

Research Article

Finding Missing Links in Complex Networks: A Multiple-Attribute Decision-Making Method

Longjie Li ¹, Shenshen Bai,¹ Mingwei Leng ², Lu Wang,¹ and Xiaoyun Chen¹

¹School of Information Science & Engineering, Lanzhou University, Lanzhou 730000, China

²College of Educational Science and Technology, Northwest Minzu University, Lanzhou 730030, China

Correspondence should be addressed to Longjie Li; ljli@lzu.edu.cn

Received 29 March 2018; Revised 28 June 2018; Accepted 25 July 2018; Published 2 September 2018

Academic Editor: Guido Caldarelli

Copyright © 2018 Longjie Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Link prediction, which aims to forecast potential or missing links in a complex network based on currently observed information, has drawn growing attention from researchers. To date, a host of similarity-based methods have been put forward. Usually, one method harbors the idea that one similarity measure is applicable to various networks, and thus has performance fluctuation on different networks. In this paper, we propose a novel method to solve this issue by regarding link prediction as a multiple-attribute decision-making (MADM) problem. In the proposed method, we consider RA, LP, and CAR indices as the multiattribute for node pairs. The technique for order performance by similarity to ideal solution (TOPSIS) is adopted to aggregate the multiattribute and rank node pairs. The proposed method is not limited to only one similarity measure, but takes separate measures into account, since different networks may have different topological structures. Experimental results on 10 real-world networks manifest that the proposed method is superior in comparison to state-of-the-art methods.

1. Introduction

In recent years, the research of link prediction in complex networks has captured much attention of researchers from various disciplines [1] not only because many available real-world networks are incomplete [2, 3] but also because link prediction is closely related to many other problems [4, 5]. It offers one possible way to understand the evolution of networks [6–8] and gives help to find potential interactions between proteins in biological networks [9, 10]. Link prediction has also been applied to friendship suggestions in social networks [11, 12], products recommendation in e-commerce systems [13, 14], collaboration prediction in coauthorship networks [15, 16] and so on.

As a fundamental research hotspot in complex network analysis, link prediction can uncover missing or latent existent links and point out spurious links in a network based on the observed information [9, 17, 18]. In this paper, we center on finding missing links. Generally speaking, two unconnected nodes with a high similarity score are deemed to be likely to have a missing link [5, 17]. That is the basic

hypothesis of the so-called similarity-based approaches. In a similarity-based approach, similarity scores of nonexisting links are estimated first, and then links at the top of the sorted score list with descending order are predicted as missing ones [17]. To date, great efforts have been devoted to link prediction based on observed network structure information [1, 4, 17], such as common neighbors [19–21], local paths [22, 23], and triangle structures [24, 25]. Along this line, a plethora of similarity-based indices and methods have been proposed. These approaches usually focus on only one similarity measure and assume that it is applicable to all networks. However, different networks always have separate inner structural features [26, 27]. Thus, the prediction performances of these approaches are not stable on different networks.

On the other hand, likelihood-based algorithms aim at identifying the most likely generative model for a network and then estimate the connection likelihood of any two nodes according to the considered model [28, 29]. These algorithms usually assume that a network has a known structure and build a model, such as hierarchical structure model [29]

and stochastic block model [9], to fit the structure and evaluate model parameters using statistical methods [4]. However, the structure of a network is not always known, and a model cannot be suitable for all networks.

In this paper, we regard link prediction as a multiple-attribute decision-making (MADM) (also called multicriteria decision-making (MCDM)) problem. MADM is an approach that has been designed to select a preferred alternative, classify alternatives in a small number of categories, and/or rank alternatives in a subjective preference order [30, 31]. It is a widely used tool in various fields [32, 33]. Among numerous MADM methods put forward to solve real-world decision problems, the technique for order preference by similarity to ideal solution (TOPSIS) continues to work satisfactorily across diverse application areas [31, 34]. TOPSIS was originally proposed to give help to determine the best alternative with a finite number of criteria [35]. TOPSIS makes full use of attribute information, affords a cardinal ranking of alternatives, and does not require attribute preferences to be independent [36]. As a well-known classical MADM method, TOPSIS has received considerable interest from researchers and practitioners [31]. In a complex network analysis, TOPSIS was used to identify influential nodes [34, 37, 38]. In this paper, we apply TOPSIS to reveal missing links. The similarity scores based on different similarity indices are considered as multiattribute to make the decision for ranking nonexistent links.

The work in [39] also adopted TOPSIS in link prediction; however, it is totally different from our work. In [39], TOPSIS is only used to evaluate the local centralities of common neighbors. The similarity score between two nodes is computed based on the local centralities of their common neighbors. In our method, TOPSIS is employed to evaluate the degrees of similarity of node pairs and determine the missing links. Three famous similarity indices, that is, RA [21], LP [21, 23], and CAR [24] are chosen as the multiattribute of TOPSIS. The reason for selecting these indices is that they are designed based on different but prominent structural features. Since each attribute is associated with a weight in TOPSIS, we present a new algorithm based on the known information about micro nodes to determine the weights. To verify the performance of the proposed method, we conduct experiments on 10 real-world networks from various fields. The experimental results demonstrate the stability and robustness of our method.

The rest of the paper is structured as follows. In Section 2, we give the description of the link prediction problem and the metrics for evaluating the accuracy of link prediction algorithms. Section 3 lists the baselines, and Section 4 introduces the proposed method. In Section 5, the experimental results and performance analysis of the proposed method are presented. Finally, Section 6 concludes this work.

2. Problem Description and Metric

Consider an undirected and unweighted network $G(V, E)$, in which V and E are the node set and link set, respectively. Multilinks and self-loops are not allowed in this study. For a network containing N nodes, the set that contains

$N(N-1)/2$ possible links is denoted by U , and then the set of unconnected node pairs is $U - E$. Each node pair in $U - E$ is assigned a similarity score according to a given similarity method. All unconnected node pairs are sorted in descending order according to their scores, and the node pairs at the top are most likely to have missing links [17].

Actually, we do not have the ground truth, that is, the missing links are not known. Therefore, to test the accuracy of a link prediction method, the link set E is randomly divided into two parts: training set E_{tr} and testing set E_{ts} , such that $E = E_{tr} \cup E_{ts}$ and $E_{tr} \cap E_{ts} = \emptyset$. Two standard metrics are employed to quantify the accuracy of link prediction algorithms: *AUC* [17] and *Precision* [40]. In this situation, the *AUC* value can be interpreted as the probability that a randomly selected missing link (i.e., a link in E_{ts}) is assigned a higher similarity score than a randomly selected nonexistent link (i.e., a link in $U - E$). In implementation, we perform n times of independent comparisons. If there are n_1 times that the missing link has a higher score and n_2 times that they have the same score, the *AUC* value is computed as

$$AUC = \frac{n_1 + 0.5n_2}{n}. \quad (1)$$

Precision characterizes the ratio of correctly predicted links within a given prediction list. If we take the top- L as the prediction list, among which m links are correctly predicted, then *Precision* is

$$Precision = \frac{m}{L}. \quad (2)$$

3. Baseline Prediction Methods

Up to date, many link prediction methods have been proposed [1, 4, 17]. Here, we list some state-of-the-art approaches used in this paper.

- (1) Resource allocation (RA) index [21]. This index models the resource allocation between two nodes through their shared neighbors. The amount of resource that one node received from another through their common neighbors is defined as their similarity, which is

$$RA(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}, \quad (3)$$

where $\Gamma(x)$ denotes the neighbor set of node x , and k_z is the degree of node z .

- (2) Adaptive degree penalization (ADP) index [27]. This method is proposed to automatically adapt to the network structure. It tries to estimate the best-performing degree penalization by the network clustering coefficient. The formal definition is

$$ADP(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} k_z^{-\beta C}, \quad (4)$$

where β is a constant, C is the average clustering coefficient of the network. β is set to 2.5 in this paper according to the authors' suggestion.

- (3) CAR index [24]. This index stems from both node-based and link-based perspectives, which suggests that two nodes are more likely to link together if their common neighbors are members of a local community. The similarity function defined by this index can be computed as

$$CAR(x, y) = CN(x, y) \cdot \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\Gamma(x) \cap \Gamma(y) \cap \Gamma(z)|}{2}, \quad (5)$$

where $CN(x, y)$ is the number of common neighbors between x and y , which is

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)|. \quad (6)$$

- (4) Clustering coefficient for link prediction (CCLP) index [25]. This index is inspired by the idea that triangle information is very useful in estimating similarities of nodes. It computes the similarity between two nodes by employing the clustering coefficient of shared neighbors:

$$CCLP(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} CC_z, \quad (7)$$

where CC_z denotes the clustering coefficient of node z , which is

$$CC_z = \frac{2t_z}{k_z(k_z - 1)}, \quad (8)$$

in which t_z is the number of triangles passing through node z .

- (5) Local path (LP) index [21, 23]. This metric extends the horizon of common neighbors to three hops paths, which estimates the similarity of two nodes as

$$LP(x, y) = \sum_{i=1}^2 \lambda^{i-1} |P^i(x, y)|, \quad (9)$$

where $|P^i(x, y)|$ is the number of paths between x and y with length i ; λ is an adjustable parameter. Generally, λ is a very small number, and we set $\lambda = 0.001$ in this paper.

- (6) Mutual information (MI) index [41]. This method defines nodes' similarities from the perspective of information theory by computing the conditional self-information of the existence of a link between two unconnected nodes given their common

neighbors. The similarity between two nodes is estimated as

$$\begin{aligned} MI(x, y) &= -I(L_{xy}^1 | \Gamma(x) \cap \Gamma(y)) \\ &= \sum_{z \in \Gamma(x) \cap \Gamma(y)} I(L_{xy}^1; z) - I(L_{xy}^1), \end{aligned} \quad (10)$$

where $I(L_{xy}^1)$ is the self-information of x and y being connected, and $I(L_{xy}^1; z)$ is the mutual information of the existence of a link between x and y and the shared neighbor z . $I(L_{xy}^1)$ and $I(L_{xy}^1; z)$ are, respectively, calculated as

$$\begin{aligned} I(L_{xy}^1) &= -\log_2 \left(1 - \prod_{i=1}^{k_y} \frac{(|E| - k_x) - i + 1}{|E| - i + 1} \right), \\ I(L_{xy}^1; z) &= \frac{1}{k_z(k_z - 1)} \sum_{\substack{u, v \in \Gamma(z), \\ u \neq v}} \left(I(L_{uv}^1) \right. \\ &\quad \left. - I(L_{xy}^1 | z) \right) \\ &= \frac{1}{k_z(k_z - 1)} \sum_{\substack{u, v \in \Gamma(z), \\ u \neq v}} \left(I(L_{uv}^1) \right. \\ &\quad \left. + \log_2(CC_z) \right). \end{aligned} \quad (11)$$

- (7) Adaptive fusion model base on logistic regression (LR index) [26]. The method was proposed based on the observation that: (i) the roles of different structural features in a network are utterly different, and (ii) the role of a structural feature in different modules is also different [26]. LR index defines the connection probability of node pair (x, y) as

$$P(x, y) = \max \{P_{M_1}(x, y), \dots, P_{M_k}(x, y)\}, \quad (12)$$

where $P_{M_k}(x, y)$ is the connection probability of (x, y) in module M_k , which is

$$P_{M_k}(x, y) = \frac{1}{1 + \exp \left(- \left(\beta_0 + \sum_{l=1}^L \beta_l S_{M_k}^{F_l}(x, y) \right) \right)}, \quad (13)$$

where $S_{M_k}^{F_l}(x, y)$ denotes the similarity score induced by feature F_l for (x, y) in module M_k . Three scenarios of modules were considered in [26]; correspondingly, the index based on these scenarios was denoted as LR_1 , LR_2 , and LR_m . LR_m is a tradeoff between LR_1 and LR_2 ; hence, we use LR_m in our experiments. As in [26],

three features, that is, CN (see (6)), PA (see (14)), and DD (see (15)), are incorporated for LR_m in this paper.

$$PA(x, y) = k_x \cdot k_y, \quad (14)$$

$$DD(x, y) = |k_x - k_y|. \quad (15)$$

- (8) Local nave Bayes (LNB) method [42]. This method calculates the connection likelihood between two nodes based on the LNB model. The likelihood score of node pair (x, y) is defined as

$$r(x, y) = s^{-1} \prod_{z \in \Gamma(x) \cap \Gamma(y)} sR_z, \quad (16)$$

where $s = |U|/|E_{tr}| - 1$, which is a constant for a network, and $R_z = (N_{\Delta z} + 1)/(N_{\wedge z} + 1)$, in which $N_{\Delta z}$ and $N_{\wedge z}$ are, respectively, the number of connected and unconnected node pairs whose common neighbors include z .

In [42], an exponent function $f(k_z)$, which is a function of the degree of node z , is added to the item sR_z in (16). Using a logarithmic function on both sides and neglecting the constant s^{-1} , a linear formula of connection likelihood is obtained, which is

$$r'(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} f(k_z) \log(sR_z). \quad (17)$$

In this paper, we use $f(k_z) = 1$; the corresponding method is named LNB-CN [42], which is

$$LNB-CN(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} (\log(s) + \log(R_z)). \quad (18)$$

- (9) Maximization entropy (MaxE) method [28]. This method is a likelihood-based algorithm based on a series of results concerning constrained entropy maximization [43]. MaxE uses the observed portion of a network as constraints of a maximization procedure defined within the exponential random graph (ERG) framework [43]. In the case of undirected and unweighted networks, the ERG framework maximizes the likelihood function $\mathcal{L} = \ln P(A^o)$, where A^o is the adjacency matrix of the observed portion, that is, training set, and $P(A^o)$ is defined as

$$P(A^o) = \prod_{i < j} p_{ij}^{a_{ij}} (1 - p_{ij})^{1 - a_{ij}}, \quad (19)$$

where $a_{ij} \in A^o$ and $p_{ij} = x_i x_j / (1 + x_i x_j)$. By solving the system of equations in (20), the maximization of likelihood can be obtained.

$$k_i(A^o) = \sum_{j(\neq i)} p_{ij} = \sum_{j(\neq i)} \frac{x_i x_j}{1 + x_i x_j} \forall i. \quad (20)$$

4. Methodology

A multitude of similarity indices has been carried out for link prediction in complex networks. In general, one similarity index only employs one or two structural features and assumes that they are suitable for all networks. However, different networks usually have different structural features; separate similarity indices need to be taken into account. To address this issue, a novel method is proposed in this paper to forecast missing links by considering link prediction as a multiple-attribute decision-making (MADM) problem. MADM is an approach to rank alternatives in a subjective preference order [30, 31]. In this paper, each potential missing link is viewed as an alternative, and each similarity index is considered as an attribute. In implementation, we adopt TOPSIS [35], a well-known classical MADM method, to uncover the missing links in this paper. Meanwhile, three classical similarity indices, that is, RA, LP, and CAR, are chosen as the multiattribute in our method. For convenience, our method is named LP_{TOPSIS} .

In the following of this part, we first introduce the TOPSIS method in Section 4.1. Then, the proposed method is presented in Section 4.2. Finally, a simple example is given to explain how the proposed method performs in Section 4.3.

4.1. TOPSIS Method. Technique for order preference by similarity to ideal solution (TOPSIS) is a simple but effective ranking method in conception and application [35]. The standard TOPSIS method attempts to determine the best alternative that simultaneously has the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution. The ranking of the alternatives is calculated according to the relative closeness to the ideal solution.

Given a decision matrix $\mathbf{X} = \{x_{ij}\}_{m \times n}$ of m alternatives and n criteria (decision attributes) in

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, \quad (21)$$

where x_{ij} denotes the value of the i th alternative under j th criteria in matrix \mathbf{X} . The procedures of the TOPSIS method are depicted as follows:

Step 1. Normalize the decision matrix \mathbf{X} by using the vector-normalization technique.

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, \quad (22)$$

where y_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) is the normalized value of i th alternative under the j th criteria.

Step 2. Calculate the weighted-normalized decision matrix $\mathbf{V} = \{v_{ij}\}_{m \times n}$ by using the associated weights.

$$v_{ij} = w_j \cdot y_{ij}, \quad (23)$$

where $w_j (j = 1, 2, \dots, n)$ is the weight of the j th criterion.

Step 3. Determine the positive ideal solution S^+ and negative ideal (anti-ideal) solution S^- , respectively.

$$S^+ = \{s_1^+, s_2^+, \dots, s_n^+\}, \quad (24)$$

$$S^- = \{s_1^-, s_2^-, \dots, s_n^-\}. \quad (25)$$

For the benefit criteria K_b :

$$s_j^+ = \left\{ \max_i v_{ij} \mid j \in K_b \right\}, \quad (26)$$

$$s_j^- = \left\{ \min_i v_{ij} \mid j \in K_b \right\};$$

and for the cost criteria K_c :

$$s_j^+ = \left\{ \min_i v_{ij} \mid j \in K_c \right\}, \quad (27)$$

$$s_j^- = \left\{ \max_i v_{ij} \mid j \in K_c \right\}.$$

Step 4. Obtain the Euclidean distance of each alternative from the positive ideal and the negative ideal solutions, respectively.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - s_j^+)^2}, \quad (28)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - s_j^-)^2}, \quad (29)$$

where $d_i^+ (i = 1, 2, \dots, m)$ and $d_i^- (i = 1, 2, \dots, m)$ are the Euclidean distances of i th alternative from the positive ideal and the negative ideal solutions, respectively.

Step 5. Compute the relative closeness of each alternative to the ideal solution.

$$C_i = \frac{d_i^-}{d_i^- + d_i^+}, \quad (30)$$

where $C_i \in [0, 1]$ for $i = 1, 2, \dots, m$. An alternative with a higher C_i is supposed to be a better solution and vice versa.

4.2. The Proposed Method. Figure 1 presents the flow chart of the proposed method. The detailed description of the proposed method is outlined as follows.

Step 1. Determine the weights of different similarity indices. In TOPSIS, each attribute is associated with a weight. In this work, we employ similarity indices as the multiattribute in

TOPSIS application for link prediction. Thence, we need to determine the weight of each index. To this end, we adjust the AUC for micro nodes to propose AUC_{v_i} for node v_i [44]. For the node v_i , let E_{v_i} denote the set of existing links between v_i and other nodes and \bar{E}_{v_i} represent the set of non-existing links between v_i and other nodes. Suppose I be a similarity index, the similarity scores of all links in E_{v_i} and \bar{E}_{v_i} are computed based on I , respectively. Then, the value of AUC_{v_i} is defined as

$$AUC_{v_i} = \frac{n'_{v_i} + 0.5n''_{v_i}}{n_{v_i}}, \quad (31)$$

where n_{v_i} is the total independent comparison time; n'_{v_i} denotes the comparison time that the link from E_{v_i} has a higher similarity score, and n''_{v_i} denotes the comparison time that has the same score.

To define the weight of similarity index I , we randomly select $p\%$ nodes from the network, and compute the value of AUC_{v_i} for each selected node. Let ρ be the group of selected nodes, we define

$$AUC_\rho = \begin{cases} \frac{\sum_{v_i \in \rho} AUC_{v_i}}{|\rho|}, & |\rho| > 0 \\ 0.5, & |\rho| = 0 \end{cases} \quad (32)$$

The weight of similarity index I is denoted as w_I , which is determined by the ratio of AUC_ρ and 0.5 (see (33)) [44]. 0.5 represents the result of random prediction.

$$w_I = \frac{AUC_\rho}{0.5}. \quad (33)$$

Algorithm 1 lists the procedure for determining the weight of a similarity index. In implementation, we randomly select 10% nodes from the networks.

Step 2. Calculate the similarity score for each potential missing link by different similarity indices.

As aforementioned, the indices of RA, LP, and CAR are used in LP_{TOPSIS} . The reason that we adopt these indices is that they are designed based on different but prominent structural features. Suppose that m unconnected node pairs are assigned similarity scores, and the decision matrix is represented as follows:

$$\mathbf{X} = \begin{bmatrix} RA_1 & LP_1 & CAR_1 \\ RA_2 & LP_2 & CAR_2 \\ \vdots & \vdots & \vdots \\ RA_m & LP_m & CAR_m \end{bmatrix}. \quad (34)$$

Step 3. Compute the normalized and weighted decision matrix. Since the values of different similarity indices are in different scales, we normalize matrix \mathbf{X} based on (22). Then, we

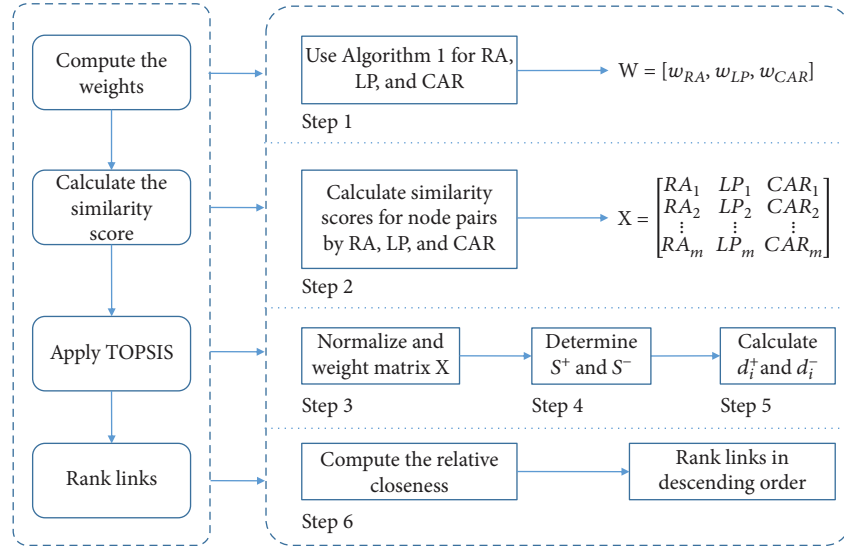


FIGURE 1: The flow chart of the proposed method.

Input: E_{tr} : training graph; I : similarity index; $p\%$: percentage of nodes
Output: weight of I
 1: $\rho \leftarrow$ randomly selected $p\%$ nodes from E_{tr} ;
 2: **for** $v_i \in \rho$ **do**
 3: Calculate similarity scores for existing links between v_i and other nodes based on I ;
 4: Calculate similarity scores for non-existing links between v_i and other nodes based on I ;
 5: Calculate $AUCv_i$ using Eq. (31)
 6: **end for**
 7: Calculate w_I using Eq. (33);
 8: **return** w_I ;

ALGORITHM 1: Determine weight.

compute the weighted normalized matrix using (23). The associated weights are obtained in Step 1.

Step 4. Determine the positive ideal and negative ideal solutions.

According to the weighted normalized matrix obtained in Step 2, the positive ideal solution S^+ and negative ideal solution S^- are determined using (24) and (25), respectively.

Step 5. Calculate the separation measures of potential missing links from the positive ideal and the negative ideal solutions. The separation measures between potential missing links and the positive ideal solution are calculated using (28), and the separation measures between potential missing links and the negative ideal solution are calculated using (29).

Step 6. Obtain the relative closeness of each potential missing link to the ideal solution.

The relative closeness is obtained by (30). Sort potential missing links in descending order according to their

closeness scores, and the links at the top are most likely to be missing ones.

4.3. Example Explanation. This section uses an example to explain how the proposed method works. Figure 2 shows the example network. In this example, we simply assume that each similarity index has the same weight. Thus, we have no need to weight the normalized matrix.

First, calculate the values of RA, LP, and CAR for unconnected node pairs of the network. Thirteen, twenty, and three unconnected node pairs are assigned values by RA, LP, and CAR, respectively. The results are listed in Figure 3(a). Because LP index considers both 2-hop and 3-hop paths, it gives values for the most unconnected node pairs. CAR index demands that there are links between the common neighbors. Since the example network is very simple, the requirement of CAR index cannot always be granted. Therefore, it assigns the score of zero to most node pairs.

Second, the decision matrix X and the normalized decision matrix Y are obtained according to Figure 3(a) and (22), which are presented in Figures 3(b) and 3(c), respectively.

Next, determine the positive ideal solution S^+ and the negative ideal solution S^- , which are

$$\begin{aligned} S^+ &= \{0.66699, 0.52204, 0.72761\}, \\ S^- &= \{0, 0.00017, 0\}. \end{aligned} \quad (35)$$

Then, compute d_i^+ and d_i^- for each potential missing link, respectively. The results are outlined in Table 1.

At last, calculate the relative closeness of all potential missing links and rank them based on their relative closeness. The results are shown in Table 1. As mentioned above, rank the potential missing links according to the relative closeness to the ideal solution and assume that the links at top ranks are the real missing ones.

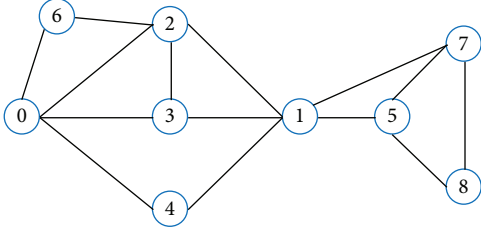


FIGURE 2: An example network with 8 nodes and 13 links.

5. Experimental Analysis

5.1. Datasets. To evaluate the performance of the proposed method, we use 10 real-world networks collected from various fields in this work (all data are downloaded from <http://deim.urv.cat/~alexandre.arenas/data/welcome.htm>, <http://www-personal.umich.edu/~mejn/netdata/>, <http://vlado.fmf.uni-lj.si/pub/networks/data/>, <http://noesis.ikor.org/datasets/link-prediction>). The brief descriptions of these networks are given as follows:

- (1) C. elegans (CE): the neural network of a Caenorhabditis elegans worm [45].
- (2) Email: a network of email interchanges between members of a university [46].
- (3) Facebook (FBK): a social network collected from <https://www.facebook.com/> [47].
- (4) Football (FTB): the network of American football games between Division IA colleges during regular season Fall 2000 [48].
- (5) HEP: the coauthorships network of scientists who posted preprints on the high-energy theory archive from 1995 to 1999 [49].
- (6) Infectious (INF): a network of people's face-to-face contacts in the exhibition "Infectious: Stay Away" in 2009 at the Science Gallery in Dublin [50].
- (7) Jazz: a network of Jazz musicians [51].
- (8) NetScience (NS): a network of coauthorships between scientists working on network theory and experiment [52].
- (9) Political blogs (PB): a network of blogs about US politics [53].
- (10) USAir: a network of the US air transportation system [17].

In this work, all networks are treated as undirected and unweighted networks, and only the giant component of each network is used. Table 2 lists the basic statistics of the giant components of these networks.

5.2. Friedman Test. To further analyze the statistical significance of the proposed method, the Friedman test [56] is introduced in our experiments. This test is a nonparametric

statistical hypothesis test used to compare multiple methods on a group of datasets [57]. It ranks the methods for each dataset separately according to their accuracy; the best performing method getting rank 1, the second best rank 2, and so on. In case of ties, average ranks are assigned.

Given k methods and N datasets, $r_{i,j}$ denotes the rank of the i th method on the j th dataset, and R_i is the average rank of the i th method, $R_i = 1/N \sum_j r_{i,j}$. The null-hypothesis in Friedman test is that all the methods are equivalent and then their ranks R_i should be equal. The Friedman statistic is

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_i R_i^2 - \frac{k(k+1)^2}{4} \right], \quad (36)$$

which is distributed according to χ_F^2 with $(k-1)$ degrees of freedom. Later, Iman and Davenport found that Friedman statistic is undesirably conservative and presented a better statistic [58], which is

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2}. \quad (37)$$

This statistic is distributed according to the F -distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. If F_F is greater than the critical value of $F(k-1, (k-1)(N-1))$, the null-hypothesis is rejected [57]. Consequently, there are significant differences between these approaches.

If the null-hypothesis is rejected, a post-hoc test is further proceeded to analyze the significant differences. The critical difference is defined as

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \quad (38)$$

where q_α is a critical value for post-hoc test [57]. If their average ranks differ by at least the critical difference, the performance of two approaches is significantly different.

5.3. Results and Analysis. Table 3 lists the predicted results of different methods under the AUC metric on the 10 networks. The numbers in the round brackets are the ranks. In case of ties, average ranks are assigned. These results are the average of 50 independent realizations for each network. In each realization, we randomly split a network into a training set and a testing set, which contain 90% and 10% links, respectively. The best value for each network is highlighted in boldface. It is evident from Table 3 that LP_{TOPSIS} achieves the best accuracy in terms of AUC on Email, FBK, FTB, INF, and NS and obtains the second best on CE and PB. These results are fairly decent. On HEP, LP outperforms others because this network possesses high average shortest distance, small average degree, and network efficiency. In other words, the network of HEP is very sparse (see Table 2), so the local methods, such as RA, CAR, and CCLP, obtain lower accuracy than LP. As stated in [23], in the relatively sparse network, common neighbor-based methods are less distinguishable. Whereas, the additional information provided by the 3-hop

Node pair	RA	LP	CAR
(0, 1)	1.083	3.003	3
(0, 5)	0	0.003	0
(0, 7)	0	0.003	0
(1, 6)	0.25	1.004	0
(1, 8)	0.667	2.002	2
(2, 4)	0.45	2.003	0
(2, 5)	0.2	1.002	0
(2, 7)	0.2	1.002	0
(2, 8)	0	0.002	0
(3, 4)	0.45	2.002	0
(3, 5)	0.2	1.002	0
(3, 6)	0.5	2.003	2
(3, 7)	0.2	1.002	0
(3, 8)	0	0.002	0
(4, 5)	0.2	1.001	0
(4, 6)	0.25	1.002	0
(4, 7)	0.2	1.001	0
(4, 8)	0	0.002	0
(5, 6)	0	0.001	0
(6, 7)	0	0.001	0

Node pair	RA	LP	CAR
(0, 1)	1.083	3.003	3
(0, 5)	0	0.003	0
(0, 7)	0	0.003	0
(1, 6)	0.25	1.004	0
(1, 8)	0.667	2.002	2
(2, 4)	0.45	2.003	0
(2, 5)	0.2	1.002	0
(2, 7)	0.2	1.002	0
(2, 8)	0	0.002	0
(3, 4)	0.45	2.002	0
(3, 5)	0.2	1.002	0
(3, 6)	0.5	2.003	2
(3, 7)	0.2	1.002	0
(3, 8)	0	0.002	0
(4, 5)	0.2	1.001	0
(4, 6)	0.25	1.002	0
(4, 7)	0.2	1.001	0
(4, 8)	0	0.002	0
(5, 6)	0	0.001	0
(6, 7)	0	0.001	0

Node pair	RA	LP	CAR
(0, 1)	0.66699	0.52204	0.72761
(0, 5)	0	0.00052	0
(0, 7)	0	0.00052	0
(1, 6)	0.15392	0.17454	0
(1, 8)	0.41046	0.34803	0.48507
(2, 4)	0.27706	0.34820	0
(2, 5)	0.12314	0.17419	0
(2, 7)	0.12314	0.17419	0
(2, 8)	0	0.00035	0
(3, 4)	0.27706	0.34803	0
(3, 5)	0.12314	0.17419	0
(3, 6)	0.30784	0.34820	0.48507
(3, 7)	0.12314	0.17419	0
(3, 8)	0	0.00035	0
(4, 5)	0.12314	0.17401	0
(4, 6)	0.15392	0.17419	0
(4, 7)	0.12314	0.17401	0
(4, 8)	0	0.00035	0
(5, 6)	0	0.00017	0
(6, 7)	0	0.00017	0

(a) Similarity scores

(b) Decision matrix

(c) Normalized decision matrix

FIGURE 3: The values of RA, LP, and CAR for unconnected node pairs in the example network.

TABLE 1: Relative closeness and rank of each node pair.

Node pair	d_i^+	d_i^-	C_i	Rank
(0, 1)	0	1.11653	1	1
(1, 8)	0.39359	0.72441	0.64795	2
(3, 6)	0.46694	0.67170	0.58992	3
(2, 4)	0.84361	0.44484	0.34525	4
(3, 4)	0.84365	0.44471	0.34517	5
(1, 6)	0.95573	0.23258	0.19572	6
(4, 6)	0.95585	0.23232	0.19553	7
(2, 5)	0.97272	0.21318	0.17976	8
(2, 7)	0.97272	0.21318	0.17976	9
(3, 5)	0.97272	0.21318	0.17976	10
(3, 7)	0.97272	0.21318	0.17976	11
(4, 5)	0.97279	0.21303	0.17965	12
(4, 7)	0.97279	0.21303	0.17965	13
(0, 5)	1.11637	0.00035	0.00031	14
(0, 7)	1.11637	0.00035	0.00031	15
(2, 8)	1.11645	0.00017	0.00016	16
(3, 8)	1.11645	0.00017	0.00016	17
(4, 8)	1.11645	0.00017	0.00016	18
(5, 6)	1.11653	0	0	19
(6, 7)	1.11653	0	0	20

paths introduced by LP can make the similarities much more distinguishable and thus enhances the accuracy. Although $L_{P_{TOPSIS}}$ integrates the advantage of LP index, the CAR index, which is not suitable for very sparse networks, weakens its performance. On Jazz and USAir, the proposed method gets rank 4 and 3, respectively. That is resulted from the poor accuracy of LP index on both networks. In this paper, we uniformly set the parameter of LP index as 0.001; however, the optimal value of the parameter is varying on different networks [21, 23], and detecting the optimal value is very

time-consuming. In addition, LR_m manifests fairly good predicted results, which obtains three best and three second. Generally speaking, LP_{TOPSIS} performs the best and LR_m does the second. The reason is that both methods aggregate several structural features. However, other baselines do not obtain satisfactory results. Take LP index as an example, it is ranked first on HEP but ranked seventh on USAir and eighth on Jazz. In a nutshell, the proposed method is more stable on different networks than baselines. For the MaxE method, which uncovers the missing links by reconstructing the network at hand based on the exponential random graph framework, it performs relative poorly on the networks with well-defined community, such as FBK, FTB, and Email. The reason is the configuration model used in MaxE to reconstruct the network may not be the optimal generative model for those networks. Actually, determining the optimal generative model is very difficult. Nevertheless, on the network with core-periphery structure, for example, PB and USAir, MaxE achieves good accuracy under the metric of AUC. The corresponding values are 0.9009 and 0.8887. These values are acceptable. The configuration model of MaxE takes as input just the nodes degrees; networks with core-periphery structure seem to be largely explained by the degree sequences [28]. Thus, for networks with a core-periphery structure, the MaxE is suitable.

Next, we applied the Friedman test [57] on the above results to analyze the significant differences between baselines and LP_{TOPSIS} . Depending on Table 3, we get $\chi_F^2 = 72.115$ and $F_F = 36.288$. In this paper, there are 10 methods and 10 networks. F_F is distributed according to the F -distribution with $10 - 1 = 9$ and $(10 - 1) \times (10 - 1) = 81$ degrees of freedom. The critical value of $F(9, 81)$ for $\alpha = 0.05$ is 1.998. Due to $F_F = 36.288 > 1.998$, we reject the null-hypothesis, which states that all the methods are equivalent.

Since the null-hypothesis is rejected, we proceed with a post-hoc test. In our experiments, the Bonferroni-Dunn test [59], in which all methods are compared only to a

TABLE 2: The basic statistics of the giant components of the 10 networks. $|V|$ and $|E|$ are the total numbers of nodes and edges, respectively. $\langle k \rangle$ and $\langle d \rangle$ present the average degree and the average shortest distance, respectively. C and r indicate the clustering coefficient [45] and assortative coefficient [54], respectively. H is the degree heterogeneity [17], and e is the network efficiency [55].

Networks	$ V $	$ E $	$\langle k \rangle$	$\langle d \rangle$	C	r	H	e
CE	297	2148	14.465	2.455	0.292	-0.163	1.801	0.445
Email	1133	5451	9.622	3.606	0.220	0.078	1.942	0.300
FBK	4015	87,882	43.777	3.985	0.595	0.071	2.427	0.294
FTB	115	613	10.661	2.508	0.403	0.162	1.007	0.450
HEP	5835	13,815	4.735	7.026	0.506	0.185	1.926	0.155
INF	410	2765	13.488	3.631	0.456	0.226	1.388	0.323
Jazz	198	2742	27.697	2.235	0.617	0.020	1.395	0.513
NS	379	914	4.823	6.042	0.741	-0.082	1.663	0.203
PB	1222	16,714	27.355	2.738	0.320	-0.221	2.971	0.398
USAir	332	2126	12.807	2.738	0.625	-0.208	3.464	0.406

TABLE 3: AUC values of different methods on 10 networks. The results are the average of 50 independent implementations with $|E_{ts}|/|E| = 0.1$. The best performance for each network is emphasized by boldface. The numbers in the round brackets are the ranks. In case of ties, average ranks are assigned.

	RA	ADP	CAR	CCLP	LP	MI	MaxE	LR _m	LNB_CN	LP _{TOPSIS}
CE	0.8680 (3)	0.8676 (4)	0.7650 (9)	0.8651 (5)	0.8628 (6)	0.8321 (8)	0.7531 (10)	0.8828 (1)	0.8612 (7)	0.8784 (2)
Email	0.8461 (6)	0.8465 (5)	0.6968 (10)	0.8421 (8)	0.9005 (2)	0.8514 (4)	0.7814 (9)	0.8957 (3)	0.8458 (7)	0.9078 (1)
FBK	0.9944 (2)	0.9943 (3)	0.9842 (9)	0.9921 (7)	0.9923 (5)	0.9891 (8)	0.8312 (10)	0.9935 (4)	0.9923 (6)	0.9954 (1)
FTB	0.8464 (4.5)	0.8464 (4.5)	0.8146 (8)	0.8420 (7)	0.8613 (2)	0.7967 (9)	0.2318 (10)	0.8531 (3)	0.8425 (6)	0.8636 (1)
HEP	0.8995 (4.5)	0.8995 (4.5)	0.7068 (9)	0.8628 (7)	0.9160 (1)	0.8043 (8)	0.6647 (10)	0.9121 (2)	0.8993 (6)	0.9108 (3)
INF	0.9444 (4.5)	0.9444 (4.5)	0.8622 (9)	0.9379 (7)	0.9575 (3)	0.9159 (8)	0.7105 (10)	0.9604 (2)	0.9402 (6)	0.9621 (1)
Jazz	0.9699 (3)	0.9711 (2)	0.9528 (7)	0.9579 (6)	0.9499 (8)	0.9455 (9)	0.7688 (10)	0.9723 (1)	0.9584 (5)	0.9667 (4)
NS	0.9588 (3)	0.9587 (4)	0.8154 (9)	0.9287 (7)	0.9576 (5)	0.8509 (8)	0.6345 (10)	0.9597 (2)	0.9571 (6)	0.9625 (1)
PB	0.9230 (6)	0.9232 (5)	0.8921 (10)	0.9209 (8)	0.9289 (3)	0.9238 (4)	0.9009 (9)	0.9383 (1)	0.9213 (7)	0.9321 (2)
USAir	0.9516 (2)	0.9517 (1)	0.9127 (8)	0.9381 (6)	0.9268 (7)	0.9122 (9)	0.8887 (10)	0.9454 (4)	0.9404 (5)	0.9475 (3)

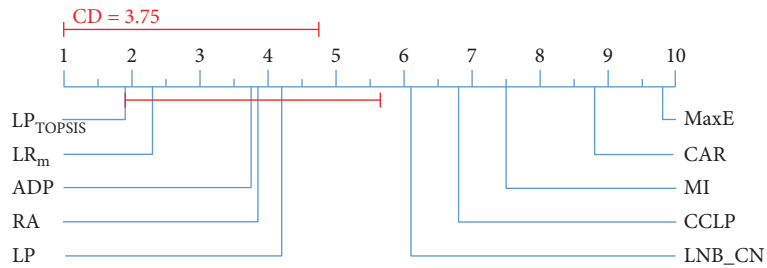


FIGURE 4: Comparison of LP_{TOPSIS} against the others with the Bonferroni-Dunn test. This comparison is based on the results in Table 3. All methods with ranks outside the marked interval are significantly different from LP_{TOPSIS}.

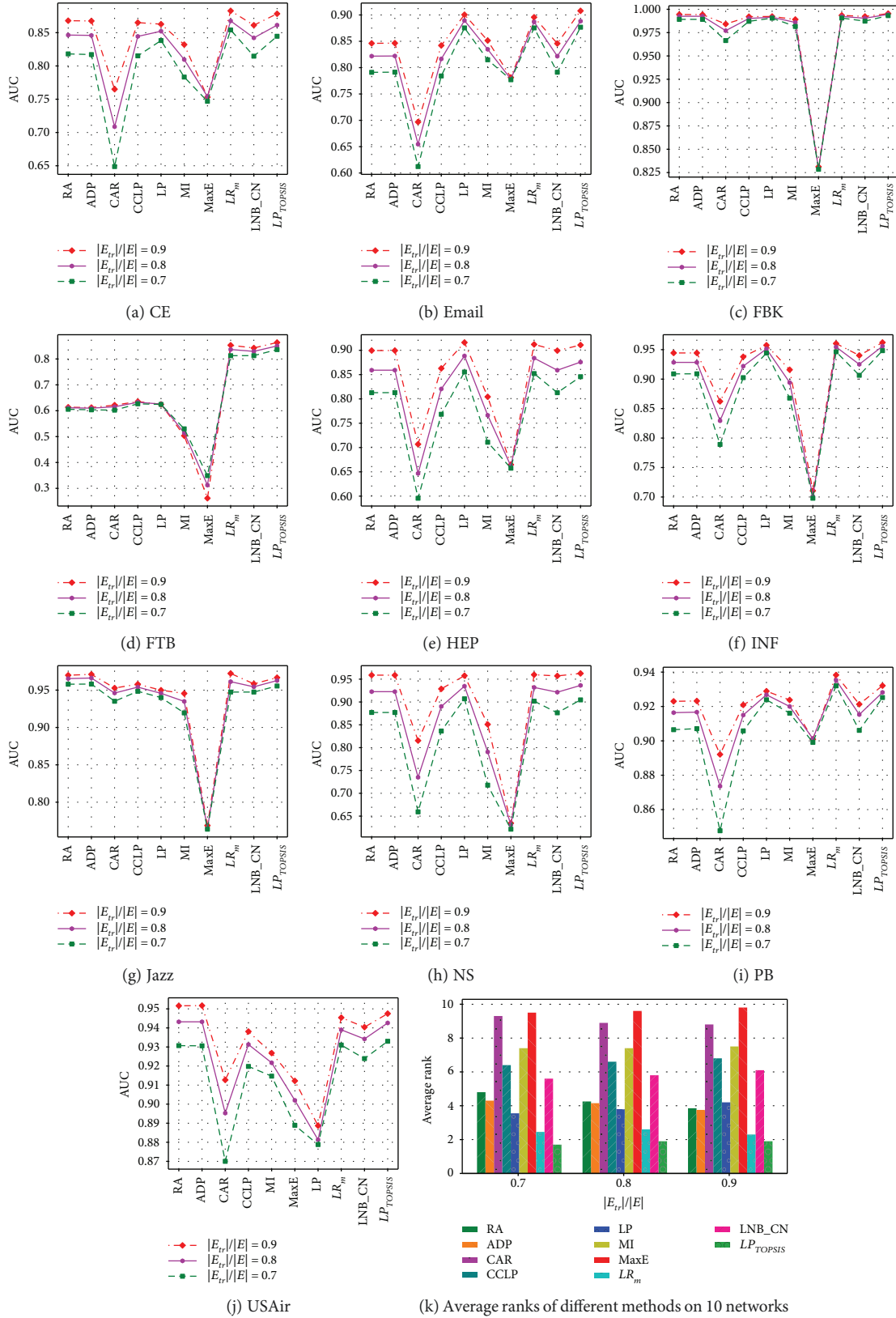


FIGURE 5: AUC results on 10 networks with different proportions of training set E_{tr} . The results are the average of 50 independent implementations.

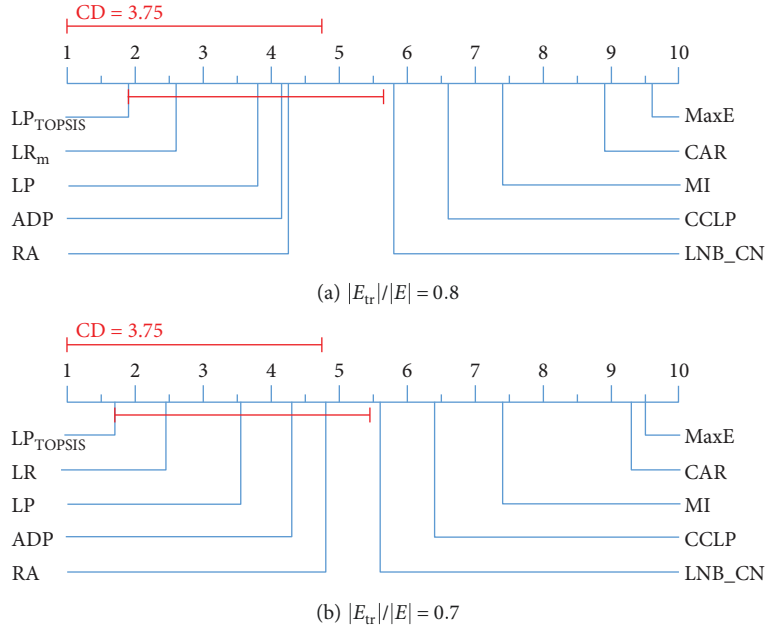


FIGURE 6: The Bonferroni-Dunn test for $|E_{tr}|/|E| = 0.8$ and 0.7 . All methods with ranks outside the marked interval are significantly different from LP_{TOPSIS} .

control method and not between themselves [57], is employed to estimate the significant differences between LP_{TOPSIS} and baselines. The critical difference is $CD = 2.773 \times \sqrt{(10 \times (10 + 1)) / (6 \times 10)} = 3.75$ for $\alpha = 0.05$. The results are graphically presented in Figure 4. In the axis, the best rank is on the left side. Figure 4 shows that LP_{TOPSIS} is significantly better than LNB_CN, CCLP, MI, CAR, and MaxE. Although there are no significant differences of LR_m, ADP, RA, and LP with LP_{TOPSIS} , the average rank of LP_{TOPSIS} is better than them.

Furthermore, Figure 5 describes the changes of AUC of all prediction methods for varying proportions of training set E_{tr} in E (from 0.7 to 0.9). Clearly, AUC scores show an upward trend when the proportion increase from 0.7 to 0.9 in Figure 5. The reason is event; the more proportion of training set E_{tr} is, the more information is provided for training. On the contrary, low proportions of E_{tr} will enhance the difficulty of link prediction. Therefore, we do not conduct experiment with lower proportions of E_{tr} , for example, 0.6 and 0.5. According to Figure 5, the AUC values of the proposed method, with varying sizes of training set, are either the best or close to the best. Figure 5(k) exhibits the average ranks of different methods for varying proportions of E_{tr} . It can be observed that the proposed method always gets the best. The significant analysis for $|E_{tr}|/|E| = 0.9$ (that is, $|E_{ts}|/|E| = 0.1$) is already presented above. Now we give the analysis for $|E_{tr}|/|E| = 0.8$ and 0.7 , respectively. The corresponding values of F_F are 25.415 and 31.898, respectively. Both values are greater than 1.998 (the critical value of $F(9, 81)$). That means the null-hypothesis, which states that all these methods are equivalent, is rejected. The results of Bonferroni-Dunn test [59] are depicted in Figure 6. From Figure 6, LP_{TOPSIS} ranks first and is significantly better than

LNB_CN, CCLP, MI, CAR, and MaxE. These results are similar with that in Figure 4.

On the other hand, the metric of *Precision* focuses on the top- L predicted links. Figure 7 shows the prediction results under the metric of *Precision* on the 10 networks with different sizes of L . These results demonstrate that LP_{TOPSIS} is invariably in the top place over most networks. However, baselines have wild fluctuations on different networks. For instance, RA index achieves the best on NS, but falls to the second last on PB. In addition, in most methods, *Precision* shows slightly downward trend when the size of L increases. The reason is that the increase of L , the probability to uncover relevant items will decrease, and then the value of *Precision* will lower. Figure 7(k) presents the average rank of each method over the 10 networks with respect to different sizes of L . Clearly, LP_{TOPSIS} outperforms others except when $L = 10$ and 20 . MI index ranks first when $L = 10$ and 20 but falls behind LP_{TOPSIS} for other values of L . MI index is superior to other baselines under *Precision*; however, its AUC values are not satisfactory enough. On the whole, LR_m performs mediocre in terms of *Precision*, in spite of its quite good performance under the metric of AUC .

At last, we depict the changes of *Precision* of all prediction methods for varying proportions of training set E_{tr} in E (from 0.7 to 0.9) in Figure 8. In this experiment, we set $L = |E_{ts}|$ for all networks. It can be seen from Figure 8 that *Precision* exhibits the opposite changing trend in comparison with AUC , that is, *Precision* scores show a downward trend when the proportion increase from 0.7 to 0.9. This phenomenon has also been observed in [5]. The main reason is that the decrease of training set E_{tr} will lead to weak n_1 and strong n_2 in the definition of AUC (see (1)) and then lower the value of AUC [5]. Oppositely, the increase of

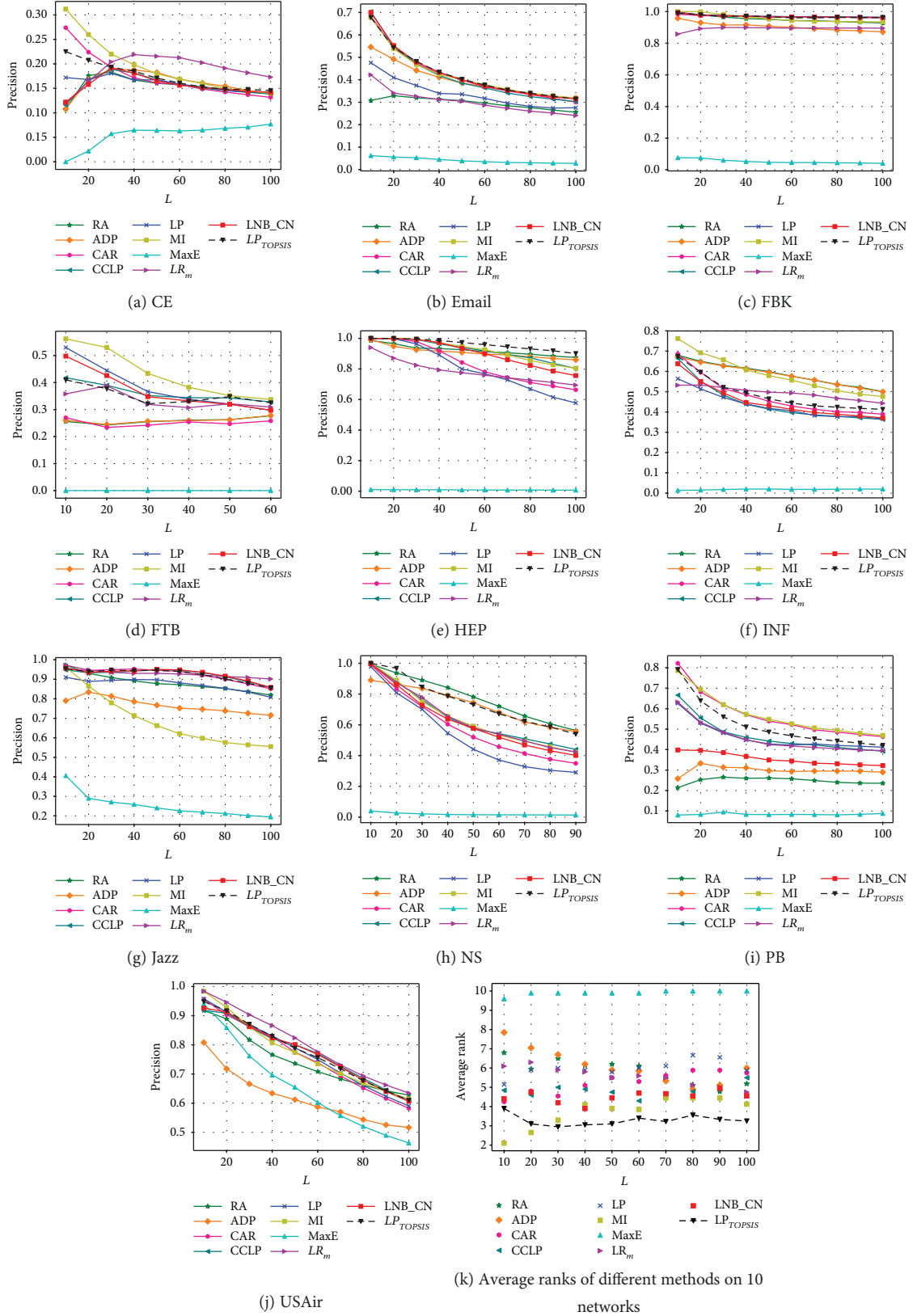


FIGURE 7: Precision results on 10 networks with different values of L . The results are the average of 50 independent implementations with $|E_{ts}|/|E| = 0.1$. The size of E_{ts} for FTB is 61, so the max L selected is 60. Similarly, the max L selected for NS is 90.

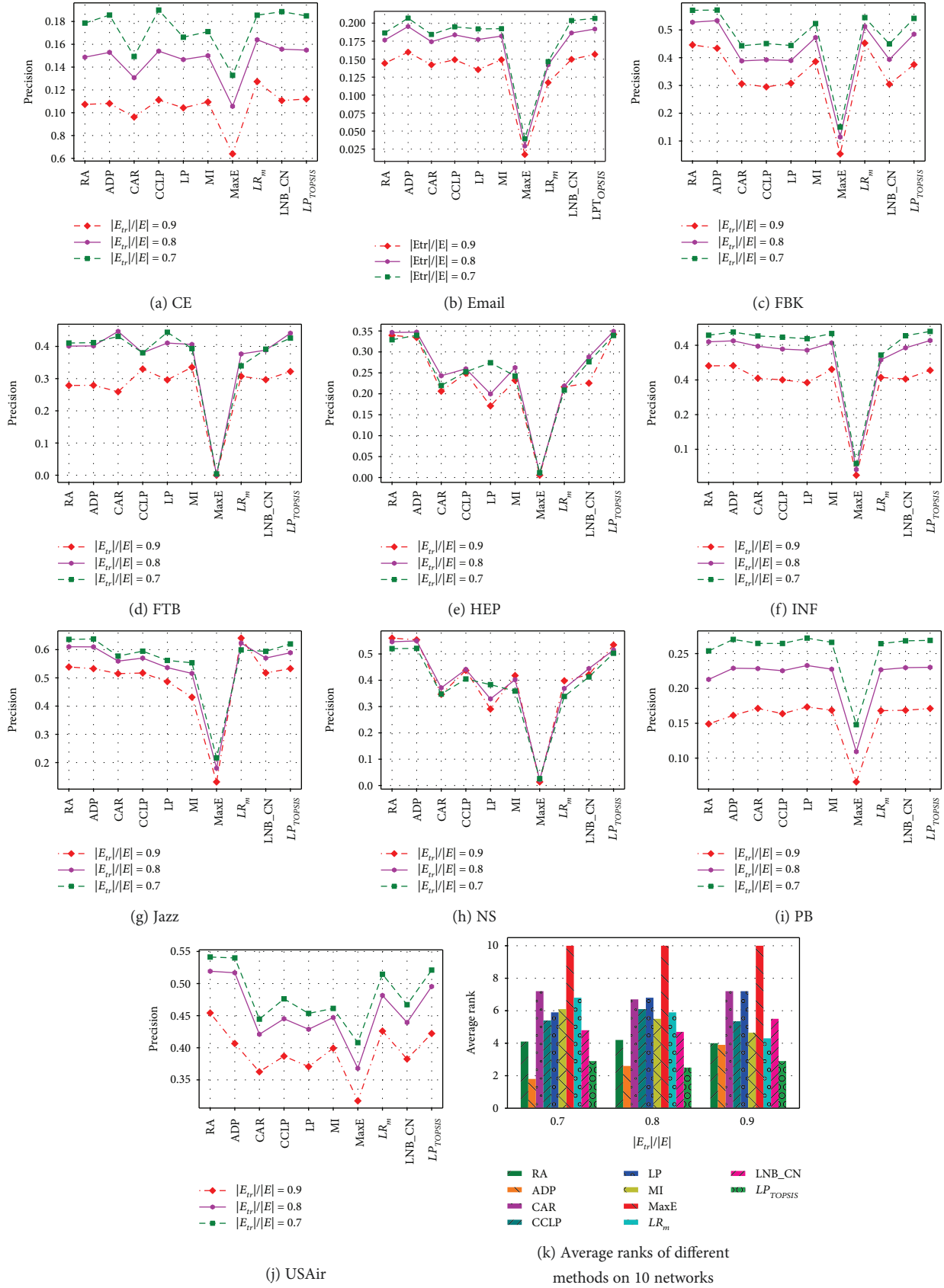


FIGURE 8: Precision results on 10 networks with different proportions of training set. The results are the average of 50 independent implementations. In this figure, $L = |E_{ts}|$ for all networks.

testing set E_{ts} , the probability to obtain the relevant items will increase, which makes it easier for uncovering the missing links [5]. Therefore, combining both metrics in the evaluation of the accuracy of a prediction method is necessary in practical application. Figure 8(k) shows the average ranks of different methods for $|E_{tr}|/|E| = 0.7, 0.8, \text{ and } 0.9$ in terms of *Precision*. In general, the proposed method is in the second place when $|E_{tr}|/|E| = 0.7$ and ranks first when $|E_{tr}|/|E| = 0.8$ and 0.9 . This indicates the stable performance of the proposed method under the metric of *Precision*.

From the aforementioned results, we can conclude that the proposed method outperforms the compared indices and is applicable to more networks. The striking characteristic of the proposed method is that it aggregates several structural features of a network by means of combining RA, LP, and CAR indices using TOPSIS. Thus, the proposed method can automatically adapt to diverse networks and then performs stability on various networks. Although the ADP index is claimed to be able to automatically adapt to the structure of a network by adaptively penalizing the degrees of common neighbors, it still focuses, in effect, on one structure in a network. Therefore, there are gaps between the accuracy of ADP and LP_{TOPSIS} . Similar with LP_{TOPSIS} , LR_m aggregates three structural features with logistic regression. This method achieves the second best under *AUC*, whereas its accuracy under *Precision* is very general.

6. Conclusion

Link prediction aims at finding the missing links and predicting future links in a network. As an important research topic in complex network analysis, it has drawn increasing attention from disparate scientific communities. Among various categories of approaches, similarity-based methods have become the mainstream due to their low complexity and high interpretability. In general, one similarity-based method assumes that its similarity measure is applicable to diverse networks. However, different networks always have different inner topological structures, which results in the unstable performance of similarity-based methods. Inspired by the applications of multiple-attribute decision-making (MADM) problem, we proposed in this paper a novel link prediction method. The proposed method employs three classical similarity indices, that is, RA, LP, and CAR, as the attributes and aggregates their scores by means of TOPSIS, a well-known MADM method, to make a decision for ranking unconnected node pairs.

The accuracy of the proposed method is experimentally evaluated over 10 real-world networks with the metrics of *AUC* and *Precision*. The experimental results indicate that the proposed method is not only more effective but also stable than the competing methods. The robustness comparisons of the proposed method with baselines for varying sizes of training sets suggest the robustness of the proposed method. The results in our experiments demonstrate that MADM method is an effective way to solve the link prediction problem. In the future work, we can further study the application of MADM method in link prediction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We would like to thank the anonymous reviewers for their contributions and the authors of LR_m and MaxE for sharing their codes. This work was supported by the National Natural Science Foundation of China (nos. 61602225, 61762077) and the Fundamental Research Funds for the Central Universities (no. lzujbky-2017-192).

References

- [1] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.
- [2] E. Sprinzak, S. Sattath, and H. Margalit, "How reliable are experimental protein–protein interaction data?," *Journal of Molecular Biology*, vol. 327, no. 5, pp. 919–923, 2003.
- [3] M. P. H. Stumpf, T. Thorne, E. de Silva et al., "Estimating the size of the human interactome," *Proceedings of the National Academy of Sciences*, vol. 105, no. 19, pp. 6959–6964, 2008.
- [4] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Computing Surveys*, vol. 49, no. 4, pp. 1–33, 2017.
- [5] J. Yang and X.-D. Zhang, "Predicting missing links in complex networks based on common neighbors and distance," *Scientific Reports*, vol. 6, no. 1, article 38208, 2016.
- [6] Q.-M. Zhang, L. Lü, W. Q. Wang, Yu-Xiao, and T. Zhou, "Potential theory for directed networks," *PLoS One*, vol. 8, no. 2, article e55437, 2013.
- [7] Q.-M. Zhang, X.-K. Xu, Y.-X. Zhu, and T. Zhou, "Measuring multiple evolution mechanisms of complex networks," *Scientific Reports*, vol. 5, no. 1, article 10350, 2015.
- [8] B. Bringmann, M. Berlingerio, F. Bonchi, and A. Gionis, "Learning and predicting the evolution of social networks," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 26–32, 2010.
- [9] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22073–22078, 2009.
- [10] S. S. Bhowmick and B. S. Seah, "Clustering and summarizing protein-protein interaction networks: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 638–658, 2016.
- [11] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici, "Link prediction in social networks using computationally efficient topological features," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 73–80, Boston, MA, USA, 2012.
- [12] C. Ma, T. Zhou, and H. F. Zhang, "Playing the role of weak clique property in link prediction: a friend recommendation model," *Scientific Reports*, vol. 6, no. 1, article 30098, 2016.

- [13] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [14] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries - JCDL '05*, pp. 141–142, Denver, CO, USA, 2005, ACM.
- [15] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks," in *Proceedings of the 2nd International Conference on Finding Experts on the Web with Semantics (FEWS-2007)*, vol. 290, pp. 42–55, Busan, Republic Korea, 2007.
- [16] Q. Yu, C. Long, Y. Lv, H. Shao, P. He, and Z. Duan, "Predicting co-author relationship in medical co-authorship networks," *PLoS One*, vol. 9, article e101214, 7 pages, 2014.
- [17] L. Lü and T. Zhou, "Link prediction in complex networks: a survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.
- [18] L. Pan, T. Zhou, L. Lü, and C.-K. Hu, "Predicting missing links and identifying spurious links via likelihood analysis," *Scientific Reports*, vol. 6, no. 1, article 22955, 2016.
- [19] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [20] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
- [21] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.
- [22] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.
- [23] L. Lü, C. H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Physical Review E*, vol. 80, no. 4, article 046122, 2009.
- [24] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, "From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks," *Scientific Reports*, vol. 3, no. 1, p. 1613, 2013.
- [25] Z. Wu, Y. Lin, J. Wang, and S. Gregory, "Link prediction with node clustering coefficient," *Physica A: Statistical Mechanics and its Applications*, vol. 452, pp. 1–8, 2016.
- [26] C. Ma, Z.-K. Bao, and H.-F. Zhang, "Improving link prediction in complex networks by adaptively exploiting multiple structural features of networks," *Physics Letters A*, vol. 381, no. 39, pp. 3369–3376, 2017.
- [27] V. Martínez, F. Berzal, and J. C. Cubero, "Adaptive degree penalization for link prediction," *Journal of Computational Science*, vol. 13, pp. 1–9, 2016.
- [28] F. Parisi, G. Caldarelli, and T. Squartini, "Entropy-based approach to missing-links prediction," *Applied Network Science*, vol. 3, no. 1, p. 17, 2018.
- [29] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.
- [30] R. V. Rao, "Introduction to multiple attribute decision-making (MADM) methods," in *Decision Making in the Manufacturing Environment*, pp. 27–41, Springer, London, UK, 2007.
- [31] M. Behzadian, S. Khanmohammadi Otaghsara, M. Yazdani, and J. Ignatius, "A state-of-the-art survey of TOPSIS applications," *Expert Systems with Applications*, vol. 39, no. 17, pp. 13051–13069, 2012.
- [32] S. Galam and J.-D. Zucker, "From individual choice to group decision-making," *Physica A: Statistical Mechanics and its Applications*, vol. 287, no. 3–4, pp. 644–659, 2000.
- [33] Y. Deng, F. T. S. Chan, Y. Wu, and D. Wang, "A new linguistic MCDM method based on multiple-criterion data fusion," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6985–6993, 2011.
- [34] Y. Du, C. Gao, Y. Hu, S. Mahadevan, and Y. Deng, "A new method of identifying influential nodes in complex networks based on TOPSIS," *Physica A: Statistical Mechanics and its Applications*, vol. 399, pp. 57–69, 2014.
- [35] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer, Berlin, Heidelberg, 1981.
- [36] S.-J. Chen and C.-L. Hwang, "Fuzzy multiple attribute decision making methods," in *Fuzzy Multiple Attribute Decision Making*, pp. 289–486, Springer, Berlin, Heidelberg, 1992.
- [37] J. Hu, Y. Du, H. Mo, D. Wei, and Y. Deng, "A modified weighted TOPSIS to identify influential nodes in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 444, pp. 73–85, 2016.
- [38] L. Fei and Y. Deng, "A new method to identify influential nodes based on relative entropy," *Chaos, Solitons & Fractals*, vol. 104, pp. 257–267, 2017.
- [39] M. Zhou, R. Liu, X. Zhao, and Q. Zhu, "Link prediction algorithm based on local centrality of common neighbor nodes using multi-attribute ranking," in *2017 12th International Conference on Computer Science and Education (ICCSE)*, pp. 506–511, Houston, TX, USA, 2017.
- [40] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
- [41] F. Tan, Y. Xia, and B. Zhu, "Link prediction in complex networks: a mutual information perspective," *PLoS One*, vol. 9, no. 9, article e107056, 2014.
- [42] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks: a local naive bayes model," *EPL (Europhysics Letters)*, vol. 96, no. 4, article 48007, 2011.
- [43] T. Squartini and D. Garlaschelli, "Analytical maximum-likelihood method to detect patterns in real networks," *New Journal of Physics*, vol. 13, no. 8, article 083001, 2011.
- [44] W. Hu, H. Wang, C. Peng, H. Liang, and B. Du, "An event detection method for social networks based on link prediction," *Information Systems*, vol. 71, pp. 16–26, 2017.
- [45] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [46] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E*, vol. 68, no. 6, article 065103, 2003.
- [47] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - NIPS'12*, vol. 1, pp. 539–547, Lake Tahoe, NV, USA, 2012, Curran Associates Inc..

- [48] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [49] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.
- [50] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J. F. Pinton, and W. van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.
- [51] P. M. Gleiser and L. Danon, "Community structure in jazz," *Advances in Complex Systems*, vol. 6, no. 4, pp. 565–573, 2003.
- [52] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, article 036104, 2006.
- [53] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery - LinkKDD '05*, pp. 36–43, Chicago, IL, USA, 2005.
- [54] M. E. J. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, article 026126, 2003.
- [55] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Physical Review Letters*, vol. 87, no. 19, article 198701, 2001.
- [56] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.
- [57] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [58] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Communications in Statistics - Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.
- [59] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.



Hindawi

Submit your manuscripts at
www.hindawi.com

