Expert Systems With Applications 188 (2022) 116033

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A novel link prediction algorithm based on inductive matrix completion

Zhili Zhao *, Zhuoyue Gou, Yuhong Du, Jun Ma, Tongfeng Li, Ruisheng Zhang

School of Information Science & Engineering, Lanzhou University, 730000 Lanzhou, China

ARTICLE INFO

Keywords:
Link prediction
Dimension reduction
Matrix completion
Feature construction
Feature selection

ABSTRACT

Link prediction refers to predicting the connection probability between two nodes in terms of existing observable network information, such as network structural topology and node properties. Although traditional similarity-based methods are simple and efficient, their generalization performance varies widely in different networks. In this paper, we propose a novel link prediction approach *ICP* based on inductive matrix completion, which recoveries node connection probability matrix by applying node features to a low-rank matrix. The approach first explores a comprehensive node feature representation by combining different structural topology information with node importance properties via feature construction and selection. The selected node features are then used as the input of a supervised learning task for solving the low-rank matrix. The node connection probability matrix is finally recovered by a bi-linear function, which predicts the connection probability between two nodes with their features and the low-rank matrix. In order to demonstrate the *ICP* superiority, we took eleven related efforts including two recent methods proposed in 2020 as baseline methods, and it is shown that *ICP* has stable performance and good universality in twelve different real networks. Compared with the baseline methods, the improvements of *ICP* in terms of the average AUC results are ranging from 3.81% ~ 12.77% and its AUC performance is improved by 0.08% ~ 3.54% compared with the best baseline method. The limitation of *ICP* lies in its high computational complexity due to the feature construction, but the complexity can be reduced by replacing complex features with node semantic attributes if there are additional data available. Moreover, it provides a potential link prediction solution for large-scale networks, since inductive matrix completion is a supervised learning task, in which the underlying low-rank matrix can be solved by representative nodes instead of all their nodes.

1. Introduction

Link prediction is one of the basic problems in complex network analysis and it attempts to estimate the connection probability between two nodes according to existing structural topology information and node properties of a network. Over the last few decades, link prediction has been extensively used in many domains, such as recommendation system (Bag et al., 2019; Li et al., 2020; Liu, 2022; Su et al., 2020), knowledge graph completion (Nayyeri et al., 2021), criminal discovery (Assouli et al., 2021), question–disease relation identification (Kaya & Gündoğan, 2018) and scientific research trend prediction (Behrouzi et al., 2020). More precisely, a large number of links in protein–protein interaction (PPI) networks are still unknown and a link prediction algorithm is helpful in guiding PPI verification experiments in terms of the prediction results, thereby accelerating the process of PPI detection experiments. In social networks, link prediction can predict which users may be friends based on the current network structure, and plays an important role in recommending friends with the same interests. Moreover, the functions such as "guess you like" and "selected recommendation" of online social network platforms can also be regarded as one of link prediction applications. If the prediction is accurate, there is no doubt that it will increase user experience and loyalty to such platforms.

In general, existing link prediction methods can be broadly classified into four main categories: similarity-based methods, probabilistic methods, dimension reduction-based methods and hybrid methods. The similarity-based methods are traditional and also the simplest ones due to their simplicity and lower computational time. In such methods, the connection probability between two nodes is calculated based on existing structural topology information and/or node properties. The similarity-based methods can be further classified into: local indices, global indices and quasi-local indices (Daud et al., 2020; Kumar et al., 2020). The limitation of the similarity-based methods mainly lies in their simplicity, which cannot identify underlying potential patterns in complex networks. The probabilistic methods focus on building

* Corresponding author.
E-mail addresses: zhaozhl@lzu.edu.cn (Z. Zhao), gouzhy19@lzu.edu.cn (Z. Gou), yhdu19@lzu.edu.cn (Y. Du), maj19@lzu.edu.cn (J. Ma), litf19@lzu.edu.cn (T. Li), zhangrs@lzu.edu.cn (R. Zhang).

https://doi.org/10.1016/j.eswa.2021.116033
Received 14 November 2020; Received in revised form 15 August 2021; Accepted 2 October 2021
Available online 9 October 2021

probabilistic models to map several parameters to the connection probability between two nodes nicely. Generally, the probabilistic methods are more complex than the similarity-based methods since they require more information including structural topology information, node and edge properties and semantic attributes to build the probabilistic models. Maximum likelihood methods can also be regarded as the probabilistic methods, since they aim to estimate the parameters for link prediction models. In the link prediction domain, dimension reduction can be used to map higher complex dimensional network space to lower dimensional space by preserving both the microscopic structure (pairwise node similarity) and mesoscopic topological structure. Network embedding and matrix factorization are two main examples of dimension reduction. However, the limitation of the network embedding-based link prediction lies in accurate representation of network information and matrix factorization needs constraints to improve the prediction performance. Besides such methods, there are hybrid link prediction models that either employ data analytic techniques (such as machine learning) or exploit the benefits of integrating multiple related models. The hybrid methods often involves hyper parameters, which are important for improving the performance, but they often need to be optimized in terms of empirical experience.

In this paper, we propose a novel link prediction approach *ICP* based on inductive matrix completion, which recoveries the node connection probability matrix by applying node features to a low-rank matrix. The approach can be regarded as a kind of dimension reduction-based method, but the difference can be found that it employs a comprehensive node feature representation by combining different structural topology information with node importance properties via feature construction and selection. The selected node features are then used as the input of a supervised learning task for solving the low-rank matrix. The node connection probability matrix is finally generated by a bilinear function with their features and the low-rank matrix. In order to demonstrate the superiority of *ICP*, we used eleven related efforts including two recent methods proposed in 2020 as baseline methods, and the results showed that *ICP* had stable performance and good universality in twelve different real networks. Compared with the baseline methods, the improvements of *ICP* in terms of the average AUC results are ranging from 3.81% $\sim$ 12.77%. Moreover, the AUC performance is improved by 0.08% $\sim$ 3.54% compared with the best baseline method. The limitation of *ICP* lies in its high computational complexity due to the feature construction, but the complexity can be reduced by replacing complex features with node semantic attributes if there are additional data available. Moreover, *ICP* depends on three hyper parameters, which are important for the *ICP* performance. In this work, we optimize such parameters based on empirical experience to obtain the optimal values for each network, but in real applications, we can obtain optimal parameters by splitting existing datasets into training set and test set and tuning the parameters in terms of the AUC results. The other benefit of *ICP* is that it provides a potential link prediction solution for large-scale networks, since inductive matrix completion is a supervised learning task, in which the underlying low-rank matrix can be solved by representative nodes instead of all their nodes.

The rest of the paper is organized as follows. In Section 2, a brief literature review of different link prediction methods is presented. In Section 3, traditional link prediction methods used the baseline methods are given first. Moreover, two proposed link prediction methods based on both standard matrix completion and inductive matrix completion are presented. In Section 4, the network datasets and the evaluation metrics that are used to evaluate the link prediction methods involved in this paper are provided. In Section 5, the evaluation results from different perspectives are presented. Section 6 discusses the superiority of *ICP* and its limitations. Section 7 summarizes the achievements and highlights of this paper.

2. Related work

Over the last few decades, link prediction has been extensively studied and there are many efficient approaches that have been proposed to predict the connection probability between two nodes in given complex networks. In general, existing methods can be broadly classified into four main categories: similarity-based methods, probabilistic methods, dimension reduction-based and hybrid methods. Different from the surveys which present link prediction techniques, applications and performance at large (Daud et al., 2020; Kumar et al., 2020), we will give a brief overview of the strengths and weaknesses of such approaches as well as the differences and innovations of the methods proposed in this work in what follows.

2.1. Similarity-based methods

The similarity-based methods are traditional and also the simplest link prediction methods due to their simplicity and lower computational time. In such methods, the connection probability between two nodes i, j of a network is based on a similarity score $s(i,j)$, which is calculated based on existing structural network topology and/or node properties. The common similarity approaches can be mainly classified into three types: local indices, global indices and quasi-local indices. The local index-based similarity methods include the prominent ones, such as common neighbor (CN) (Rafiee et al., 2020), Adamaic Adar (AA) (Rafiee et al., 2020), resource allocation (RA) (Liu et al., 2017), Jaccard Coefficient (Bag et al., 2019). CN assumes that if two nodes have many common neighbors, their connection probability is increased (Rafiee et al., 2020). This is because that, in general, the nodes in the same circle are closely connected (there are more common friends), while the nodes in different circles are sparsely connected (there are fewer common friends) (Li et al., 2020). The local index-based similarity methods are the most straightforward link prediction approaches and have low computational complexity since they only consider the local neighborhood information of nodes (i.e., node neighbors) and are very suitable for large-scale networks. However, they often have lower accuracy due to the limited information considered. The global index-based similarity methods calculate similarity scores among node pairs in terms of global network structural information, e.g., the path distance (which is usually longer than two) between two nodes. The global structural topology information can be obtained from the adjacency matrix of a network. The classical global index-based approaches are Katz index (Vural & Kaya, 2018), Leicht–Holme–Newman (LHN) global index (Wahid-Ul-Ashraf et al., 2019) and random walk-based methods, such as random walk with restart (RWR) (Zhou et al., 2021), average commute time (ACT) (Wahid-Ul-Ashraf et al., 2019) and Cos^+ (Kumar et al., 2020). Katz considers all paths between two nodes and punishes long paths exponentially, i.e., assigning shorter paths with higher weights. LHN assumes that two nodes are similar to each other if either of them has an immediate neighbor, which is similar to the other node. The random walk-based methods consider the link structures by treating all nodes of a network equivalently, and ignore the centrality of nodes and have shown outstanding performance in link prediction (Li et al., 2011; Liu et al., 2015). However, such approaches are usually time-consuming since they consider entire structural topology information of a network, although they have better accuracy. There are also trade-off efforts that employ both local and global information and such efforts are also known as the quasi-local index-based similarity methods. Examples of the approaches include local path (LP) index (Aziz et al., 2020), local random walk (LRW) index (Song et al., 2019). CCPA is also a quasi-local similarity method, which estimates the connection probability between two nodes of a network based on the number of common neighbors and their closeness centrality (Ahmad et al., 2020). Moreover, CNDP estimates the similarity score between two nodes by their common neighbors and the network average clustering coefficient (Rafiee et al., 2020).

Compared with other common neighbor-based link prediction methods, CNDP treats each common neighbors differently. Such quasi-local index-based approaches not only have better performance, but also have lower computational complexity. On the whole, the limitation of the similarity-based methods mainly lies in their simplicity. They cannot identify underlying potential patterns in complex networks and their performance is highly dependent on datasets and applications.

2.2. Probabilistic methods

The probabilistic methods focus on building probabilistic models to map several parameters to the connection probability between two nodes nicely. In other words, specified with the parameters, the models predict the connection probability between two nodes by calculating a probability value. Existing efforts such as, Javari et al. proposed a probabilistic model for link prediction based on both local and global structures of a network (Javari et al., 2017). The model has demonstrated high tolerance toward the sparsity problem, i.e., it relies more on global structures if the sparsity increases and it gives more weights to local structures when the sparsity decreases. Zhang et al. proposed a general intermediary probability model for link prediction based on four indexes indicating different structural features (Zhang et al., 2018). The experiments on real-world networks showed that the model had more precise prediction. Different from the efforts which only considered the adjacency matrix, Stanley et al. proposed a probabilistic model SBM, in which nodes had an associated vector of continuous attributes for community structure in networks (Stanley et al., 2019). The results showed that the attributed SBM was helpful for link prediction and collaborative filtering. Das et al. proposed a Markov model for link prediction over the time-varying graph of dynamic social networks, and the results showed that the Markov model had better prediction accuracy than other two dynamic approaches (Das & Das, 2017). Moreover, Zhao et al. proposed a Bayesian probabilistic approach NARM that incorporated various kinds of node attributes for link prediction on both directed and undirected relational networks (Zhao et al., 2017). Generally, the probabilistic methods require more information including structural topology information, node and edge properties and semantic attributes to build the probabilistic models. Maximum likelihood methods can also be regarded as the probabilistic methods, since they aim to estimate the parameters for link prediction models. The parameters maximize the likelihood that is calculated by link prediction methods based on actually observed data. For example, Gaucher et al. established a non-asymptotic bound on the risk of the maximum likelihood estimator of network connections probabilities (Gaucher & Klopp, 2021). Kuang et al. presented a maximum a posteriori estimation based model to reconstruct a specific layer in the multilayer network (Kuang & Scoglio, 2021). However, the maximum likelihood methods are usually complex and time-consuming and are not suitable for large-scale networks (Kumar et al., 2020).

2.3. Dimension reduction-based methods

Dimension reduction is often used in machine learning and it can transform data from high-dimensional space into low-dimensional space while preserving almost all features in the original space. In the link prediction domain, dimension reduction can be used to map higher complex dimensional network space to lower dimensional space by preserving both the microscopic structure (pairwise node similarity) and mesoscopic topological structure. One common dimension reduction technique used for link prediction is network embedding, which aims to specify the components of a network with low-dimensional vectors, in which the structural topology information and properties of the network are maximally preserved (Zhao et al., 2021). Network embedding is widely used not only in link prediction, but also in other network analysis domains, this is because low-dimensional vectors captured by network embedding can be easily processed by various classical machine learning methods. Example efforts such as Chen et al. proposed a projected metric embedding model PME based on the metric learning to capture both first-order and second-order proximity in a unified way (Chen et al., 2018). Wang et al. presented a deep-learning-based network embedding framework to extract user latent representations from heterogeneous networks and predict the sign of unobserved sentiment links (Wang et al., 2018). The limitation of the network embedding-based link prediction lies in accurate representation of the network information.

Moreover, matrix factorization has been regarded as a constrained low rank approximation method for dimension reduction. Matrix factorization solves the link prediction problem as a matrix representation and carries out the prediction by taking the product of two lower dimensional matrices (Daud et al., 2020). However, although matrix factorization is often used for link prediction, in reality, there are constraints that have been imposed on the process to improve the prediction performance (Ma et al., 2017; Wang et al., 2017; Wu & Chen, 2016). Among such efforts, non-negative matrix factorization (NMF) has been often used for link prediction. For example, the NMF-based link prediction model DeepEye proposed by Ahmed et al. learned latent features from the temporal and topological structure of dynamic networks and achieved better accuracy prediction results (Ahmed et al., 2018). Different from general matrix factorization, NMF requires that W and H must be two non-negative matrices. In addition, Pech et al. regarded link prediction as a matrix completion problem from the incomplete adjacency matrix of a network, and proposed LR, which decomposed the adjacency matrix into a low-rank matrix containing the true links and a sparse error matrix containing spurious links based on robust principal component analysis (Pech et al., 2017). The network was finally recovered based on the low-rank matrix and the results showed that LR was effective when target networks were connected and sufficiently dense.

2.4. Hybrid methods

Besides the methods mentioned above, there are hybrid link prediction models that either employ data analytic techniques, such as machine learning or exploit the benefits of integrating multiple related models. For example, Liu et al. proposed a novel link prediction method LLSLP based on stacked generalization (Liu & Li, 2021). Different from traditional link prediction methods based on single similarity indexes, LLSLP regarded link prediction as a binary classification problem and took fifteen similarity indexes including CN, RA, AA, LP, Katz, ACT, Cos^{+}, RWR, etc. as the features between two nodes to build Logistic Regression and LightGBM-based basic models first, respectively. Furthermore, Logistic-regression and LightGBM were integrated by stacked generalization. The results on ten different networks showed that LLSLP had better accuracy and generalization. Li et al. proposed a generative model GTRBM which combined temporal restricted Boltzmann machine with gradient boosting decision tree for link prediction in dynamic networks (Li et al., 2018). GTRBM not only integrated topological features and temporal attributes, but also could model both linear and nonlinear transitions. In order to obtain better link prediction performance, Liu et al. also proposed a degree-related clustering coefficient index to quantify the clustering ability of nodes (Liu et al., 2016). Moreover, there are efforts that have been put into employing community information into link prediction. For example, Karimi et al. proposed a community-based link prediction method CLPES in multiplex networks (Karimi et al., 2021). In addition to nodes and links information, community information was also employed for link prediction. Singh et al. also presented a community-based link prediction approach CLP-ID based on an information diffusion algorithm (Singh et al., 2020). They first divided a network into clusters and then used an algorithm based on information diffusion and community structure to predict missing links. In general, the disadvantage of the hybrid methods is their hyper parameters, which are important for improving

Table 1
Categories of existing main approaches for link prediction. In general, existing link prediction methods can be broadly classified into four categories: similarity-based methods, probabilistic methods, dimension reduction-based and hybrid methods. The characteristics of such methods and their recent efforts are briefly described in Section 2.

Method category	Method subcategory	Characteristics	Strengths	Weaknesses	Recent efforts
Similarity-based methods	Local indices	Only neighborhood information is considered	Straightforward; low computational complexity	Low accuracy	CN (Rafiee et al., 2020); AA (Rafiee et al., 2020); RA (Liu et al., 2017); Jaccard Coefficient (Bag et al., 2019); etc.
	Global indices	The entire topological information is considered	Better accuracy in general	Complex; time-consuming	Katz (Vural & Kaya, 2018); LHN (Wahid-Ul-Ashraf et al., 2019); RWR (Zhou et al., 2021); ACT (Wahid-Ul-Ashraf et al., 2019); Cos^+ (Kumar et al., 2020)
	Quasi-local indices	Trade-off between local and global indices	Lower computation complexity than global indices	Dependent on datasets and applications	LP (Aziz et al., 2020); LRW (Song et al., 2019); CCPA (Ahmad et al., 2020); CNDP (Rafiee et al., 2020)
Probabilistic methods	Probabilistic models	Build a probabilistic model based on several parameters	Classical statistical models can be used	Key parameters need to be identified	The probabilistic model (Javari et al., 2017); SBM (Stanley et al., 2019); The Markov model (Das & Das, 2017); NARM (Zhao et al., 2017)
	Maximum likelihood models	Estimate the parameters that maximize the likelihood	Key parameters are estimated from observed data	Complex and time-consuming	The maximum likelihood estimator of network connections (Gaucher & Klopp, 2021); The maximum a posteriori estimation (Kuang & Scoglio, 2021)
Dimension reduction-based methods	Network embedding	Specify a network with low-dimensional vectors	Classical machine learning methods can be used	Accurate network embedding is difficult	PME (Chen et al., 2018)
	Matrix factorization	Factorize the adjacency matrix into two lower dimension matrices	Widely used in recommendation system	Constraints need to be imposed on matrix factorization	DeepEye (Ahmed et al., 2018)
	Matrix completion	Recover the network by the factorized matrices	Widely used in recommendation system	Constraints need to be imposed on matrix factorization	LR (Pech et al., 2017)
Hybrid methods	Machine learning-based methods	Map related features to the connection probability by machine learning algorithms	Classical machine learning algorithms can be used	Features used for machine learning need to identified	LLSLP (Liu & Li, 2021), GTRBM (Li et al., 2018)
	Community-based link prediction	Community information is employed	Better performance	Community needs to detected first	CLPES (Karimi et al., 2021); CLP-ID (Singh et al., 2020)

the performance, but they often need to be optimized in terms of empirical experience.

To sum up, there are many approaches that have been proposed for link prediction and Table 1 summaries the categories of existing link prediction methods as well as their characteristics, strengths, weakness and recent efforts mentioned in this section. Compared with the aforementioned efforts, we propose a novel link prediction approach *ICP* based on inductive matrix completion, which recoveries the node connection probability matrix by applying node features to a low-rank matrix. The approach can be regarded as a kind of dimension reduction-based method, but the difference is that it employs a comprehensive node feature representation by combining different structural topology information with node importance properties. The node connection probability matrix is then generated by a bi-linear function with their features and the low-rank matrix. Moreover, inductive matrix completion is a supervised learning task, and the solution of the underlying low-rank matrix can be based on representative nodes and makes it possible to be extended for link prediction in large-scale networks.

3. Methods

Given an undirected and unweighted network $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, $E = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ is the set of edges. The basic idea of link prediction is to estimate a score $s(i, j)$ to a pair of nodes i and j without edges. The scores of all unconnected pairs are then sorted from large to small, and the top node pair has the highest connection probability. In this section, we first provide the definitions and motivations of existing well-known link prediction methods, which will be used as the baselines to compare with the method proposed in this work. Afterwards, two link prediction methods based on both standard matrix completion and inductive matrix completion are detailed. Note that the explanation of the main symbols and notations involved in this paper can be found in Appendix.

3.1. Traditional baseline methods

In order to demonstrate the superiority of our inductive matrix completion-based approach *ICP*, we introduce eleven classical link prediction methods as the baselines:

3.1.1. CN

CN is the most widely used link prediction method. Moreover, it is simple, efficient and can also be used in medium-sized and large-scale networks. The definition of CN is as follows,

$$s_{ij}^{CN} = |\Gamma(i) \cap \Gamma(j)| \tag{1}$$

where $\Gamma(i)$ denotes the set of neighbor nodes of node i and $|\Gamma(i) \cap \Gamma(j)|$ denotes the number of common neighbors of nodes i and j. The core idea of CN is that the more common neighbors between two nodes, the more likely the two nodes are to connect.

3.1.2. RA

The idea of RA is that each node in a network has a unit of resources, and each node can allocate resources equally to its neighbors. In other words, a node with degree k allocates $\frac{1}{k}$ of resources to each neighbor node. In RA, the connection probability is calculated as follows,

$$s_{ij}^{RA} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{k_z} \tag{2}$$

where $z \in \Gamma(i) \cap \Gamma(j)$ is a node which belongs to the set of common neighbors of nodes i and j. k_z denotes the degree of node z. We can see that, the connection probability between two nodes depends on the total resources of their common neighbors.

3.1.3. AA

Based on CN, AA considers that the nodes with smaller degree have more contribution than the nodes with higher degree when it calculates the connection probability between two nodes.

$$s_{ij}^{AA} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log k_z} \tag{3}$$

where k_z denotes the degree of a common neighbor z of nodes i and j. $\frac{1}{\log k_z}$ ensures that a node with higher degree has less contribution to s_{ij}^{AA}.

3.1.4. LP

LP goes beyond CN and also considers the contribution of the third-order path.

$$s_{ij}^{LP} = (A^2)_{ij} + \beta \cdot (A^3)_{ij} \tag{4}$$

where A is the adjacency matrix of a network. A^2 and A^3 are $n \times n$ matrices, which denote the second-order path and the third-order path of the network, respectively. β is an adjustable parameter and it is usually set to a small value less than one to make shorter paths have more weights.

3.1.5. L3

L3 is a variation of CN and it only considers the contribution of the third-order path, and the connection probability between two nodes is calculated as follows,

$$s_{ij}^{L3} = (A^3)_{ij} \tag{5}$$

We can see that, if the number of paths with the length of three between two nodes is more, the two nodes are more likely to connect.

3.1.6. CCPA

CCPA proposed by Ahmad et al. estimates the connection probability between two nodes based on the number of common neighbors and their closeness centrality (Ahmad et al., 2020).

$$s_{ij}^{CCPA} = \beta \cdot |\Gamma(i) \cap \Gamma(j)| + (1 - \beta) \cdot \frac{n}{d(i,j)} \tag{6}$$

where n is the number of nodes in the network, and $d(i,j)$ is the distance between nodes i and j. $\frac{n}{d(i,j)}$ denotes closeness centrality between nodes i and j. Moreover, $\beta \in [0,1]$ is a user defined value that controls the weight or importance of common neighbor and centrality.

3.1.7. CNDP

CNDP estimates the connection probability between two nodes i and j in terms of the topological characteristics including common neighbors of each two nodes and the network average clustering coefficient (Rafiee et al., 2020).

$$s_{ij}^{CNDP} = |C_z| \cdot (|\Gamma(z)|)^{-\beta C} \tag{7}$$

where z is a common neighbor of nodes i and j, and $|C_z|$ is the number of neighbors of z which consists of the common neighbors of i and j in addition to i and j. $|\Gamma(z)|$ is the number of neighbors of z, C is the network average clustering coefficient, and β is a constant value which needs to be optimized. We can see that, in addition to the common neighbors of two nodes, CNDP also considers the neighbors of common neighbors (Rafiee et al., 2020).

3.1.8. Katz

Katz not only considers the second-order and third-order paths between nodes, in fact, it considers all paths between two nodes and assigns higher weight to shorter paths.

$$s_{ij}^{Katz} = \sum_{z=1}^{\infty} \beta^z \cdot (A^z)_{ij} = ((I - \beta \cdot A)^{-1} - I)_{ij} \tag{8}$$

where I is an identity matrix. β is the weight attenuation factor which controls the weight of the paths with different lengths. β is generally between 0 and 1, and a small value of β will result in penalizing longer paths. In general, Katz provides a good trade-off among different lengths of paths between two nodes, thereby demonstrating good prediction performance in various types of networks.

3.1.9. ACT

ACT is a kind of link prediction method based on random walk, which is one of the most effective and simplest tools for traversing a network using only local information and investigating the properties of a network (Masuda et al., 2017; Nasiri et al., 2021). Let $t(i,j)$ be the average number of steps a random particle needs to take from node i to node j, then the average commuting time of node i and node j $act(i,j)$ is defined as: $act(i,j) = t(i,j) + t(j,i)$. $act(i,j)$ can be further calculated by solving the pseudo inverse L of the Laplacian matrix of the network, i.e.,

$$act(i,j) = c \cdot (L_{ii} + L_{jj} - 2L_{ij}) \tag{9}$$

where L is the pseudo inverse of the Laplacian matrix of the network, and c is a constant. Generally, the smaller $act(i,j)$, the closer of two nodes i and j are. Therefore, the ACT-based similarity between two nodes i and j is further defined as,

$$s_{ij}^{ACT} = \frac{1}{L_{ii} + L_{jj} - 2L_{ij}} \tag{10}$$

3.1.10. Cos^+

Cos^+ is also a kind of link prediction method based on random walk and it defines the similarity between nodes i and j as,

$$s_{ij}^{Cos+} = \frac{P_{ij}}{P_{ii} \cdot P_{jj}} \tag{11}$$

where $P_{ij} = v_i \cdot v_j$, and $v_i = \Lambda^{1/2} J^T e_i$. Here, J is a standard orthogonal matrix, which is obtained by arranging the eigenvectors of L from large to small according to their corresponding eigenvalues. The definition of L is the same with the definition in ACT. Λ is a diagonal matrix with the eigenvalues as its diagonal elements. e_i represents a one dimension vector in which only the ith element is 1 and other elements are 0.

3.1.11. RWR

RWR can be regarded as an application of the extended PageRank algorithm. For a node i, it will iteratively move to a neighbor randomly with probability α and it will return to the node i with probability $1-\alpha$. Let q_{ij} denote the probability that a random particle walks from node i to node j, then the connection probability between nodes i and j is defined as follows.

$$s_{ij}^{RWR} = q_{ij} + q_{ji}. \tag{12}$$

The q_{ij} is the probability that a particle at node i moves to node j, which can be accessed by

$$q_i = (1-\alpha) \cdot (I - \alpha P^T)^{-1} \cdot e_i \tag{13}$$

where q_i is a probability vector, which contains the probabilities that a particle at node i moves to other nodes. P is the transition probability matrix of a network and the definition of e_i is the same with the definition in Cos^+.

3.2. Link prediction-based on standard matrix completion

The goal of link prediction is to predict potential links for a given pair of two nodes. We form a node association matrix $M \in R^{n\times n}$, $M_{ij} = 1 (i \neq j)$ if there is a link between nodes i and j, otherwise $M_{ij} = 0$. Although it is easy to obtain the adjacency matrix $A \in R^{n\times n}$ of a network, there may be missing or unobservable links and predicting such links are also known as matrix completion. In other words, $M = A +$ missing links. Matrix completion is one of the most successful and well-studied techniques for recommendation systems (Chen et al., 2021; Mongia & Majumdar, 2021; Yang & Xu, 2021). Given the observed links of a network from its adjacency matrix A, the goal of link prediction is to identify missing links based on the structure of the adjacency matrix A. Since M is typically very sparse, the rank of M is low and it can be derived from a linear combination of two smaller matrices, i.e., $M = WH^T$, where $W \in R^{n\times r}$ and $H \in R^{n\times r}$ ($r \ll n$). Moreover, factoring M into W and H can be accomplished by solving the following optimization problem,

$$\arg\min_{W,H} \|A - WH^T\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2) \tag{14}$$

where λ is a regularization parameter. Note that the low rank constraint on W and H is simplified to minimize the Frobenius norms $\frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)$, since the optimization problem with the low-rank constraint is NP-hard. Once Eq. (14) is converged, we use the following equation to measure the connection probability between two nodes i and j,

$$s(i,j) = (WH^T)_{ij} \tag{15}$$

However, the limitation of standard matrix completion is that it only exploits the adjacency matrix of the network and does not take advantage of other additional information, such as local, quasi-local and global structural topology information. Intuitively, the richer the information is used, the better performance a link prediction method has.

3.3. Link prediction-based on inductive matrix completion

Due to the limitation of standard matrix completion, we propose a novel link prediction method *ICP*, which employs inductive matrix completion for link prediction (Ding et al., 2019; Si et al., 2016). Different from traditional standard matrix completion, inductive matrix completion also exploits low-dimensional vector representations of node features to predict the connection probability between two nodes. Formally, let $F = [F_1, F_2, \ldots, F_n]^T$ to be the matrix of node features, and each row F_i represents a feature vector of node i. Inductive matrix completion regards the link prediction problem as that of solving a low-rank projection matrix Z which maps the low-dimensional feature space from one node to the other. Moreover, the difference between A and $F_i Z F_j^T (1 \le i, j \le n)$ should be small. *ICP* is then formulated as follows,

$$A_{ij} \approx F_i Z F_j^T \tag{16}$$

Note that Z is of dimension $n \times n$. Since solving low-rank matrix Z is a NP-hard problem, instead of optimizing low-rank matrix Z directly, we consider low-rank decomposition, i.e., $Z = WH^T$ ($W \in R^{n\times r}$, $H \in R^{n\times r}$, $r \ll n$) and optimize W and H jointly. The process of inductive matrix completion-based link prediction is shown in Fig. 1. Similarly, we simplify the minimization of $Z = WH^T$ by minimizing their Frobenius norms, i.e., $\frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)$. Finally, we have the following optimization problem,

$$\arg\min_{W,H} \sum_{1\le i,j\le n(i\neq j)} (A_{ij} - F_i W H^T F_j^T)^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2) \tag{17}$$

where λ is a regularization term added to avoid overfitting. In order to solve W and H, we regard the problem as a supervised learning task and using existing and non-existing links to train the *ICP* model. Since real-world networks are sparse in many cases, i.e., there are unbalanced number of existing and non-existing links. To avoid the model that be in favor of a class with more samples, in this paper, we used all known existing links and generated an equal-size set of non-existing links. In other words, the desired number of non-existing links were randomly selected.

Once W and H are solved, we then use the following formula to measure the connection probability between two nodes i and j,

$$s(i,j) = F_i W H^T F_j^T \tag{18}$$

3.3.1. Solving the optimization problem

In order to find optimal W and H, we need to minimize the loss function of the *ICP* model, i.e., Eq. (17) and update the parameters of the model, i.e., W, H at the same time iteratively. The updating rules based on standard gradient descent approach are presented in what follows.

1. Updating W with H fixed

First of all, we fix the value of H and solve the following optimization problem only related to W,

$$\arg\min_{W} O(W) = \sum_{1\le i,j\le n(i\neq j)} (A_{ij} - F_i W H^T F_j^T)^2 + \frac{\lambda}{2}\|W\|_F^2 \tag{19}$$

By using the property that $\|X\|_F^2 = tr(XX^T)$ (tr is the trace of a square matrix, i.e., the sum of the diagonal elements of the matrix), we have the following equivalent object function,

$$\arg\min_{W} O(W) = \sum_{1\le i,j\le n(i\neq j)} (A_{ij} - F_i W H^T F_j^T)^2 + \frac{\lambda}{2} tr(WW^T) \tag{20}$$

Furthermore, we can get the partial derivative of $O(W)$ with respect to W,

$$\frac{\partial O(W)}{\partial W} = \sum_{1\le i,j\le n(i\neq j)} F_i^T (F_i W H^T F_j^T - A_{ij}) F_j H + \lambda W \tag{21}$$

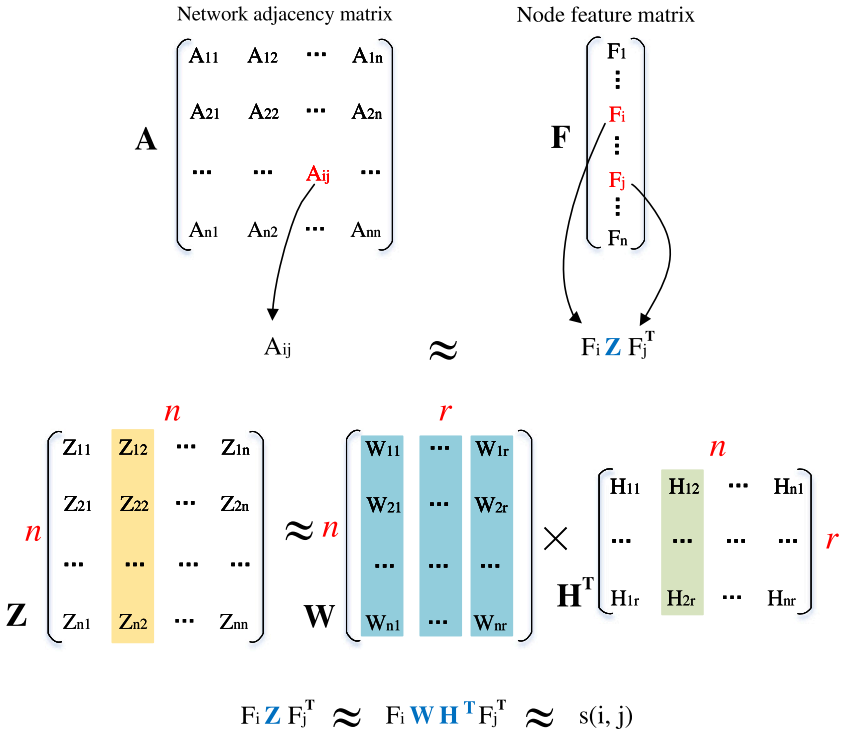

Fig. 1. The process of inductive matrix completion-based link prediction. Inductive matrix completion regards the link prediction problem as that of solving a low-rank projection matrix Z which maps the low-dimensional feature space from one node to the other. Instead of optimizing low-rank matrix Z directly, we factorize Z into $W \in R^{n\times r}$ and $H \in R^{n\times r}$ ($r \ll n$), which are two lower dimensional matrices. Once W and H are solved, their connection probability is the product of W, H^T and their features.

Based on standard gradient descent method, we can get the following updating rule,

$$W = W - \lambda \frac{\partial O(W)}{\partial W} \tag{22}$$

2. Updating H with W fixed

Next, we fix the value of W and solve the following optimization problem only related to H,

$$\arg\min_{H} O(H) = \sum_{1\le i,j\le n(i\neq j)} (A_{ij} - F_i W H^T F_j^T)^2 + \frac{\lambda}{2}\|H\|_F^2 \tag{23}$$

Similarly, we can get the partial derivative of $O(H)$ with respect to H,

$$\frac{\partial O(H)}{\partial H} = \sum_{1\le i,j\le n(i\neq j)} F_j^T (F_j W H^T F_i^T - A_{ij}) F_i W + \lambda H \tag{24}$$

Based on standard gradient descent method, we can get the following updating rule,

$$H = H - \lambda \frac{\partial O(H)}{\partial H} \tag{25}$$

Algorithm 1 Training the *ICP* model

Require: : A, F, *epochs*
Ensure: : W, H
1: **function** *ICP*(A, F)
2: initialize W and H randomly
3: **for** epoch from 1 to *epochs* **do**
4: update W via Eqs. (21) and (22)
5: update H via Eqs. (24) and (25)
6: **end for**
7: **return** W, H
8: **end function**

Algorithm 1 presents the general process of training the *ICP* model in pseudo code, where A denotes the adjacency matrix of a network, F is the matrix of node features, and *epochs* denotes a fixed number of training epochs. At the beginning of the training process, W and H are initialized randomly. During the training process of the model, we updated W and H in terms of the above rules until the objective function is converged. The model training can be run for a fixed number of epochs. Generally, too many training epochs can lead to over-fitting problem, whereas too few may result in an underfit model. Additionally, the training process can be stopped if the difference between current loss and the previous loss is below a threshold, which is quantified as no improvement. In this paper, we utilized the former method and stopped the training process after a fixed number of epochs based on the convergence analysis in Section 5.1.

3.4. Feature construction

Compared with other efforts, *ICP* explores a comprehensive node feature representation by combining different structural topology information with node importance properties. More precisely, in order to specify the structural topology information of a node, we employ its degree, the first-order, the second-order and the third-order paths. The first-order path of a node is a binary vector used to describe whether other nodes are adjacent to it or not. The second-order and third-order paths denote the number of paths with length two and three between a node and other nodes, respectively. Note that we do not consider global structural topology information of a node in this work since it is time-consuming to decide the optimal weight attenuation factor which controls the weight of the paths with different lengths. It is generally enough to obtain the reasonable performance with the information of both the second-order and the third-order paths. More details about the performance comparison between the *ICP* models with and without global structural topology information will be found in Section 6.

Moreover, *ICP* considers the node properties related its importance, such as clustering coefficient, average shortest path length, degree centrality, closeness centrality, betweenness centrality and average

Table 2
The employed node features. There were $7+3n$ features that were used to describe a node comprehensively. Besides the node degree, the node properties related its importance, i.e., clustering coefficient, average shortest path length, degree centrality, closeness centrality, betweenness centrality and average neighbor degree were also included. Moreover, we also considered adjacency vector, the second and third-order neighbors of a node.

Notation	Name	Definition	Formula
f_1	Node degree	The number of edges connected to node i.	$k_i = \sum_j a_{ij}$
f_2	Clustering coefficient	A measure of local density of connections of node i.	$\frac{2\Delta_i}{k_i(k_i-1)}$
f_3	Average shortest path length	The average shortest path length of node i.	$\frac{1}{n-1}\sum_j d(i,j)$
f_4	Degree centrality	The fraction of nodes connected to node i.	$\frac{k_i}{n-1}$
f_5	Closeness centrality	A measure of centrality of node i in a network.	$\frac{n-1}{\sum_j d(i,j)}$
f_6	Betweenness centrality	A measure of centrality in a graph based on shortest paths.	$\sum_{i\neq j\neq k}\frac{\sigma_{j,k}(i)}{\sigma_{j,k}}$
f_7	Average neighbor degree	The average degree of the neighborhood of node i.	$\frac{1}{\lvert\Gamma(i)\rvert}\sum_{j\in\Gamma(i)} k_j$
$f_{8..7+n}$	Adjacency vector	A binary vector used to describe whether other nodes is adjacent to node i.	$(a_{i1}, a_{i2}, \ldots, a_{in})$
$f_{8+n..7+2n}$	The second-order neighbors	The number of paths with length 2 between node i and other nodes	$(u_{i1}, u_{i2}, \ldots, u_{in})$
$f_{8+2n..7+3n}$	The third-order neighbors	The number of paths with length 3 between node i and other nodes	$(t_{i1}, t_{i2}, \ldots, t_{in})$

* In the table, Δ_i denotes the number of triangles through node i. $d(i,j)$ denotes the distance between nodes i and j. $\sigma_{j,k}$ is the total number of shortest paths existing between nodes j and k. $\sigma_{j,k}(i)$ is the number of shortest paths that passes through node i. u_{ij} and t_{ij} are the elements of A^2 and A^3, and they denote the second-order path and the third-order path of the network, respectively.

Table 3
The time complexity of different link prediction methods. *ICP* involves feature construction, preprocessing, selection and inductive matrix completion, and its time complexity is $\mathcal{O}(n^3)$.

Method	CN	RA	AA	LP	L3	CCPA	CNDP	Katz	ACT	Cos^+	RWR	ICP
Time complexity	$\mathcal{O}(nd^3)$	$\mathcal{O}(nd^3)$	$\mathcal{O}(nd^3)$	$\mathcal{O}(ln^2d)$	$\mathcal{O}(n^3)$	$\mathcal{O}(nd^3)$	$\mathcal{O}(nd^2)$	$\mathcal{O}(nd^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(\alpha n^2 d)$	$\mathcal{O}(n^3)$

neighbor degree. The clustering coefficient is a measure of local density of connection of a node. The average shortest path length is the average shortest path length of a node. The degree centrality is defined as the fraction of nodes connected to a node. The closeness centrality is a measure of the average shortest distance from a node to others and it is the inverse of the average shortest path length of the node. The betweenness centrality is also a measure of centrality based on shortest paths. The average neighbor degree denotes the average degree of the neighborhood nodes of a node. In total, there are $7+3n$ features that are used to describe a node, thereby constructing a feature space $F \in R^{n\times(7+3n)}$. Here, $F = [F_1, F_2, \ldots, F_n]^T$ and each row $F_i = [f_1, f_2, \ldots, f_{7+3n}]$ represents the corresponding feature vector of node i. The node features employed in this paper are listed in Table 2.

Moreover, we will use a matrix $B \in R^{n\times 7}$ to denote the node feature space consisting of f_1, f_2, ...and f_7 in what follows for clarity. The node feature spaces consisting of $f_{8..7+n}$, $f_{8+n..7+2n}$ and $f_{8+2n..7+3n}$ will be denoted by $A \in R^{n\times n}$, $U \in R^{n\times n}$ and $T \in R^{n\times n}$, respectively.

3.5. Feature dimension reduction

This work employs $7+3n$ features to specify a node, where n is the number of nodes in a network. If the network is in large scale, it is time-consuming to build link prediction models based on such large node feature space. Moreover, the matrices A, U and T are sparse and contain a large number of zeros. Additionally, the values in U and T vary widely since they denote the number of paths with length 2 or 3 between a node and other nodes. Therefore, we employed PCA to deal with the features $f_{8..7+n}$, $f_{8+n..7+2n}$ and $f_{8+2n..7+3n}$ before building the *ICP* models. Compared with other dimension reduction methods, PCA is very simple and can select the most important features as needed. Moreover, the PCA results have nothing to do with users, since PCA does not need any empirical experience to set parameters. The process of PCA-based feature dimension reduction is shown in Fig. 2.

First, we normalized each column of the matrices of A, U and T by Min–Max normalization to reduce the fluctuation range of data. Moreover, we used the square roots of the normalized results to make the data values much smaller. We then transformed the path information into the similarity information by calculating the Euclidean distance between different nodes and got the matrices A', U' and T'. In other words, the similarity between two nodes was calculated by calculating the Euclidean distance between two node vectors. The smaller the distance is, the higher the similarity between two nodes. Afterwards, we combined A', U' with T' and fed the combined matrix to PCA to obtain the main node feature information. With the dimension reduction, we used k eigenvalues to specify a node, thereby generating a node feature subspace $Y \in R^{n\times k}$. Finally, after combining Y with the feature space B, we got the new feature subspace $F' \in R^{n\times(7+k)}$.

During the process of dimension reduction, the feature space B describing features F_1, F_2, ...and F_7 were kept same and finally we got a new $n\times(7+k)(k \ll 3n)$ feature subspace F'.

3.6. The time complexity

In *ICP*, we first constructed and integrated different structural topology information with node importance properties as the node features. The time complexity of this step is $\mathcal{O}(n^3)$ (n is the number of nodes in a network), since the feature construction involves complex calculations, such as average shortest path length and closeness centrality. Afterwards, we normalized such information via Min–Max normalization and calculated the Euclidean similarity of different nodes, and then used PCA to identify the main feature components. The time complexity of this step is $\mathcal{O}(nk+n^2+n^3)$, and k is the reduced dimension of the node feature. Finally, we used inductive matrix completion to learn the low-rank projection matrix Z which maps the feature space from one node to the other. The time complexity of this step is $\mathcal{O}((nk+m)r^2)$, where m is the number of edges in a network, and r is the dimension of W and H of *ICP*. Therefore, the total time complexity of our method is $\mathcal{O}(n^3)$. In other words, *ICP* is complex due to complex feature construction and its time complexity is higher than or at least equal with the baseline methods, whose time complexity is shown in Table 3.

4. Materials

4.1. The network datasets

We evaluated the performance of different link prediction models involved in this paper in twelve real networks. The details of such datasets can be found in Table 4.

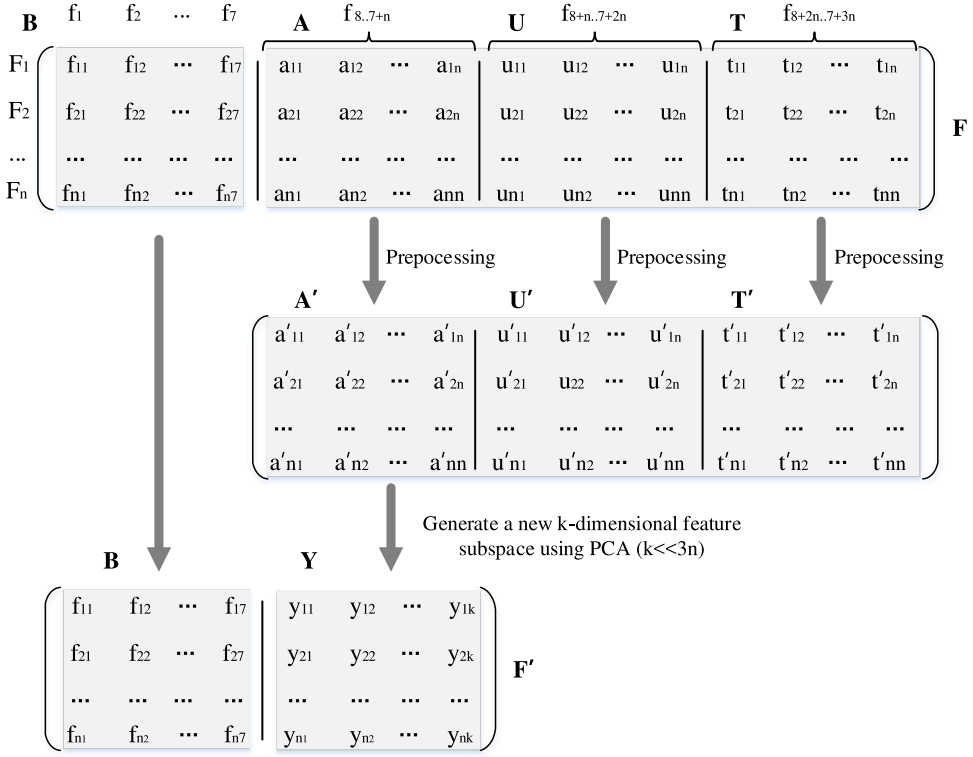

Fig. 2. The process of feature construction and dimension reduction. The feature spaces A, U and T were first processed by Min–Max normalization and the path information was transformed into the similarity information by calculating the Euclidean distance between different nodes. The processed feature spaces were then reduced by PCA. The final feature space was the combination of the feature space B with the reduced feature space Y.

Table 4
The details of real networks employed in the experiments. To demonstrate the superiority of *ICP*, we employed six simple networks, i.e., Leadership, Revolution, Karate, Dolphins, Jazz and USAir. Moreover, we selected six complex networks with medium sizes, i.e., NS, PPI, KHN, ADV, ZWL and HTC. In the table, $|V|$ and $|E|$ denote the number of nodes and edges in the networks, respectively. $\langle L\rangle$, $\langle K\rangle$ and $\langle C\rangle$ are the average shortest path, the average degree and the cluster coefficient of the networks, respectively.

Dataset	$\|V\|$	$\|E\|$	$\langle L\rangle$	$\langle K\rangle$	$\langle C\rangle$
Leadership	64	99	2.87	4.95	0.00
Revolution	277	160	3.11	2.27	0.00
Karate	34	78	2.41	4.59	0.57
Dolphins	62	159	3.36	5.13	0.26
Jazz	198	2742	2.24	27.70	0.62
USAir	332	2126	2.74	12.81	0.63
NS	1598	2742	5.82	3.45	0.88
PPI	2374	11693	5.09	9.85	0.31
KHN	3772	12718	3.63	6.74	0.25
ADV	5155	39285	3.22	15.24	0.25
ZWL	6651	54182	3.85	16.29	0.32
HTC	7610	15751	5.68	4.14	0.49

Leadership is a bipartite network of company leaders, in which the left part represents persons and the right part represents companies. An edge denotes that a person has a leading position in a company. Revolution is a network of American revolutionary organizations. It is also a bipartite network, in which the left part represents persons, the right part represents organizations, and an edge indicates that a person participates in an organization. Karate is a friendship network of karate clubs in an American University in the 1970s. Dolphins is a social network of dolphins. Jazz is a network of collaboration between jazz musicians. An edge in the network means that two musicians play together in a band. USAir is an airline network of the United States. In the network, a node represents an airport and an edge indicates that there is a route connecting two airports. Network science (NS) is a collaborative network of scientists and there is a link between two scientists if they publish papers together. PPI is a protein–protein interaction network. An edge of the network represents the metabolic interaction between two proteins. Such networks can be found in the KONECT project (KONECT, 2021), which is a project in the area of network science with the goal to collect and analyze network datasets. Moreover, we also selected four medium-sized networks, i.e., KHN, ZWL, HTC and ADV. KHN, ZWL and HTC which are co-authorship networks for different fields of study and ADV is a social network. Such four networks are collected by NOESIS, which is an open source framework for network data mining (Martínez et al., 2019).

4.2. The evaluation metrics

In this paper, two metrics were employed to evaluated the performance of the involved models, i.e., AUC and precision. AUC measures the accuracy of a link prediction method as a whole, while precision only considers whether the top links are predicted accurately.

4.2.1. AUC

More precisely, AUC can be interpreted as the probability that the score of an edge in the test set is higher than that of a randomly selected non-existing edge. AUC can be calculated as follows.

$$AUC = \frac{n' + 0.5 * n''}{n} \tag{26}$$

where n is the times of independent comparisons, n' denotes the times of the edges in the test set having greater scores, and n'' denotes the number of situations that two scores are same. If all scores are randomly generated, $AUC = 0.5$. The AUC score of an algorithm greater than 0.5 indicates that its accuracy is better than a random method.

4.2.2. Precision

Precision is a simple evaluation metric of link prediction and it is defined as the ratio of the actual number of connected edges to the predicted number of connected edges. In other words, we calculate the connection probability between two nodes without connection in a network first, and then sort the probability scores from large to small. If m of the top L edges are in the test set, then the prediction precision is calculated as follows:

$$Precision = \frac{m}{L} \tag{27}$$

The higher the precision score, the more accurate the prediction is. However, if the AUC scores of two algorithms are the same, the one with a higher precision score is better, since it only considers whether the top L edges with the highest scores are predicted accurately. In other words, we will employ AUC as a primary evaluation metric since it measures the performance of a link prediction method as a whole, and we will use precision as a secondary evaluation metric to compare two link prediction methods if they have similar AUC results.

5. Performance analysis

In this section, several experiments are conducted to comprehensively evaluate the *ICP* performance. Section 5.1 discusses the *ICP* convergence in different datasets. Section 5.2 presents the performance of PCA dimension reduction in reconstructing and selecting node features. Section 5.3 discusses the recommended setting of *ICP*'s regularization term λ. Section 5.4 presents the results of the performance evaluation between *ICP* and the baseline methods in twelve real networks.

5.1. The ICP model convergence verification

In the training process of *ICP*, the model converges if there is usually no significant error decrease or the model performance does not increase any more, and it is important for a model to converge after a fixed number of training epochs. The update rules of *ICP*, i.e., Eqs. (22) and (25) can ensure the convergence of the training process, since the updating rules are all based on standard gradient descent processes. Fig. 3 presents the convergence result graphs in different datasets. The X-axis shows the number of training iterations and the Y-axis shows the objective value of Eq. (17) after each iteration. We can see that the values of Eq. (17) at the first several iterations vary greatly due to different sizes of networks. However, all of them are converged after 500 iterations, which proves that *ICP* has excellent convergence performance. In our experiments, the training iterations were set to 1000 to ensure that the *ICP* performance does not increase any more.

5.2. The PCA dimension reduction analysis

In this work, our inductive matrix completion method exploits $7 + 3n$ features, including the local, quasi-local structural topology information and the importance properties of a node to specify its feature space. The constructed features are sparse and even bloated if a network in large scale. To deal with the problem, we employed PCA to transform high-dimensional original feature space into low-dimensional space, where the remaining features can still explain the maximum variance within the original space. Smaller feature spaces will lead to less computation time and ultimately speed up building link prediction models. Moreover, the prediction performance may be improved in lower dimensions by removing redundant features. In order to analyze the performance of dimension reduction, we used both the original high-dimensional feature space and the reduced low-dimensional feature space to build *ICP* models in different datasets, respectively. We analyzed the performance of PCA dimension reduction in terms of AUC, precision and time used in building the models. More precisely, the time is the training period of *ICP* models with a fixed regularization parameter λ=1.0 after 1000 training epochs in different networks. Note that the regularization parameter, the number of training epochs as well as other parameters were kept the same except the dimension of the node features in the experiments. The results are shown in Table 5.

We can see that the PCA-based dimension reduction can effectively reduce the running time in building *ICP* models. The time reduction is not obvious in the first three datasets, i.e., the time is saved by 11.5%, 4.9% and 4.5%, as shown in Fig. 4. This is because the size of these datasets is small. However, with the size of the dataset increases in other datasets, the PAC technology can greatly shorten the time of training *ICP*. For example, the time is saved by 70.5%, 71.0%, 75.8%, 82.2% and 87.7% in Jazz, PPI, ADV, ZWL and HTC, respectively. Moreover, the dimension reduction helps in improving the prediction accuracy of the algorithm. Although the AUC improvements are not obvious in all datasets except in Revolution, such as 0.1% in Leadership, 0.2% in ZWL, 0.5% in Karate, Jazz, PPI and KHN, 0.6% in USAir and HTC, 0.8% in NS and ADV, 1.2% in Dolphins, the prediction performance of all datasets does not decline at least.

5.3. The regularization parameter analysis

λ is the hyper parameter of our model, different settings may lead to different results. In our *ICP* model, we used it to avoid overfitting of the objection function in Eq. (17). In order to analyze the effect of λ, we built *ICP* models in different networks and varied the value of λ from 0 to 2.0 with the step size of 0.1. Moreover, we adopted *AUC* as the metric to evaluate the results with respect to different parameter settings, the results are shown in Fig. 5.

We can see that, the final results of our model in different networks are sensitive to λ, which can highlight the significance of the regularization parameter in our model. For example, our *ICP* models have the best results when λ is around 0.7 in Leadership, Revolution, USAir and ZWL, and they have the best results when λ is around 0.9 in Dolphins and Jazz. However, the best AUC values are got when λ is zero in NS, ADV and HTC. In other words, λ is not necessary in NS, ADV and HTC. The results also indicate the fact that it is hard to find a fixed value of λ that can ensure our *ICP* models have the best result in all datasets. We suggest to use the best λ values before optimizing other parameters, such as the number of training epochs. The best λ values used in different *ICP* models are listed in Table 6.

5.4. Comparison with the baseline methods

To illustrate the performance of *ICP*, we used CN, RA, AA, LP, L3, CCPA, CNDP, Katz, ACT, Cos^+ and RWR as baseline methods to build link prediction models in twelve different real networks. Afterwards, we compared the performance of such models with *ICP* in terms of both AUC and precision. Tables 7 and 8 present the AUC and precision results of different models, respectively. The last row of Table 7 lists the average AUC results of different methods in all datasets. Note that the best results are indicated in bold and the second best results are indicated with asterisks. In order to have a reasonable comparison, besides the optimal λ values listed in Table 6, the *ICP* models also used optimal values of k and r, which are also provided in Table 7. Moreover, LP, Katz, CCPA and CNDP were optimized with different β values, respectively. As to RWR, the probability α was set to 0.85.

We can see that, although the performance of the baseline methods varies in different datasets, the quasi-local index-based similarity methods, i.e., LP, L3, CCPA and CNDP have better performance than the local index-based similarity methods, i.e., CN, RA and AA on the whole since they have three second best AUC results, which are more than the local index-based similarity methods that have only two second best AUC results. All quasi-local index-based similarity methods even have better AUC results than all local index-based similarity methods in KHN and ZWL. Moreover, the average AUC results of LP, L3 and Katz in all datasets are 0.8804, 0.8476 and 0.8888, respectively which are higher

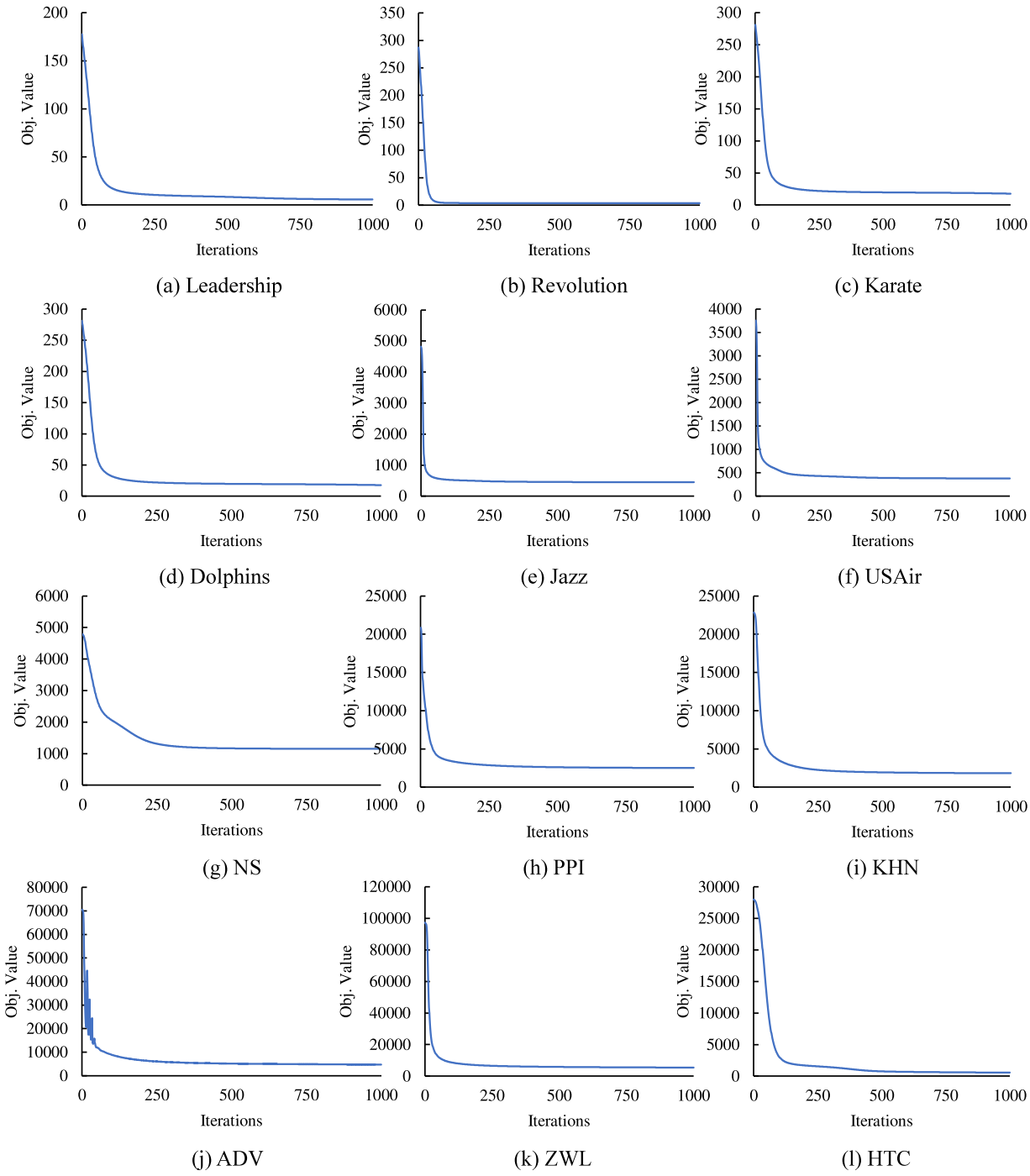

Fig. 3. The *ICP* model converge verification. *ICP* has excellent convergence performance, since its update rules are based on standard gradient descent processes. The X-axis shows the number of training iterations and the Y-axis shows the objective value of Eq. (17) after each iteration.

than any local index-based similarity method here. This is because that the local index-based similarity methods only consider the limited neighborhood information, while the quasi-local index-based similarity methods go beyond them and consider global structural information additionally. The global index-based similarity methods, i.e., Katz, ACT, Cos^+ and RWR have better performance than the quasi-local index-based similarity methods and they have six second best AUC results, and Cos^+ even has the highest AUC result in PPI. The global index-based similarity methods focus on entire global structural information, but they often suffer from higher computational complexity for larger networks. However, not all global index-based similarity methods have similar performance, and the average AUC results of ACT and Cos^+ are 0.8182 and 0.8320, which are much lower than Katz and RWR with the average AUC results 0.8888 and 0.8709, respectively.

Compared with the baseline methods, *ICP* has the best average AUC result 0.9227, which is much higher than the baseline methods. More precisely, the *ICP* models have significant better AUC performance in all datasets except in PPI. For example, the *ICP* model demonstrates its superiority over the baseline methods in Leadership with an AUC results 0.8861, which is higher than the second best AUC result 0.8365 by Katz; the AUC result of the *ICP* model in Revolution is 0.9649 which is much higher than the baseline methods except ACT with the closest performance 0.9523; the AUC result of the *ICP* model in Dolphins is 0.8227 which is higher than the second best result 0.8089 by LP.

Table 5
The dimension reduction analysis. In order to analyze the performance of dimension reduction, the parameters used to build *ICP* models, such as the regularization parameter, the number of training epochs as well as other parameters were kept the same except the dimension of the node features.

Networks	Before dimension reduction			After dimension reduction		
	AUC	Precision	Time	AUC	Precision	Time
Leadership	0.8552	0.0747	142.2 ms	0.8564	0.0747	125.8 ms
Revolution	0.6915	0.0725	2450.1 ms	0.9649	0.1025	2329.6 ms
Karate	0.8449	0.1038	84.3 ms	0.8494	0.1038	80.5 ms
Dolphins	0.8117	0.0485	331.4 ms	0.8218	0.0672	215.1 ms
Jazz	0.9671	0.1754	18.2 s	0.9723	0.1787	5.4 s
USAir	0.9518	0.1392	21.1 s	0.9571	0.1486	9.3 s
NS	0.9339	0.0981	868 s	0.9416	0.1009	415.7 s
PPI	0.9386	0.1141	13915.9 s	0.9431	0.1185	4031.6 s
KHN	0.8687	0.0238	93324.5 s	0.8727	0.0254	36163.4 s
ADV	0.9230	0.0244	107374.7 s	0.9300	0.0320	26013.4 s
ZWL	0.9715	0.0369	551089.4 s	0.9738	0.0437	98146.5 s
HTC	0.8760	0.0275	944449.4 s	0.8810	0.0276	116343.6 s

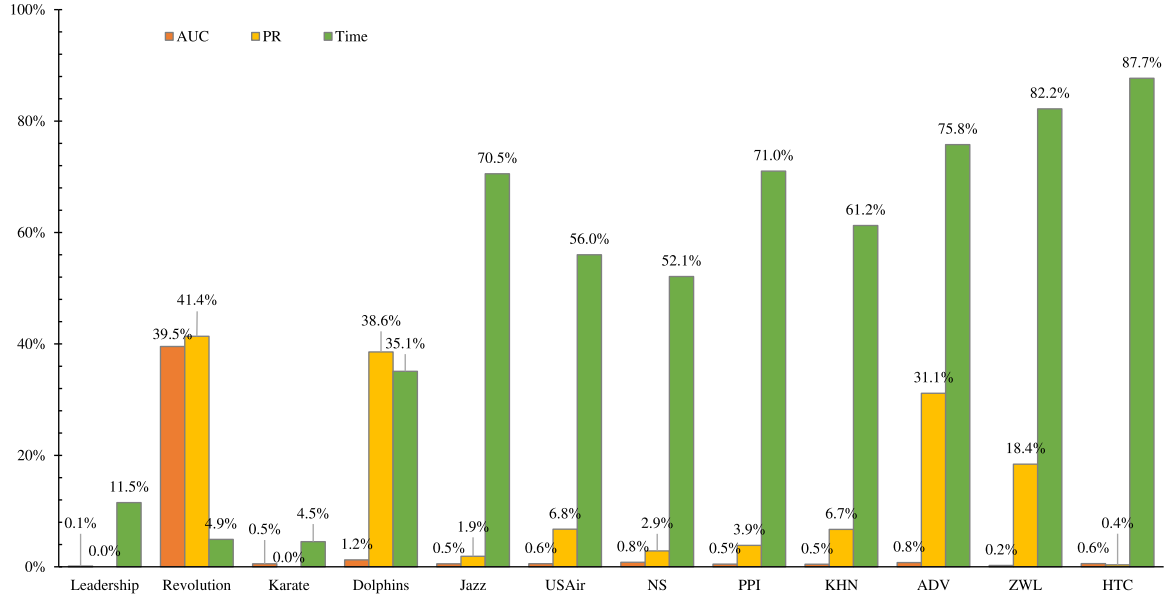

Fig. 4. The performance analysis of PCA dimension reduction. The PCA-based dimension reduction can effectively reduce the running time in building *ICP* models. Moreover, it also helps in improving the link prediction accuracy.

Table 6
The best λ values used in different networks. The results show that the optimal λ values vary in different networks. It helps in improving the model performance in Leadership, Revolution, Karate, Dolphins, Jazz, USAir, PPI, KHN and ZWL, but it is not necessary in the datasets of NS, ADV and HTC.

Dataset	Leadership	Revolution	Karate	Dolphins	Jazz	USAir	NS	PPI	KHN	ADV	ZWL	HTC
λ	0.65	0.7	0.3	0.9	0.9	0.6	0.0	0.3	0.3	0.0	0.65	0.0

Table 7
The AUC results of different methods. The last row of the table lists the average AUC results of different methods in all datasets. Note that the best results are indicated in bold and the second best results are indicated with asterisks. It is shown that the *ICP* models have better AUC performance than the baseline methods in all datasets except in PPI. Although *ICP* does not perform better than Cos^+, which has the highest AUC result in PPI, it still has close performance with Cos^+. Moreover, *ICP* has better precision result than Cos^+ in PPI, which means that *ICP* tends to assign real edges with high connection probabilities.

Method	CN	RA	AA	LP	L3	CCPA	CNDP	Katz	ACT	Cos^+	RWR	ICP (k, r)
Leadership	0.6437	0.6437	0.6437	0.8349	0.8264	0.5126	0.6437	0.8365*	0.5663	0.5875	0.6507	**0.8661** (42, 20)
Revolution	0.6444	0.6444	0.6444	0.7167	0.6093	0.7619	0.6444	0.9021	0.9523*	0.5333	0.6894	**0.9649** (40, 19)
Karate	0.7003	0.7494	0.7404	0.7632	0.7823	0.7118	0.7408	0.7572	0.7118	0.7408	0.8747*	**0.8762** (27, 16)
Dolphins	0.7883	0.7892	0.7904	0.8089*	0.7718	0.7935	0.7912	0.7978	0.7935	0.7912	0.7908	**0.8227** (25, 14)
Jazz	0.9549	0.9701*	0.9614	0.9503	0.9026	0.9541	0.9658	0.9503	0.9541	0.9658	0.9475	**0.9725** (46, 36)
USAir	0.9349	0.9527*	0.9466	0.9269	0.8949	0.9270	0.9503	0.9238	0.9270	0.9503	0.9395	**0.9590** (140, 90)
NS	0.9382	0.9386	0.9386	0.9429*	0.8765	0.9394	0.9358	0.9405	0.9394	0.9358	0.9421	**0.9437** (720, 120)
PPI	0.8934	0.8943	0.8941	0.9417	0.9342	0.9194	0.8940	0.9219	0.8568	**0.9479**	0.9273	0.9452*(750, 140)
KHN	0.8071	0.8137	0.8137	0.8739	0.8611	0.8387	0.8137	0.8585	0.7996	0.8767	0.8971*	**0.8977** (900, 90)
ADV	0.8994	0.9036	0.9030	0.9305	0.9176	0.9097	0.9029	0.9253	0.8927	0.9207	0.9312*	**0.9366** (800, 110)
ZWL	0.9364	0.9381	0.9379	0.9705	0.9611	0.9618	0.9380	0.9685	0.8092	0.9715	0.9752*	**0.9755** (950, 85)
HTC	0.8891	0.8893	0.8893	0.9047*	0.8328	0.8836	0.8893	0.8835	0.6160	0.7629	0.8850	**0.9124** (880, 88)
AUC_{avg}	0.8358	0.8439	0.8420	0.8804	0.8476	0.8428	0.8425	0.8888*	0.8182	0.8320	0.8709	**0.9227**

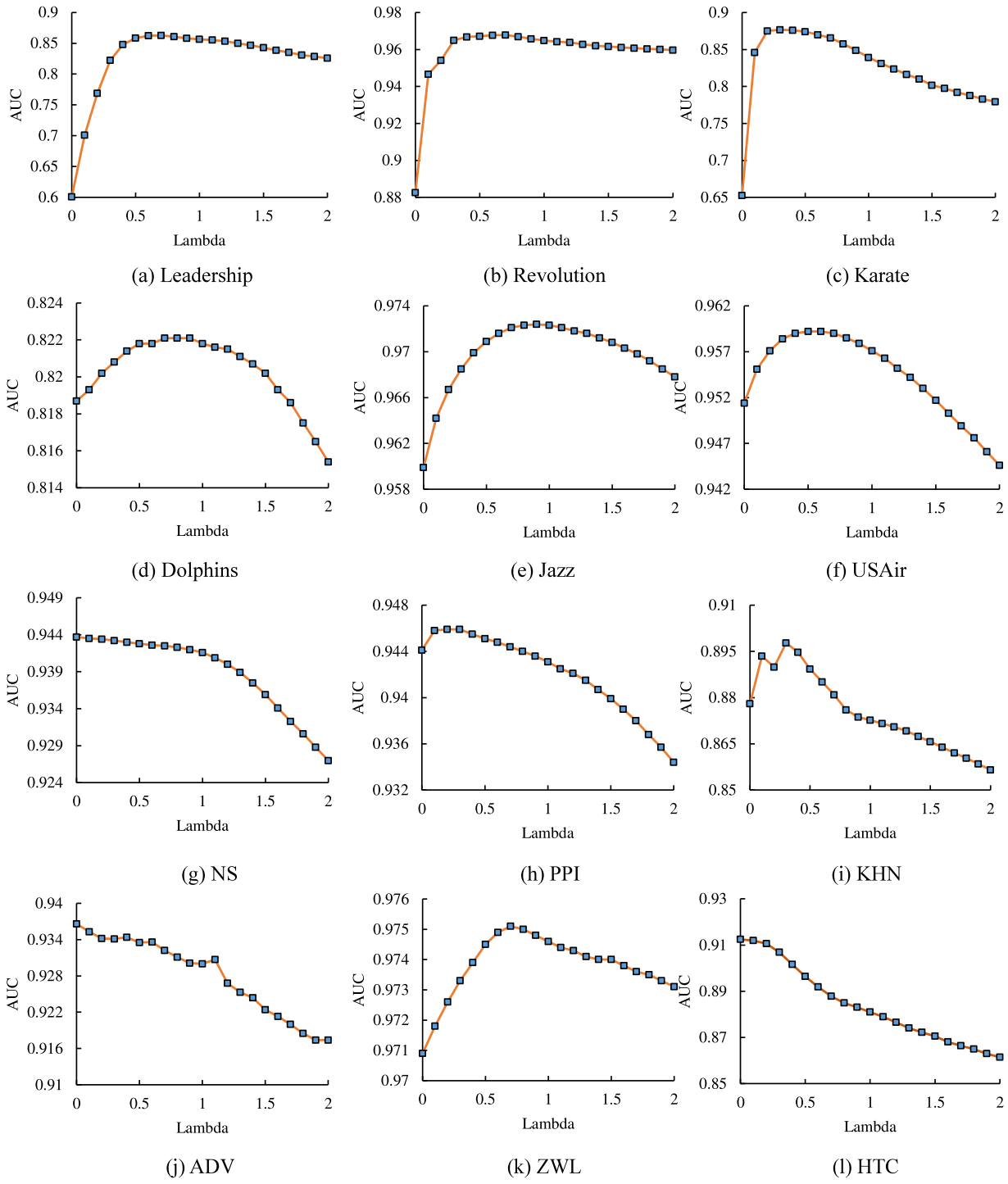

Fig. 5. The regularization parameter analysis. We analyzed the effect of λ in the different real networks in terms of *AUC*. The value of λ varied from 0 to 2.0 with the step size of 0.1. The results showed that *ICP* was sensitive to λ.

Moreover, *ICP* has the best AUC result 0.8762 in Karate, although RWR has the close performance with the AUC result 0.8747, *ICP* also has the better precision result 0.1141, which is higher than 0.0949 of RWR. We can find the similar results in KHN, ADV and ZWL, i.e., although RWR has the second best AUC results, which are close to the best AUC results of *ICP*, their precision results are worse than *ICP*. In such datasets, we still think *ICP* is better than RWR since AUC measures the performance of a link prediction method as a whole while precision only considers the top edges with the highest scores. The only exception can be found that, the AUC results of *ICP* in PPI is 0.9452, which is a bit lower than 0.9479 of Cos^{+}. However, the precision result of *ICP* is 0.1202, which not only is the highest among all methods, but also is much better the precision result 0.0462 of Cos^{+}. Therefore, we can conclude that, the *ICP* models present significant superiority performance over the baseline methods on the whole. For clarity, the results are also shown in the radar charts of Fig. 6, in which the performance of *ICP* models is indicated by a red dotted line. We can see that, the *ICP* models have obvious better performance than the baseline methods since it is at the outermost layer in terms of the AUC results. However, it is worth noticing that *ICP* has the time complexity $\mathcal{O}(n^3)$, which is higher than or at least equal with the baseline methods. This is because that the feature construction of *ICP* involves the calculations of clustering coefficient,

Table 8
The precision results of different models. Note that the best results are indicated in bold and the second best results are indicated with asterisks. Moreover, L was set to the number of edges in each network to calculate the precision results.

Method	CN	RA	AA	LP	L3	CCPA	CNDP	Katz	ACT	Cos^+	RWR	ICP (k, r)
Leadership	0.0141	0.0182	0.0060	0.0040	0.0758*	0.0152	0.0111	0.0060	0.0384	0.0151	0.0081	**0.0808** (42, 20)
Revolution	0.0013	0.0038	0.0006	0.0013	0.0275	0.0000	0.0019	0.0044	0.1025*	0.0013	0.0069	**0.1087** (40, 19)
Karate	0.0410	0.0731	0.0641	0.0538	0.0769	0.0397	0.0705	0.0538	0.0615	0.0436	0.0949*	**0.1141** (27, 16)
Dolphins	0.0541	0.0509	0.0528	0.0623	0.0597	0.0044	0.0742*	0.0629	0.0245	0.0384	0.0396	**0.0736** (25, 14)
Jazz	0.1608	0.1778*	0.1665	0.1564	0.1258	0.0113	0.1729	0.1564	0.0652	0.1480	0.1532	**0.1780** (46, 36)
USAir	0.1164	0.1465*	0.1251	0.1169	0.1137	0.0151	0.1367	0.1169	0.0927	0.1060	0.1113	**0.1493** (140, 90)
NS	0.1313	0.1608*	0.1578	0.1304	0.0857	0.0007	**0.1683**	0.1304	0.0004	0.0009	0.1246	0.1269 (720, 120)
PPI	0.0982	0.1068	0.1050	0.0975	0.0786	0.0016	0.1090*	0.0975	0.0034	0.0462	0.0468	**0.1202** (750, 140)
KHN	0.0244	0.0309	0.0326*	0.0276	0.0212	0.0083	0.0322	0.0276	0.0048	0.0083	0.0281	**0.0354** (900, 90)
ADV	0.0342	**0.0401**	0.0386*	0.0333	0.0269	0.0062	0.0375	0.0332	0.0047	0.0108	0.0250	0.0310 (800, 110)
ZWL	0.0433	**0.0496**	0.0481	0.0425	0.0325	0.0020	0.0489*	0.0425	0.0056	0.0189	0.0356	0.0451 (950, 85)
HTC	0.0506	0.0617	0.0631*	0.0487	0.0285	0.0003	**0.0663**	0.0488	0.0023	0.0002	0.0478	0.0340 (880, 88)

average shortest path length, degree centrality, closeness centrality, betweenness centrality and average neighbor degree. Such calculations are complex, but the complexity can be reduced by replacing the features with node semantic attributes if there are additional data available. The other disadvantage of *ICP* is its hyper parameters, which play an important role in dimension reduction, matrix factorization and the *ICP* optimization. In this work, we optimized such parameters based on empirical experience to obtain the optimal values for each network. Although we do not have the test set to testify such parameters in advance for real world problems, we can obtain optimal parameters by splitting existing datasets into training set and test set and tuning such parameters in terms of the AUC results.

6. Discussion

Compared with traditional link prediction methods, we proposed a novel link prediction approach based on inductive matrix completion. The node features include not only basic topological information, but also node local, quasi-local and importance properties. The results in Section 5 showed that, our *ICP* models had better performance than traditional prominent link prediction algorithms. If we compare such methods in terms of the average AUC results in all datasets, *ICP* performs much better than the baseline methods, as shown in Fig. 7. We can see that, the improvements of *ICP* in terms of the average AUC results are ranging from 3.81% $\sim$ 12.77% compared with the baseline methods. Moreover, if we take the second best AUC values as the reference lines, we can find that the improvements are ranging from 0.08% $\sim$ 3.54%. More precisely, as shown in Fig. 8, the AUC results are improved by 3.54%, 1.71%, 1.32%, 0.85%, 0.66%, 0.58% compared to the second best methods, i.e, the best baseline methods in Leadership, Dolphins, Revolution, HTC, USAir and ADV, respectively. Although the AUC percentages do not improve a lot in NS, KHN and ZWL, the improvements are considerable since such datasets are much larger. Although *ICP* does not have the best performance in PPI, it has very close performance with Cos^+, which is in the second place of the AUC results.

Moreover, it is feasible for our method to predict missing links in large-scale networks. According to Eq. (16), besides the features F_i and F_j of node i and j, the connection probability between two nodes is only related with the low-rank matrix Z, which is the product of W and H^T that are learned by solving the optimization problem shown in Eq. (17). In other words, we can use representative nodes to solve W and H by a supervised learning task if a network is in large scale. We can use *ICP* for large-scale networks in three steps. First, we need to find some representative nodes and calculate their features. Second, the *ICP* model is trained by solving W and H iteratively by the features of such nodes. Finally, we predict the connection probability of two nodes by feeding their features into the *ICP* model. We can see that, for large-scale networks, *ICP* only needs representative nodes instead of all their nodes to train the model and the only difficulty for *ICP* to predict the missing links in large-scale networks lies in the node feature construction. This is because the calculation of closeness centrality and average shortest path length is too time-consuming. Moreover, the first-order, the second-order and the third-order paths are difficult to calculate if a network in large scale. In order to perform link prediction in large-scale networks, it is necessary to choose the features that can be relatively easier calculated. Although the accuracy of prediction may be affected, the prediction efficiency will be improved. Therefore, we need to make a trade-off between the prediction performance and efficiency when we use *ICP* to predict missing links in large-scale networks. The other possible solution is reducing the *ICP* complexity by replacing the features with node semantic attributes if there are additional semantic data available.

In this work, *ICP* only considers the first-order, the second-order and the third-order paths of a network as the node local features instead of the global structural topology information. In order to prove whether the global structural topology information is helpful for improving the *ICP* performance, we revised *ICP* by adding the Katz results as a feature to representing global structural topology information, and Table 9 shows their AUC results. Note that ICP^* denotes the *ICP* model with the global structural topology additionally and the parameters used for both *ICP* and ICP^* were kept the same except the dimension of the node features in the experiments in order to have a reasonable comparison. We can see that, although the ICP^* models have better performance than the *ICP* models without the global information in Revolution, Jazz, USAir, PPI, KHN and ADV, their AUC results are worse in the rest six networks. Moreover, the AUC results of ICP^* and *ICP* in all networks are very close. Therefore, there is no obvious improvements by considering the global structural topology information. In fact, considering the global structural topology information in *ICP* even increases its complexity, since the weight attenuation factor that controls the weight of the different paths of the global structural topology information needs to be optimized additionally.

7. Conclusion

In this paper, we proposed a novel link prediction approach *ICP* based on inductive matrix completion. The approach aimed to solve a low-rank projection matrix which can map the low-dimensional feature space from one node to the other. The approach explored the combination of different structural topology information with the node importance properties to comprehensively present the node features via feature construction and selection. In order to demonstrate the superiority of *ICP*, we compared it with eleven related efforts including two recent methods proposed in 2020 in twelve different real networks. The results showed that *ICP* had stable performance and good universality in all networks compared with the baseline methods, which have various performance in different datasets. Compared with the baseline methods, the improvements of *ICP* in terms of the average AUC results are ranging from 3.81% $\sim$ 12.77%. Moreover, the AUC performance is improved by 0.08% $\sim$ 3.54% compared with the best baseline method. The limitation of *ICP* lies in its high computational complexity since

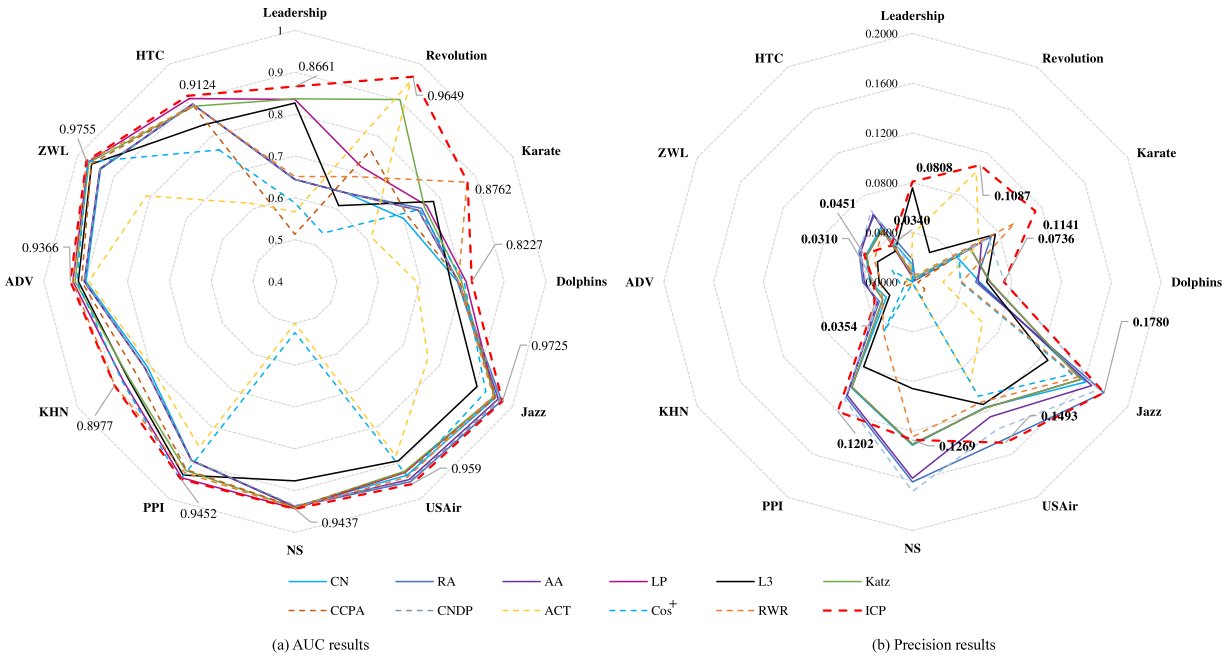

Fig. 6. The comparison of different models with radar charts. The AUC performance of *ICP* models is indicated by a red solid line at the outermost layer. In other words, the *ICP* models have superiority over the baseline methods.

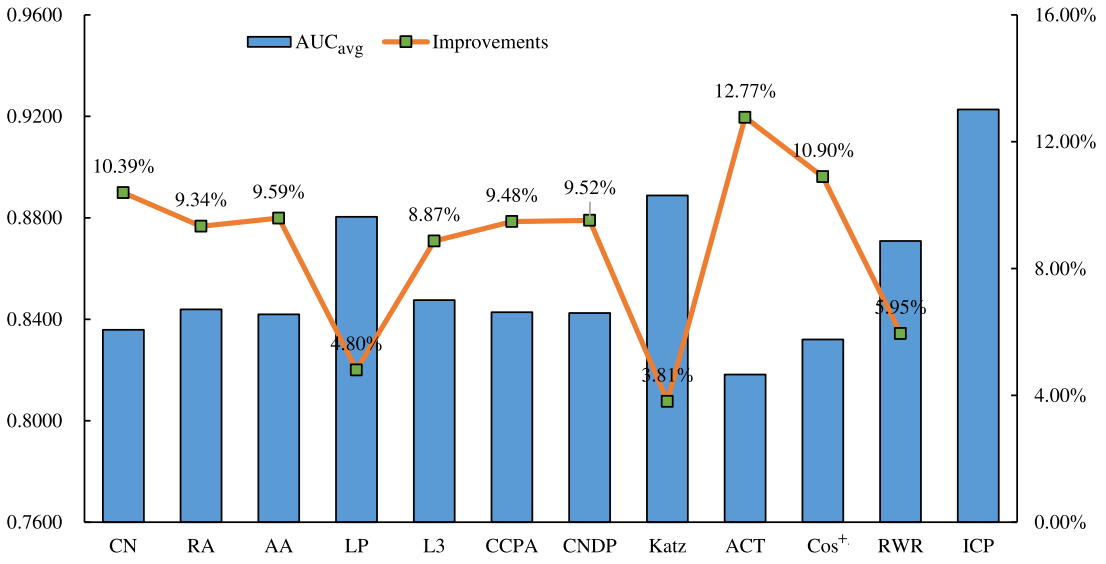

Fig. 7. Comparison of different methods in terms of the average AUC results. *ICP* performs much better than the baseline methods in terms of the average AUC results in all datasets. Compared with the baseline methods, the improvements of *ICP* in terms of the average AUC results are ranging from 3.81% ∼ 12.77%.

Table 9
The performance of the *ICP* models with/without the global structural topology information. ICP^* denotes the *ICP* model with the global structural topology additionally. We can see that, the performance of the ICP^* models is not improved obviously, and it is even worse than the *ICP* models without the global information in some datasets.

Method	Leadership	Revolution	Karate	Dolphins	Jazz	USAir	NS	PPI	KHN	ADV	ZWL	HTC
ICP	0.8661	0.9649	0.8762	0.8227	0.9725	0.9590	0.9437	0.9452	0.8977	0.9366	0.9755	0.9124
ICP^*	0.8651	0.9651	0.8588	0.8188	0.9734	0.9591	0.9435	0.9456	0.9014	0.9371	0.9751	0.9112

the feature construction involves the calculations of average shortest path length, closeness centrality, etc. Such calculations are complex, but the complexity can be reduced replacing the features with node semantic attributes if there are additional semantic data available. The other disadvantage of *ICP* is its hyper parameters, which are important for the *ICP* performance. In this work, we optimized such parameters based on empirical experience to obtain optimal parameters for each network. In real applications, we can obtain optimal parameters by splitting existing datasets into training set and test set and tuning the parameters in terms of the AUC results. The other benefit of our approach is that it provides a potential link prediction solution for large-scale networks, since building *ICP* models is a supervised learning task, in which the underlying low-rank matrix can be solved by on representative nodes instead of all nodes of their network. Moreover, there are richer node semantic attributes available. For example, users

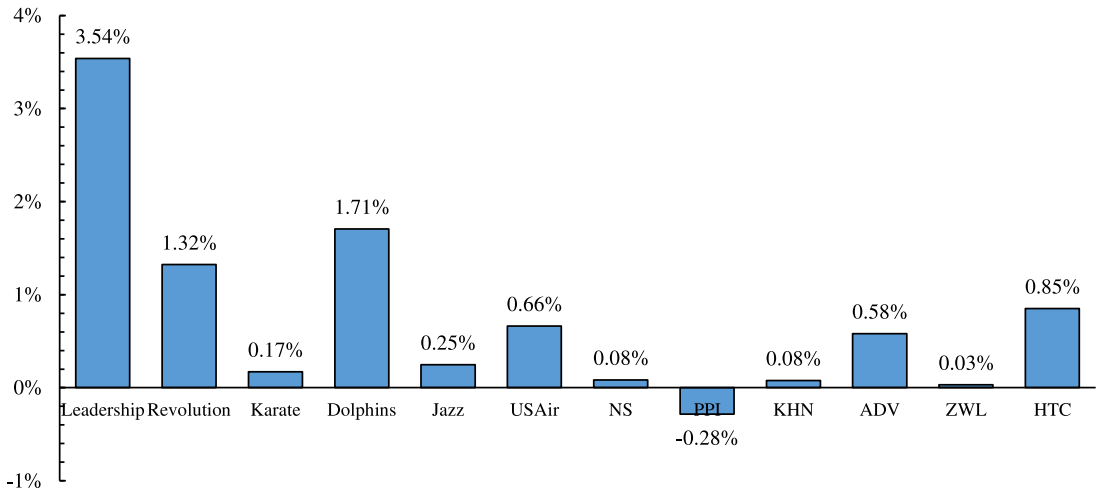

Fig. 8. The AUC improvements of the *ICP* models in different datasets. The *ICP* models were further analyzed by taking the second best AUC values as the reference lines. The *ICP* models improve 3.54%, 1.71%, 1.32%, 0.85%, 0.66%, 0.58% compared to the second best methods, i.e, the best baseline methods in Leadership, Dolphins, Revolution, HTC, USAir and ADV, respectively. Although the AUC percentages do not improve a lot in NS, KHN and ZWL, the improvements are considerable since such datasets are much larger.

in online social networking sites might be annotated with, e.g., demographical information and interests and it is possible to employ such information as the node features in our future work.

CRediT authorship contribution statement

Zhili Zhao: Writing – original draft, Supervision, Conceptualization. **Zhuoyue Gou:** Data curation, Investigation, Methodology, Validation, Software. **Yuhong Du:** Writing – review & editing. **Jun Ma:** Writing – review & editing. **Tongfeng Li:** Writing – review & editing. **Ruisheng Zhang:** Project administration.

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.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Project No. 61702240) and Natural Science Foundation of Gansu Province, China (Project No. 21JR7RA460).

Appendix. List of symbols and notations

* A: the adjacency matrix of a network.
* $act(i, j)$: the average commuting time of node i and node j of a network.
* B: the node feature space involving node degree, clustering coefficient, average shortest path length, degree centrality, closeness centrality, betweenness centrality and average neighbor degree.
* C: the average clustering coefficient of a network.
* $|C_z|$: the number of neighbors of z which consist of the common neighbors of i and j in addition to i and j. z is a common neighbor of nodes i and j.
* d: the average degree of a network.
* $d(i, j)$: the distance between nodes i and j.
* E: the set of edges of a network.
* F: the node feature space of a network.
* F_i: the feature vector of node i.
* f_j: the jth feature of a node.
* G: an undirected and unweighted network.
* k: the reduced dimension of the node features in *ICP*.
* k_i: the degree of node i.
* M: the node association matrix of a network.
* m: the number of edges in a network.
* n: the number of nodes in a network.
* q_i: a probability vector, which contains the probabilities that a particle at node i moves to other nodes of a network.
* q_{ij}: the probability that a particle at node i moves to node j.
* r: the dimension of W and H of *ICP*.
* R_+: the set of non-negative real numbers.
* s_{ij}: the connection probability between nodes i and j.
* T: the third-order path space of a network.
* $t(i, j)$: the average number of steps a random particle needs to take from node i to node j.
* $tr(X)$: the trace of a square matrix X, i.e., the sum of the diagonal elements of X.
* U: the second-order path space of a network.
* V: the set of nodes of a network.
* $|V|$: the number of nodes in the node set V.
* W and H: two smaller matrices used to construct Z of the *ICP* model.
* z: a common neighbor of nodes i and j.
* Z: the low-rank matrix of inductive matrix completion.
* α: in RWR, a node is iteratively move to a neighbor randomly with probability α or it will return to the node i with probability $1 - \alpha$.
* $\Gamma(i)$: the set of neighbor nodes of node i.
* λ: the regularization parameter used in matrix completion-based link prediction.
* $\sigma_{j,k}$: the total number of shortest paths existing between nodes j and k.
* $\sigma_{j,k}(i)$: the number of shortest paths that passes through node i.
* $\triangle_i$: the number of triangles through node i.

References

Ahmad, I., Akhtar, M., Noor, S., & Shahnaz, A. (2020). Missing link prediction using common neighbor and centrality based parameterized algorithm. *Scientific Reports*, *10*, 364.

Ahmed, N. M., Chen, L., Wang, Y., Li, B., Li, Y., & Liu, W. (2018). DeepEye: Link prediction in dynamic networks based on non-negative matrix factorization. *Big Data Mining and Analytics*, *1*(1), 19–33.

Assouli, N., Benahmed, K., & Gasbaoui, B. (2021). How to predict crime — informatics-inspired approach from link prediction. *Physica A: Statistical Mechanics and its Applications*, *570*, Article 125795.

Aziz, F., Gul, H., Muhammad, I., & Uddin, I. (2020). Link prediction using node information on local paths. *Physica A: Statistical Mechanics and its Applications*, *557*, Article 124980.

Bag, S., Kumar, S. K., & Tiwari, M. K. (2019). An efficient recommendation generation using relevant jaccard similarity. *Information Sciences*, *483*, 53–64.

Behrouzi, S., Shafaeipour Sarmoor, Z., Hajsadeghi, K., & Kavousi, K. (2020). Predicting scientific research trends based on link prediction in keyword networks. *Journal of Informetrics*, *14*(4), Article 101079.

Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q. V. H., & Li, X. (2018). PME: Projected metric embedding on heterogeneous networks for link prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1177–1186). New York, USA: Association for Computing Machinery.

Chen, Z., Zhao, W., & Wang, S. (2021). Kernel meets recommender systems: A multi-kernel interpolation for matrix completion. *Expert Systems with Applications*, *168*, Article 114436.

Das, S., & Das, S. K. (2017). A probabilistic link prediction model in time-varying social networks. In *2017 IEEE international conference on communications* (pp. 1–6).

Daud, N. N., Ab Hamid, S. H., Saadoon, M., Sahran, F., & Anuar, N. B. (2020). Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, *166*, Article 102716.

Ding, X., Xia, J.-F., Wang, Y.-T., Wang, J., & Zheng, C.-H. (2019). Improved inductive matrix completion method for predicting microrna-disease associations. In D.-S. Huang, K.-H. Jo, & Z.-K. Huang (Eds.), *Intelligent computing theories and application* (pp. 247–255). Cham: Springer International Publishing.

Gaucher, S., & Klopp, O. (2021). Maximum likelihood estimation of sparse networks with missing observations. *Journal of Statistical Planning and Inference*, *215*, 299–329.

Javari, A., Qiu, H., Barzegaran, E., Jalili, M., & Chang, K. (2017). Statistical link label modeling for sign prediction: Smoothing sparsity by joining local and global information. In G. Karypis, S. Alu, V. Raghavan, X. Wu, & L. Miele (Eds.), *Proceedings - 17th IEEE international conference on data mining* (pp. 1039–1044). United States: Institute of Electrical and Electronics Engineers Inc..

Karimi, F., Lotfi, S., & Izadkhah, H. (2021). Community-guided link prediction in multiplex networks. *Journal of Informetrics*, *15*(4), Article 101178.

Kaya, B., & Gündoğan, E. (2018). Evaluating reliability of question-disease relations in online health forms: A link prediction approach. *Telematics and Informatics*, *35*(7), 1799–1808.

KONECT (2021). The KONECT project. http://konect.cc/networks/. (Accessed 5 August 2021).

Kuang, J., & Scoglio, C. (2021). Layer reconstruction and missing link prediction of a multilayer network with maximum *aposteriori* estimation. *Physical Review E*, *104*, Article 024301.

Kumar, A., Singh, S. S., Singh, K., & Biswas, B. (2020). Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, *553*, Article 124289.

Li, S., Song, X., Lu, H., Zeng, L., Shi, M., & Liu, F. (2020). Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm. *Expert Systems with Applications*, *139*, Article 112839.

Li, T., Wang, B., Jiang, Y., Zhang, Y., & Yan, Y. (2018). Restricted Boltzmann machine-based approaches for link prediction in dynamic networks. *IEEE Access*, *6*, 29940–29951.

Li, R.-H., Yu, J., & Liu, J. (2011). Link prediction: the power of maximal entropy random walk. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 1147–1156).

Liu, G. (2022). An ecommerce recommendation algorithm based on link prediction. *Alexandria Engineering Journal*, *61*(1), 905–910.

Liu, S., Ji, X., Liu, C., & Bai, Y. (2017). Extended resource allocation index for link prediction of complex network. *Physica A: Statistical Mechanics and its Applications*, *479*, 174–183.

Liu, X., & Li, X. (2021). A social network link prediction method based on stacked generalization. *The Computer Journal*, bxab102.

Liu, Y., Tong, H., Xie, L., & Tang, Y. (2015). Supervised link prediction using random walks. In X. Zhang, M. Sun, Z. Wang, & X. Huang (Eds.), *Social media processing* (pp. 107–118). Singapore: Springer Singapore.

Liu, Y., Zhao, C., Wang, X., Huang, Q., Zhang, X., & Yi, D. (2016). The degree-related clustering coefficient and its application to link prediction. *Physica A: Statistical Mechanics and its Applications*, *454*, 24–33.

Ma, X., Sun, P., & Qin, G. (2017). Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability. *Pattern Recognition*, *71*, 361–374.

Martínez, V., Berzal, F., Cubero, J.-C., & Cimini, G. (2019). NOESIS: A framework for complex network data analysis. *Complexity*, *2019*.

Masuda, N., Porter, M. A., & Lambiotte, R. (2017). Random walks and diffusion on networks. *Physics Reports*, *716–717*, 1–58.

Mongia, A., & Majumdar, A. (2021). Matrix completion on learnt graphs: Application to collaborative filtering. *Expert Systems with Applications*, *185*, Article 115652.

Nasiri, E., Berahmand, K., & Li, Y. (2021). A new link prediction in multiplex networks using topologically biased random walks. *Chaos, Solitons & Fractals*, *151*, Article 111230.

Nayyeri, M., Cil, G. M., Vahdati, S., Osborne, F., Rahman, M., Angioni, S., Salatino, A., Recupero, D. R., Vassilyeva, N., Motta, E., & Lehmann, J. (2021). Trans4E: Link prediction on scholarly knowledge graphs. *Neurocomputing*.

Pech, R., Hao, D., Pan, L., Cheng, H., & Zhou, T. (2017). Link prediction via matrix completion. *EPL (Europhysics Letters)*, *117*(3), 38002.

Rafiee, S., Salavati, C., & Abdollahpouri, A. (2020). CNDP: Link prediction based on common neighbors degree penalization. *Physica A: Statistical Mechanics and its Applications*, *539*, Article 122950.

Si, S., Chiang, K.-Y., Hsieh, C.-J., Rao, N., & Dhillon, I. S. (2016). Goal-directed inductive matrix completion. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1165–1174). New York, USA: Association for Computing Machinery.

Singh, S. S., Mishra, S., Kumar, A., & Biswas, B. (2020). CLP-ID: Community-based link prediction using information diffusion. *Information Sciences*, *514*, 402–433.

Song, A., Liu, Y., Wu, Z., Zhai, M., & Luo, J. (2019). A local random walk model for complex networks based on discriminative feature combinations. *Expert Systems with Applications*, *118*, 329–339.

Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M., & Mucha, P. (2019). Stochastic block models with multiple continuous attributes, *4*(1).

Su, Z., Zheng, X., Ai, J., Shen, Y., & Zhang, X. (2020). Link prediction in recommender systems based on vector similarity. *Physica A: Statistical Mechanics and its Applications*, *560*, Article 125154.

Vural, H., & Kaya, M. (2018). Prediction of new potential associations between LncRNAs and environmental factors based on KATZ measure. *Computers in Biology and Medicine*, *102*, 120–125.

Wahid-Ul-Ashraf, A., Budka, M., & Musial, K. (2019). How to predict social relationships — Physics-inspired approach to link prediction. *Physica A: Statistical Mechanics and its Applications*, *523*, 1110–1129.

Wang, W., Feng, Y., Jiao, P., & Yu, W. (2017). Kernel framework based on non-negative matrix factorization for networks reconstruction and link prediction. *Knowledge-Based Systems*, *137*, 104–114.

Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., & Liu, Q. (2018). SHINE: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 592–600). New York, USA: Association for Computing Machinery.

Wu, Z., & Chen, Y. (2016). Link prediction using matrix factorization with bagging. In *2016 IEEE/ACIS 15th international conference on computer and information science* (pp. 1–6).

Yang, M., & Xu, S. (2021). A novel deep quantile matrix completion model for top-N recommendation. *Knowledge-Based Systems*, *228*, Article 107302.

Zhang, X., Pang, W., & Xia, Y. (2018). An intermediary probability model for link prediction. *Physica A: Statistical Mechanics and its Applications*, *512*, 902–912.

Zhao, H., Du, L., & Buntine, W. (2017). Leveraging node attributes for incomplete relational data. In *Proceedings of the 34th international conference on machine learning (vol. 70)* (pp. 4072–4081). JMLR.org.

Zhao, Z., Zhou, H., Li, C., Tang, J., & Zeng, Q. (2021). DeepEmLAN: Deep embedding learning for attributed networks. *Information Sciences*, *543*, 382–397.

Zhou, Y., Wu, C., & Tan, L. (2021). Biased random walk with restart for link prediction with graph embedding method. *Physica A: Statistical Mechanics and its Applications*, *570*, Article 125783.