase in the

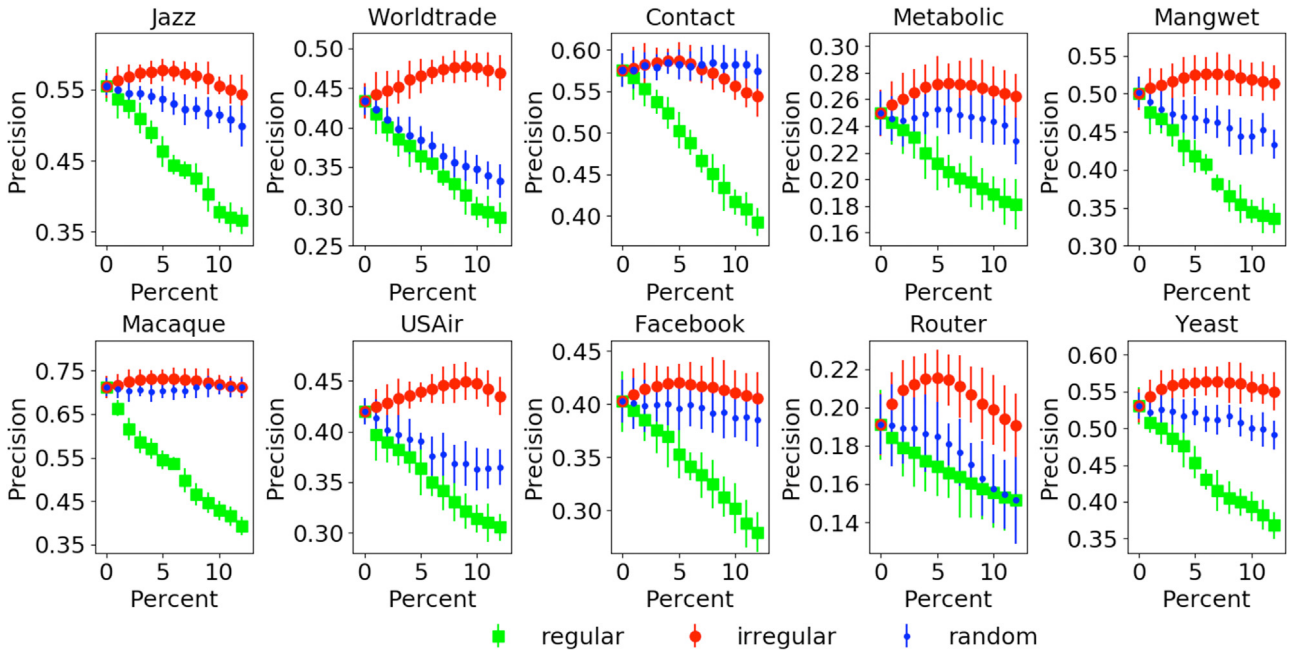


Fig. 9. Link predictability regulation with network link-based structure perturbation. The results show the average precision of the link prediction algorithms NMF, SPM, RPCA, and LFLP under various percentages of links removed from 1% to 9%. The error bars represent the standard deviations of the prediction precision.

percentage of the removed regular links. Similarly, as the random link-based structure perturbation continues, the link prediction precisions on the networks decrease gradually. By comparing the results corresponding to the three types of network links, we can find that the regular link-based structure perturbation has less adverse influence on the prediction precision than the random link-based perturbation. This is because the regular links generally have more equivalent links than the random links and their removal has less impact on the regularity level of networks.

According to the above discussion, network links can be categorized into multiple types with different influences on the link predictability of networks. Compared to regular links, the random link-based structure perturbation can reduce networks' link predictability more effectively, which is consistent with the random anonymization strategy in privacy preserving [86]. For the link predictability-regulating task, the regular link-based perturbation may not have a significant effect. Moreover, real-world networks often have different levels of regularity, and their link predictability can be improved based on the irregular link-based structure perturbation. In addition, the results of Fig. 9 show that, in networks with high prediction precision, there is not much room for irregular link perturbation-based predictability improvement.

6. Conclusions and discussion

This paper introduces the LPMR problem. Theoretically, exploring and controlling the link predictability of networks is of significance in network analysis and graph mining. Link predictability can be used to indicate the expected link prediction accuracy of networks. Moreover, exploring link predictability can help us uncover the organization principle of networks and understand the structural roles of network links. From the practical viewpoint, via irregular link identification, the abnormal social relations in online social networks can be detected and help in identifying zombie accounts. In data mining applications, detecting outliers via link role learning is critical for data preprocessing. In social networks, the anonymized sensitive social relationships may be disclosed with link prediction, and regulating networks' predictability with critical link-based perturbation can enhance the guarantee level of privacy preserving.

To explore the LPMR problem, we present a self-representation model for network structure description, where the networks are represented as a linear combination of the local subgraphs. The model allows us to explore the organization principle of networks by analyzing the intercommunity of subgraphs in global network organization. By applying the self-representation model in link prediction, the LFLP algorithm is proposed. The experimental results show that LFLP is quite effective compared to the state-of-the-art algorithms. Based on the self-representation model, we define a structural regularity index to measure the intrinsic link predictability of networks. Moreover, we define a link importance metric from the perspective of network self-representation, thereby classifying network links into different types. Finally, the influence of the links on network predictability is explored. Overall, we believe that our study provides a fundamental framework for link predictability research.

However, as we need to solve the self-representation model for LPMR, the proposed methods on a larger-scale network are quite difficult because of the time complexity limitation. Therefore, improving the proposed methods or introducing new theories for reducing computation time are good topics for future studies.

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.

CRediT authorship contribution statement

Xingping Xian and Tao Wu contributed the conceptualization, data curation, methodology and wrote the original draft. The remaining authors contributed to validating the ideas, carrying out additional analyses and reviewing this paper. All authors read and approved the manuscript.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61802039, 61772098, 61772091, 61802035; National Key R&D Program of China under Grant No. 2018YFB0904900, 2018YFB0904905; Science and Technology Research Program of Chongqing Municipal Education Commission, China under Grant No. KJQN201800630; Innovative Talents Program, China under Grant No. BYJS201811.

References

[1] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[2] F. Bonchi, C. Castillo, A. Gionis, A. Jaimes, Social network analysis and mining for business applications, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–22.
[3] M. Newman, Network structure from rich but noisy data, Nat. Phys. (2018) 1.
[4] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Modern Phys. 74 (1) (2002) 47.
[5] A.-L. Barabási, Scale-free networks: a decade and beyond, science 325 (5939) (2009) 412–413.
[6] G. Kossinets, D.J. Watts, Empirical analysis of an evolving social network, Science 311 (5757) (2006) 88–90.
[7] M.E.J. Newman, A. Clauset, Structure and inference in annotated networks, Nature Commun. 7 (2–3) (2015) 11863.
[8] M.E.J. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2) (2004) 321–330.
[9] L. Peel, D.B. Larremore, A. Clauset, The ground truth about metadata and community detection in networks, Sci. Adv. 3 (5) (2017) e1602548.
[10] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016) 1–63.
[11] T. Wu, L. Chen, L. Zhong, X. Xian, Enhanced collective influence: A paradigm to optimize network disruption, Physica A 472 (2017) 43–52.
[12] T. Wu, X. Xian, L. Zhong, X. Xiong, H.E. Stanley, Power iteration ranking via hybrid diffusion for vital nodes identification, Physica A 506 (2018) 802–815.
[13] H. Wang, P. Zhang, X. Zhu, I. Tsang, L. Chen, C. Zhang, X. Wu, Incremental subgraph feature selection for graph classification, IEEE Trans. Knowl. Data Eng. 29 (1) (2017) 128–142.
[14] J.T. Vogelstein, R.W. Gray, R.J. Vogelstein, C.E. Priebe, Graph classification using signal-subgraphs: applications in statistical connectomics, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1539–1551.
[15] D. Koutra, U. Kang, J. Vreeken, C. Faloutsos, Summarizing and understanding large graphs, Stat. Anal. Data Min. 8 (3) (2015) 183–202.
[16] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Assoc. Inf. Sci. Technol. 58 (7) (2007) 1019–1031.
[17] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (6) (2011) 1150–1170.
[18] T. Wu, L. Chen, L. Zhong, X. Xian, Predicting the evolution of complex networks via similarity dynamics, Physica A 465 (2017) 662–672.
[19] R. Pech, D. Hao, Y.-L. Lee, Y. Yuan, T. Zhou, Link prediction via linear optimization, Physica A 528 (2019) 121319–121348.
[20] P. Jiao, C. Fei, Y. Feng, W. Wang, Link predication based on matrix factorization by fusion of multi class organizations of the network, Sci. Rep. 7 (1) (2017) 8937.
[21] B. Baruch, B. Albert-Løszl, Network link prediction by global silencing of indirect correlations, Nature Biotechnol. 31 (8) (2013) 720–725.
[22] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, S. Yang, Detecting suspicious following behavior in multimillion-node social networks, in: Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 305–306.
[23] M. L Linyuan Mats, H.Y. Chi, Y.C. Zhang, Z.K. Zhang, T. Zhou, Recommender systems, Phys. Rep. 519 (1) (2012) 1–49.
[24] A. Anil, D. Kumar, S. Sharma, R. Singha, R. Sarmah, N. Bhattacharya, S.R. Singh, Link prediction using social network analysis over heterogeneous terrorist network, in: IEEE International Conference on Smart City/Socialcom/Sustaincom, 2016, pp. 267–272.
[25] E. Zheleva, L. Getoor, Preserving the privacy of sensitive relationships in graph data, in: International Workshop on Privacy, Security, and Trust in KDD, 2007, pp. 153–171.
[26] X. Ying, X. Wu, On link privacy in randomizing social networks, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2009, pp. 28–39.
[27] Y. Zhang, M. Humbert, B. Surma, P. Manoharan, J. Vreeken, M. Backes, CTRL+Z: Recovering anonymized social graphs, 2017, arXiv preprint arXiv:1711.05441.
[28] M. Fire, G. Katz, L. Rokach, Y. Elovici, Links reconstruction attack, in: Security and Privacy in Social Networks, Springer, 2013, pp. 181–196.
[29] S. Nilizadeh, A. Kapadia, Y.-Y. Ahn, Community-enhanced de-anonymization of online social networks, in: Proceedings of the 2014 Acm Sigsac Conference on Computer and Communications Security, 2014, pp. 537–548.
[30] K. Shu, S. Wang, J. Tang, R. Zafarani, H. Liu, User identity linkage across online social networks: A review, Acm Sigkdd Explor. Newsl. 18 (2) (2017) 5–17.
[31] X. Zhou, X. Liang, X. Du, J. Zhao, Structure based user identification across social networks, IEEE Trans. Knowl. Data Eng. 30 (6) (2018) 1178–1191.
[32] E.C. Mutlu, T.A. Oghaz, Review on graph feature learning and feature extraction techniques for link prediction, 2019, arXiv preprint arXiv:1901.03425.
[33] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (3) (2003) 211–230.
[34] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (4) (2009) 623–630.
[35] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.
[36] G. Jeh, J. Widom, Simrank: a measure of structural-context similarity, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 538–543.
[37] W. Liu, L. Lü, Link prediction based on local random walk, Europhys. Lett. 89 (5) (2010) 58007–58012.
[38] H. Gao, J. Huang, Q. Cheng, H. Sun, B. Wang, H. Li, Link prediction based on linear dynamical response, Physica A 527 (2019) 121397.
[39] S. Rafiee, C. Salavati, A. Abdollahpouri, CNDP:link prediction based on common neighbors degree penalization, Physica A 539 (2020) 122950.
[40] M. Zhang, Y. Chen, Weisfeiler-Lehman neural machine for link prediction, in: Acm Sigkdd International Conference on Knowledge Discovery and Data Mining ACM, 2017, pp. 575–583.
[41] R. Guimer, M. Salespardo, Missing and spurious interactions and the reconstruction of complex networks, Proc. Natl. Acad. Sci. USA 106 (52) (2009) 22073.
[42] A. Clauset, C. Moore, M.E. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (7191) (2008) 98.
[43] L. Pan, T. Zhou, L. Linyuan, C.K. Hu, Predicting missing links and identifying spurious links via likelihood analysis, Sci. Rep. 6 (2016) 22955.
[44] A.K. Menon, C. Elkan, Link prediction via matrix factorization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2011, pp. 437–452.
[45] W. Wang, F. Cai, P. Jiao, P. Lin, A perturbation-based framework for link prediction via non-negative matrix factorization, Sci. Rep. 6 (2016) 38938.
[46] R. Pech, D. Hao, L. Pan, H. Cheng, T. Zhou, Link prediction via matrix completion, Europhys. Lett. 117 (3) (2017) 38002.
[47] Z. Wang, L. Jiye, L. Ru, A fusion probability matrix factorization framework for link prediction, Knowl.-Based Syst. 159 (2018) 72–85.
[48] 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.
[49] M. Zhang, Y. Chen, Link prediction based on graph neural networks, Adv. Neural Inf. Process. Syst. (2018) 5165–5175.
[50] S. Harada, H. Akita, M. Tsubaki, Y. Baba, I. Takigawa, Y. Yamanishi, H. Kashima, Dual convolutional neural network for graph of graphs link prediction, 2018, arXiv preprint arXiv:1810.02080.
[51] T. Li, J. Zhang, S.Y. Philip, Y. Zhang, Y. Yan, Deep dynamic network embedding for link prediction, IEEE Access 6 (2018) 29219–29230.
[52] N. Przulj, D.G. Corneil, I. Jurisica, Modeling interactome: scale-free or geometric? Bioinformatics 20 (18) (2004) 3508–3515.
[53] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (2002) 824–827.
[54] N.K. Ahmed, J. Neville, R.A. Rossi, N. Duffield, Efficient graphlet counting for large networks, in: IEEE International Conference on Data Mining, 2015, pp. 1–10.
[55] R.A. Rossi, Z. Rong, N.K. Ahmed, Estimation of graphlet statistics, 2017, arXiv preprint arXiv:1701.01772.
[56] A.R. Benson, D.F. Gleich, J. Leskovec, Higher-order organization of complex networks, Science 353 (6295) (2016) 163–166.
[57] D. Koutra, U. Kang, J. Vreeken, C. Faloutsos, Vog: Summarizing and understanding large graphs, Stat. Anal. Data Min. 8 (3) (2015) 183–202.
[58] S. Hua-Wei, C. Xue-Qi, G. Jia-Feng, Exploring the structural regularities in networks, Phys. Rev. E 84 (5) (2011) 056111.
[59] L. Lü, L. Pan, T. Zhou, Y.C. Zhang, H.E. Stanley, Toward link predictability of complex networks, Proc. Natl. Acad. Sci. USA 112 (8) (2015) 2325–2330.
[60] D.-B. Chen, G.-N. Wang, A. Zeng, Y. Fu, Y.-C. Zhang, Optimizing online social networks for information propagation, PLoS One 9 (5) (2014) e96614.
[61] J. Ash, D. Newth, Optimizing complex networks for resilience against cascading failure, Physica A 380 (2007) 673–683.

[62] Z. Wang, D. Zhao, L. Wang, G. Sun, Z. Jin, Immunity of multiplex networks via acquaintance vaccination, Europhys. Lett. 112 (4) (2015) 48002.
[63] Y. Liu, M. Tang, T. Zhou, Y. Do, Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics, Sci. Rep. 5 (2015) 13172.
[64] I.T. Jolliffe, Principal component analysis, J. Mark. Res. 87 (100) (1986) 513.
[65] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 11.
[66] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2790–2797.
[67] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: International Conference on Machine Learning, 2010, pp. 663–670.
[68] J. Yang, X.D. Zhang, Predicting missing links in complex networks based on common neighbors and distance, Sci. Rep. 6 (2016) 38208.
[69] J.H. Abawajy, M.I.H. Ninggal, T. Herawan, Privacy preserving social network data publication, IEEE Commun. Surv. Tutor. 18 (3) (2016) 1974–1997.
[70] W.H. Lee, C. Liu, S. Ji, P. Mittal, R.B. Lee, Blind de-anonymization attacks using social networks, in: ACM Workshop on Privacy in the Electronic Society, 2017, pp. 1–4.
[71] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3501–3508.
[72] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, 2010, arXiv preprint arXiv: 1009.5055.
[73] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, Siam Rev. 52 (3) (2007) 471–501.
[74] J.F. Cai, E.J.S. Cand, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2008) 1956–1982.
[75] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.
[76] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 6 (04) (2003) 565–573.
[77] D.A. Smith, D.R. White, Structure and dynamics of the global economy: Network analysis of international trade 1965-1980, Soc. Forces 70 (4) (1992) 857–893.
[78] J. Kunegis, Konect: the koblenz network collection, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 1343–1350.
[79] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 027104.
[80] D. Baird, J. Luczkovich, R.R. Christian, Assessment of spatial and temporal variability in ecosystem attributes of the St Marks National Wildlife Refuge, Apalachee Bay, Florida, Estuar. Coast. Shelf Sci. 47 (3) (1998) 329–349.
[81] L. da F Costa, M. Kaiser, C.C. Hilgetag, Predicting the connectivity of primate cortical networks from topological and spatial node properties, BMC Syst. Biol. 1 (2007) 16.
[82] R.A. Rossi, N.K. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 4292–4293.
[83] B. Viswanath, A. Mislove, M. Cha, K.P. Gummadi, On the evolution of user interaction in facebook, in: Proceedings of the 2nd ACM Workshop on Online Social Networks, ACM, 2009, pp. 37–42.
[84] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with rocketfuel, ACM SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 133–145.
[85] 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, Nucl. Acids Res. 31 (9) (2003) 2443–2450.
[86] A.M. Fard, K. Wang, Neighborhood randomization for link privacy in social network analysis, World Wide Web 18 (2013) 9–32.