iScience

CellPress
OPEN ACCESS

Review

Progresses and challenges in link prediction

Tao Zhou[1,*]

SUMMARY

Link prediction is a paradigmatic problem in network science, which aims at estimating the existence likelihoods of nonobserved links, based on known topology. After a brief introduction of the standard problem and evaluation metrics of link prediction, this review will summarize representative progresses about local similarity indices, link predictability, network embedding, matrix completion, ensemble learning, and some others, mainly extracted from related publications in the last decade. Finally, this review will outline some long-standing challenges for future studies.

INTRODUCTION

Network is a natural and powerful tool to characterize a huge number of social, biological, and information systems that consist of interacting elements, and network science is currently one of the most active interdisciplinary research domains (Barabasi, 2016; Newman, 2018). Link prediction is a paradigmatic problem in network science that attempts to uncover missing links or predict future links (Lü and Zhou, 2011), which has already found many theoretical and practical applications, such as reconstruction of networks (Squartini et al., 2018; Peixoto, 2018), evaluation of evolving models (Wang et al., 2012; Zhang et al., 2015), inference of biological interactions (Csermely et al., 2013; Ding et al., 2014), online recommendation of friends and products (Aiello et al., 2012b; Lü et al., 2012), and so on.

Thanks to a few pioneering works (Liben-Nowell and Kleinberg, 2007; Clauset et al., 2008; Zhou et al., 2009; Guimerà and Sales-Pardo, 2009), link prediction has been one of the most active research domains in network science. Early contributions were already summarized by a well-known survey article (Lü and Zhou, 2011), and this review will first define the standard problem and discuss some well-known evaluation metrics and then introduce most representative achievements in the last decade (mostly published after (Lü and Zhou, 2011)), including local similarity indices, link predictability, network embedding, matrix completion, ensemble learning, and some others. Lastly, this review will show limitations of existing studies as well as open challenges for future studies.

EVALUATION

Consider a simple network $G(V,E)$, where V and E are sets of nodes and links, the directionalities and weights of links are ignored, and multiple links and self-connections are not allowed. We assume that there are some missing links or future links in the set of nonobserved links $U \setminus E$, where U is the universal set containing all $|V|(|V|-1)/2$ potential links. The task of link prediction is to find out those missing or future links. To test the algorithm's accuracy, the observed link, E, is divided into two parts: the training set E^T is treated as known information, while the probe set E^P is used for algorithm evaluation, and no information in E^P is allowed to be used for prediction. The majority of known studies applied "random division", namely E^P is randomly drawn from E. In the case of predicting future links, "temporal division" is usually adopted where E^P contains most recently appeared links (Lichtenwalter et al., 2010). In some real networks, missing links have different topological features from observed links. For example, missing links are more likely to be associated with low-degree nodes since interactions between hubs are easy to be known. In such situations, we may apply "biased division" to ensure that E^P is consisted of links with similar topological features to missing links (Zhu et al., 2012).

Performance evaluation metrics can be roughly divided into two categories: threshold-dependent metrics (e.g., fixed threshold accuracy) and threshold-independent metric (e.g., area under threshold curve). Precision and recall are the two most widely used metrics in the former category. Precision is defined as the ratio of relevant items selected to the number of items selected. That is to say, if we take the top-L links as the predicted ones,

[1]CompleX Lab, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China

*Correspondence: zhutou@ustc.edu

https://doi.org/10.1016/j.isci.2021.103217

among which L_r links are correctly predicted; then, the Precision equals L_r/L. Recall is defined as the ratio of relevant items selected to the total number of relevant items, say $L_r/|E^P|$. An obvious drawback of threshold-dependent metrics is that we generally do not have a reasonable way to determine the threshold, like the number of predicted links L or the threshold score for the existence of links. A widely adopted way is setting $L = |E^P|$, at which precision = recall (Lü and Zhou, 2011; Liben-Nowell and Kleinberg, 2007). Although $|E^P|$ is generally unknown, an experiential and reasonable setting is $|E^P| = 0.1|E|$ because 10% of links in the probe set are usually enough for us to get statistical solid results while the removal of 10% of links will probably not destroy the structural features of the target network (Lü et al., 2015).

Some studies argued that a single value might not well reflect the performance of a predictor (Lichtenwalter et al., 2010; Yang et al., 2015). Therefore, robust evaluation based on threshold-dependent metrics should cover a range of thresholds (e.g., by varying L), which is actually close to the consideration of threshold curves. The precision–recall (PR) curve (Yang et al., 2015) and receiver operating characteristic (ROC) curve (Hanely and McNeil, 1982) are frequently considered in the literature. The former shows precision with respect to recall at all thresholds and the latter represents performance trade-off between true positives and false positives at different thresholds. We say algorithm X is strictly better than algorithm Y only if X's threshold curve completely dominates Y's curve (mathematically speaking, a curve A dominates another curve B in a space is B is always equal or below curve A (Provost et al., 1998)). Davis and Goadrich (2006) have proved that a curve dominates in ROC space if and only if it dominates in PR space.

As the condition of domination is too rigid, we usually use areas under threshold curves as evaluation metrics. The area under the ROC curve (AUC) can be interpreted as the probability that a randomly chosen link in E^P is assigned a higher existence likelihood than a randomly chosen link in $U\ E$. If all likelihoods are generated from an independent and identical distribution, the AUC value should be about 0.5. Therefore, the degree to which the value exceeds 0.5 indicates how better the algorithm performs than pure chance. Thus far, AUC is the most frequently used metric in link prediction, probably because it is highly interpretable, easy to be interpolated, and of good visualization. Meanwhile, readers should be aware of some remarkable disadvantages of AUC, for example, AUC is inadequate to evaluate the early retrieval performance which is critical in real applications especially for many biological scenarios (Saito and Rehmsmeier, 2015), and AUC will give misleadingly overhigh score to algorithms that can successfully rank many negatives in the bottom while this ability is less significant in imbalanced learning (Yang et al., 2015; Lichtenwalter and Chawla, 2012). A typical viewpoint in the early studies is that AUC is suitable for imbalanced learning because AUC is not sensitive to the ratio of positives to negatives and thus can reflect an algorithm's ability that is independent to the data distribution (Fawcett, 2006). However, recent perspective in machine learning and big data is that talking about the performance or ability of a classification algorithm without specified datasets is meaningless so that this previous advantage gets increasing criticisms (Yang et al., 2015; Lichtenwalter and Chawla, 2012). Hand argued that AUC is fundamentally incoherent because AUC uses different misclassification cost distributions for different classifiers (Hand 2009). Hand's criticism is deep, but AUC indeed measures relative ranks instead of absolute loss and is irrelevant to misclassification cost so that to dissect AUC in the narration involving misclassification cost may be unfair. To overcome the above disadvantages, scientists have proposed a number of alternatives of AUC, such as H measure (Hand 2009), concentrated ROC (Swamidass et al., 2010), and normalized discounted cumulative gain (Wang et al., 2013). At the same time, the area under the precision-recall curve (AUPR) becomes increasingly popular, especially for biological studies. Though the AUPR score is less interpretable than AUC, each point in the PR curve has an explicit meaning, and the absolute accuracy metrics (e.g., precision and recall) are usually closer to practical requirements than relative ranks.

In summary, empirical comparisons and systematic analyses about evaluation metrics for link prediction are important tasks in this stage because many publications use AUC as the sole metric, while an ongoing empirical study (by Y.-L. Lee and T. Zhou, unpublished) shows that in about 1/3 pairwise comparisons, AUC and AUPR give different ranks of algorithms, and a recent large-scale experimental study also shows inconsistent results by AUC and AUPR (Muscoloni and Cannistraci, 2021). Before a comprehensive and explicit picture obtained, my suggestion is that we have to at least simultaneously report both AUC and AUPR, and only if an algorithm can obviously beat another one in both metrics for a network, we can say the former performs better in this case.

LOCAL SIMILARITY INDICES

A similarity-based algorithm will assign a similarity score to each nonobserved link, and the one with a higher score is of a larger likelihood to be a missing link. Liben-Nowell and Kleinberg (2007) indicated that a very simple index named "common neighbor" (CN), say

$$S_{xy}^{CN} = |\Gamma_x \cap \Gamma_y|, \quad \text{(Equation 1)}$$

with Γ_x and Γ_y being sets of neighbors of nodes x and y, performing very well in link prediction for social networks. Zhou et al. (2009) proposed the "resource allocation" (RA) index via weakening the weights of large-degree common neighbors, namely

$$S_{xy}^{RA} = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{k_z}, \quad \text{(Equation 2)}$$

where k_z is the degree of node z. The simplicity, elegance, and good performance of CN, RA, and some other alternatives (Lü and Zhou, 2011) lead to increasing attention on local similarity indices.

In the recent decade, probably, the most impressive achievement on local similarity indices is the proposal of the local community paradigm (Cannistraci et al., 2013a), which suggests that two nodes are more likely to link together if their common neighbors are densely connected. Accordingly, Cannistraci et al. (2013a) proposed the CAR index where the CN index is multiplied by the number of observed links between common neighbors, as follows:

$$S_{xy}^{CAR} = S_{xy}^{CN} \cdot \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{|\gamma_z|}{2}, \quad \text{(Equation 3)}$$

where γ_z is the subset of z's neighbors that are also common neighbors of x and y. Analogously, RA index can be improved by accounting for the local community paradigm as follows:

$$S_{xy}^{CRA} = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{|\gamma_z|}{k_z}. \quad \text{(Equation 4)}$$

By integrating the idea of Hebbian learning rule, the above index is further extended and renamed as Cannistraci-Hebb (CH) index (Muscoloni et al., 2018):

$$S_{xy}^{CH} = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1 + k_z^{(i)}}{1 + k_z^{(e)}}, \quad \text{(Equation 5)}$$

where $k_z^{(i)}$ is the internal degree of z, say the number of z's neighbors that are also in $\Gamma_x \cap \Gamma_y$, and $k_z^{(e)}$ is the external degree of z, say the number of z's neighbors that are not in $\Gamma_x \cap \Gamma_y \cap \{x, y\}$. The core idea of CH index is to consider the negative impacts of external local community links (see Muscoloni et al. (2020) for more CH indices according to the core idea). Extensive empirical analyses (Muscoloni et al., 2020; Zhou et al., 2021) indicated that the introduction of local community paradigm and Hebbian learning rule could considerably improve the performance of routine local similarity indices.

In most known studies, the presence of many 2-hop paths between a pair of nodes is considered to be the strongest evidence indicating the existence of a corresponding missing link or future link. Although in local path index (Lü et al., 2009) and Katz index (Katz, 1953) longer paths are taken into account, they are considered to be less significant than 2-hop paths. Surprisingly, some recent works have argued that 3-hop-based similarity indices perform better than 2-hop-based indices. Pech et al. (2019) assumed that the existence likelihood of a link is a linear sum of all its neighbors' contributions. After some algebra, Pech et al. (2019) obtained a global similarity index called linear optimization (LO) index, as follows:

$$S^{LO} = \alpha A\left(\alpha A^T A + I\right)^{-1} A^T A = \alpha A^3 - \alpha^2 A^5 + \alpha^3 A^7 - \alpha^4 A^9 + \cdots, \quad \text{(Equation 6)}$$

where A and I are adjacency matrix and identity matrix, respectively. Clearly, the number of 3-hop paths A^3 can be interpreted as a degenerated index of LO. Indeed, Daminelli et al. (2015) have already applied 3-hop-based indices to predict missing links in bipartite networks, while almost no one at that time has tried 3-hop-based indices on unipartite networks. Kovács et al. (2019) noted the bipartite nature of the protein-protein interaction networks (not fully bipartite, but of high bipartivity (Holme et al., 2003)) and independently proposed a degree-normalized index (called *L3* index) based on 3-hop paths as follows:

$$S_{xy}^{L3} = \sum_{u,v} \frac{a_{xu} a_{uv} a_{vy}}{\sqrt{k_u k_v}}. \quad \text{Equation (7)}$$

They showed its advantage compared with 2-hop-based indices in predicting protein-protein interactions. Muscoloni et al. (2018) further proposed a theory that generalized 2-hop-based indices to *n*-hop-based indices with $n > 2$ and demonstrated the superiority of 3-hop-based indices over 2-hop-based indices on protein-protein interaction networks, world trade networks, and food webs. For example, in their framework (Muscoloni et al., 2018), the *n*-hop-based RA index reads as follows:

$$S_{xy}^{RA(n)} = \sum_{z_1, z_2, \cdots, z_{n-1} \in \mathbb{L}(n)} \frac{1}{\sqrt[n-1]{k_{z_1} k_{z_2} \cdots k_{z_{n-1}}}}, \quad \text{(Equation 8)}$$

where $\mathbb{L}(n)$ is the set of all *n*-hop simple paths connecting *x* and *y*, and $z_1, z_2, \cdots, z_{n-1}$ are the intermediate nodes on the considered path. Accordingly, the *L*3 index is exactly the same to the 3-hop-based RA index.

We have implemented extensive experiments on 137 real networks (Zhou et al., 2021), suggesting that (i) 3-hop-based indices outperform 2-hop-based indices subject to AUC, while 3-hop-based and 2-hop-based indices are competitive on precision; (ii) CH indices perform the best among all considered candidates; and (iii) 3-hop-based indices are more suitable for disassortative networks with lower densities and lower average clustering coefficient. Furthermore, we have showed that a hybrid of 2-hop-based and 3-hop-based indices via collaborative filtering techniques can result in overall better performance (Lee and Zhou, 2021).

LINK PREDICTABILITY

Quantifying link predictability of a network allows us to evaluate link prediction algorithms for this network and to see whether there is still a large space to improve the current prediction accuracy. Lü et al. (2015) raised a hypothesis that missing links are difficult to predict if their addition causes huge structural changes, and thus, a network is highly predictable if the removal or addition of a set of randomly selected links does not significantly change structural features of this network. Denote *A* the adjacency matrix of a simple network $G(V, E)$ and ΔA the adjacency matrix corresponding to a set of randomly selected links ΔE from *E*. After the removal of ΔE, the remaining network G^R is also a simple network so that the corresponding adjacency matrix, $A^R = A - \Delta A$, can be diagonalized as follows:

$$A^R = \sum_{k=1}^{N} \lambda_k x_k x_k^T, \quad \text{(Equation 9)}$$

where $N = |V|$ and λ_k and x_k are the *k*th eigenvalue and corresponding orthogonal and normalized eigenvector of A^R, respectively. Considering ΔE as a perturbation to A^R, which results in an updated eigenvalue $\lambda_k + \Delta\lambda_k$ and a corresponding eigenvector $x_k + \Delta x_k$, then we have the following equation:

$$(A^R + \Delta A)(x_k + \Delta x_k) = (\lambda_k + \Delta\lambda_k)(x_k + \Delta x_k). \quad \text{(Equation 10)}$$

Similar to the process to get the expectation value of the first-order perturbation Hamiltonian, we neglect the second-order small terms and the changes of eigenvectors and then obtain the following equation:

$$\Delta\lambda_k \approx \frac{x_k^T \Delta A x_k}{x_k^T x_k}, \quad \text{(Equation 11)}$$

as well as the perturbed matrix:

$$\tilde{A} = \sum_{k=1}^{N} (\lambda_k + \Delta\lambda_k) x_k x_k^T, \quad \text{(Equation 12)}$$

which can be considered as the linear approximation of *A* if the expansion is only based on A^R. If the perturbation does not significantly change the structural features, the eigenvectors of A^R and those of *A* should be almost the same, and thus, $\tilde{A}$ should be very close to *A* according to Equation (12). We rank all links in $U\ E^R$ in a descending order according to their values in $\tilde{A}$ and select the top-*L* links to form the set E^L, where $L = |\Delta E|$. Links in E^R and E^L constitute the perturbed network, and if this network is close to *G* (because $\tilde{A}$ is close to *A*), E^L should be close to ΔE. Therefore, Lü et al. (2015) finally proposed an index called structural consistency to measure the inherent difficulty in link prediction as follows:

$$\sigma_c = \frac{|E^L \cap \Delta E|}{|\Delta E|}. \quad \text{Equation (13)}$$

The above perturbation method can also be applied to predict missing links, and the resulted "structural perturbation method" (SPM) is still one of the most accurate methods till far (Muscoloni and Cannistraci, 2021).

Koutra et al. (2015) found that the major part of a seemingly complicated real network can be represented by a few elemental substructures like cliques, stars, chains, bipartite cores, and so on. Inspired by this study, Xian et al. (2020) claimed that a network is more regular and thus more predictable if it can be well represented by a small number of subnetworks. To reduce the tremendous complexity caused by countless subnetworks, they further set a strong restriction that candidate subnetworks are ego networks of all nodes, where ego network (also called egocentric network) is a subnetwork induced by a central node (known as the ego) and all other nodes directly connected to the ego (called alters), see (Wang et al., 2016a) for example. Obviously, the ego network of node i can be represented by the ith row or ith column of the adjacency matrix A, and if a network can be perfectly represented by all ego networks, there exists a matrix $Z \in \mathbb{R}^{N\times N}$ such that $A = AZ$. Intuitively, if a network G is very regular, the corresponding representation Z should have three properties: (i) G can be well represented by its ego networks so that AZ is close to A; (ii) G can be well represented by a small number of ego networks so that Z is of low rank since the redundant ego networks correspond to zero rows in Z; and (iii) each ego network of G can be represented by a very few other ego networks so that Z is sparse. Accordingly, the best representation matrix Z^* can be obtained by solving the following optimization problem:

$$\min_Z \text{rank}(Z) + \alpha\|Z\| + \beta\|A - AZ\|, \quad \text{(Equation 14)}$$

where α and β are tradeoff parameters. Based on Z^*, Xian et al. (2020) proposed an ad hoc index named structural regularity index, as follows:

$$\sigma_r = \frac{1}{\sqrt{\frac{n-r}{n}}\sqrt{\frac{\tau}{nr}}}, \quad \text{(Equation 15)}$$

where r is the rank of Z^*, τ is the number of zero entries in Z^*, $\frac{n-r}{n}$ denotes the proportion of identical ego networks, and $\frac{\tau}{nr}$ characterizes the density of zero entries of the reduced echelon form of Z^*. Clearly, a lower r and a larger τ will result in a smaller σ_r, corresponding to a more predictable network.

Xian et al. (2020) suggested that the structural regularity corresponds to redundant information in the adjacency matrix, which can be characterized by a low-rank and sparse representation matrix. Sun et al. (2020) proposed a more direct method to measure such redundancy. Their train of thought is that a more predictable network contains more structural redundancy and thus can be compressed by a shorter binary string. As the shortest possible compression length can be calculated by a lossless compression algorithm, they used the obtained normalized shortest compression length of a network to quantify its structure predictability.

To validate their methods, Lü et al. (2015) and Xian et al. (2020) tested on many real networks about whether σ_c and σ_r are strongly correlated with prediction accuracies of a few well-known algorithms. This is rough because any algorithm cannot stand for the theoretical best algorithm. Sun et al. (2020) adopted an improved method that uses the best performance among a number of known algorithms for each tested network to approximate the performance of the theoretically best predicting algorithm. Garcia-Perez et al. (2020) analyzed the ensemble of simple networks, where each can be constructed by generating a link between any node pair i and j with a known linking probability p_{ij}. For such theoretical benchmark, the best possible algorithm is to rank unobserved links with largest linking probabilities in the top positions and the theoretical limitation of precision can be easily obtained. They showed that if the size of ΔA is too small in compared with A, the evaluation of predictability by σ_c is less accurate.

Structural consistence, structural regularity, and compression length are all ad hoc methods. They can be used to probe the intrinsic difficulty in link prediction but cannot mathematically formulate the limitation of prediction. Mathematically speaking, there could be a God algorithm that correctly predicts all missing links, except for those indistinguishable from nonexistent links. A link (i, j) is indistinguishable from another link (u, v) if and only if there is a certain automorphism f such that $f(i) = u$ and $f(j) = v$, or

$f(i) = v$ and $f(j) = u$. This extremely rigid definition from automorphism-based symmetry makes virtually all real networks have predictability very close to 1, which is indeed meaningless in practice. Using synthetic networks with known prediction limitation is a potentially promising way to evaluate predictors as well as indices for predictability (Garcia-Perez et al., 2020; Muscoloni and Cannistraci, 2018b), but the results may be irrelevant to real-world networks.

All above studies target static networks, while a considerable fraction of real networks are time varying (named as temporal networks) (Holme and Saramäki, 2012). Temporal networks are usually more predictable since one can utilize both topological and temporal patterns. Ignoring topological correlations, the randomness, and thus predictability of a time series can be quantified by the entropy rate (Xu et al., 2019). Tang et al. (2020) listed weights of all possible links as an expanded vector with dimension N^2 (self-connections are allowed, directionalities are considered), and thus, the evolution of a temporal network can be fully described by a matrix $M \in \mathbb{R}^{N^2 \times T}$, where T is the number of snapshots under consideration. After M, the evolution of a temporal network can be treated as a stochastic vector process, and how to measure the predictability of temporal networks is transformed to a solved problem based on the generalized Lempel-Ziv algorithm (Kontoyiannis et al., 1998). An obvious defect is that the vector dimension is too big, resulting in huge computational complexity. Tang et al. (2020) thus replaced M by a much smaller matrix where only links occurring $\geq 10\%$ of snapshots are taken into consideration. An intrinsic weakness of Lempel-Ziv algorithm is that it tends to overestimate the predictability, and thus, in many situations the estimated values are very close to 1 (Xu et al., 2019; Song et al., 2010). Tang et al. (2020) proposed a clever method that compares the predictability of the target network with the corresponding null network, and thus, the normalized predictability is able to characterize the topological-temporal regularity in addition to the least predictable one.

NETWORK EMBEDDING

A network embedding algorithm will produce a function $g : V \rightarrow \mathbb{R}^d$ with $d \ll N$ so that every node is represented by a low-dimensional vector (Cui et al., 2018). Then, the existence likelihood of a nonobserved link (u, v) can be estimated by the inner product, the cosine similarity, the Euclidean distance, or the geometrical shortest path of the two learned vectors $g(u)$ and $g(v)$ (Cui et al., 2018; Cannistraci et al., 2013b). Early methods cannot handle large-scale networks because they usually rely on solving the leading eigenvectors (Tenenbaum et al., 2000; Roweis and Saul, 2000).

Mikolov et al. (2013a, 2013b) proposed a language embedding algorithm named SkipGram that represents every word in a given vocabulary by a low-dimensional vector. Such representation can be obtained by maximizing the co-occurrence probability among words appearing within a window t in a sentence, via some stochastic gradient descent methods. Based on SkipGram, Perozzi et al. (2014) proposed the so-called DeepWalk algorithm, where nodes and truncated random walks are treated as words and sentences. Grover and Leskovec (2016) proposed the node2vec algorithm that learns the low-dimensional representation by maximizing the likelihood of preserving neighborhoods of nodes. Grover and Leskovec argued that the choice of neighborhoods plays a critical role in determining the quality of the representation. Therefore, instead of simple definitions of the neighborhood of an arbitrary node u, such as nodes with distance no more than a threshold to u (like the breadth-first search) and nodes sampled from a random walk starting from u (like the depth-first search), they utilized a flexible neighborhood sampling strategy by biased random walks, which smoothly interpolates between breadth-first search and depth-first search. Considering a random walk that just traversed link (z, v) and now resides at node v, the transition probability from v to any v's immediate neighbor x is $\pi_{vx} / \sum_{y \in \Gamma_v} \pi_{vy}$. The node2vec algorithm sets the unnormalized probability as follows:

$$\pi_{vx} = \alpha_{pq}(z, x) \cdot w_{vx}, \qquad \text{(Equation 16)}$$

where w_{vx} is the weight of link (v, x) and

$$\alpha_{pq}(z, x) = \begin{cases} 1/p, & d_{zx} = 0 \\ 1, & d_{zx} = 1 \\ 1/q, & d_{zx} = 2 \end{cases}. \qquad \text{(Equation 17)}$$

Obviously, the sampling strategy in DeepWalk is a special case of node2vec with $p = 1$ and $q = 1$. By tuning p and q, node2vec can achieve better performance than DeepWalk in link prediction.

Tang et al. (2016) argued that DeepWalk lacks a clear objective function tailored for network embedding and proposed the LINE algorithm that learns node representations on the basis of a carefully designed objective function that preserves both the first-order and second-order proximity. The first-order proximity is captured by the observed links and thus can be formulated as follows:

$$O_1 = -\sum_{(u,\,v)\in E} w_{uv} \log p_1(u,\,v), \qquad \text{(Equation 18)}$$

where w_{uv} is the weight of the observed link (u, v) and

$$p_1(u,\,v) = \frac{1}{1+\exp[-g(u)\cdot g(v)]} \qquad \text{(Equation 19)}$$

describes the likelihood of the existence of (u, v) given the embedding g. Of course, one can adopt other alternatives of Equation (19). The second-order proximity assumes that nodes sharing many connections to other nodes are similar to each other. Accordingly, each node is also treated as a specific context and nodes with similar distributions over contexts are assumed to be similar. Then, the second-order proximity can be characterized by the objective function:

$$O_2 = -\sum_{(u,\,v)\in E} w_{uv} \log p_2(u|v), \qquad \text{(Equation 20)}$$

where $p_2(u|v)$ denotes the probability that node v will generate a context u, namely

$$p_2(u|v) = \frac{\exp[g'(u)\cdot g(v)]}{\sum_{z\in V}\exp[g'(z)\cdot g(v)]}, \qquad \text{(Equation 21)}$$

with $g'(u)$ being the context representation of u. Clearly, O_2 is naturally suitable for directed networks. By minimizing O_1 and O_2, LINE learns two kinds of node representations that, respectively, preserve the first-order and second-order proximity and takes their concatenation as the final representation.

In addition to DeepWalk, LINE, and node2vec, other well-known network embedding algorithms that have been applied in link prediction include DNGR (Cao et al., 2016), SDNE (Wang et al., 2016b), HOPE (Ou et al., 2016), GraphGAN (Wang et al., 2018), and so on. On the one hand, embedding is currently a very hot topic in network science and thought to be a promising method for link prediction. On the other hand, some very recent empirical studies (Muscoloni et al., 2020; Mara et al., 2020; Ghasemian et al., 2020) involving more than a thousand networks showed negative evidence that network embedding algorithms perform worse than some elaborately designed mechanistic algorithms. This is not a bad news because one can expect that some link prediction algorithms will enlighten and energize researchers in network embedding and thus make contributions to other aspects of network analyses like community detection, classification, and visualization.

Another notable embedding method is based on the hyperbolic network model (Krioukov et al., 2010; Papadopoulos et al., 2012), where each node is represented by only two coordinates (i.e., $d = 2$) in a hyperbolic disk. The hyperbolic network models are very simple yet can reproduce many topological characteristics of real networks, such as sparsity, scale-free degree distribution, clustering, small-world property, community structure, self-similarity, and so on (Krioukov et al., 2010; Papadopoulos et al., 2012; Muscoloni and Cannistraci, 2018a). Papadopoulos et al., (2014) applied the hyperbolic model to the Internet at Autonomous Systems (AS) level and showed better performance than traditional methods (e.g., CN index and Katz index) in predicting missing links associated with low-degree nodes (see again (Zhu et al., 2012) for biased sampling preferring low-degree nodes). Wang et al. (2016c) proposed a link prediction algorithm for networks with community structure based on hyperbolic embedding, showing good performance for community networks with power-law degree distributions. Alessandro and Cannistraci proposed the so-called nonuniform popularity-similarity-optimization (nPSO) model as a generative model to grow random networks embedded in the hyperbolic space (Muscoloni and Cannistraci, 2018a), and they leveraged the nPSO model as a synthetic benchmark for link prediction algorithms, showing that the algorithm only accounting for hyperbolic distance does not perform well at the presence of communities (Muscoloni and Cannistraci, 2018b). They further proposed a variant of geometric embedding named minimum curvilinear automata (MCA) (Muscoloni and Cannistraci, 2018c), whose link prediction accuracy is higher than the simple hyperbolic distance (Muscoloni and Cannistraci, 2018b) but lower than coalescent embedding in hyperbolic space (Muscoloni et al., 2017). Kitsak et al. (2020) proposed an embedding algorithm named

HYPERLINK in hyperbolic space, whose goal is to obtain more accurate prediction of missing links, so that HYPERLINK performs better than previous hyperbolic embedding algorithms (most of these algorithms are designed to reproduce topological features, not to predict missing links). HYPERLINK is often competitive to other well-known link predictors, and it is in particular good at predicting missing links that are really hard to predict.

MATRIX COMPLETION

Matrix completion aims to reconstruct a target matrix, given a subset of known entries. Since links can be fully conveyed by the adjacency matrix A, it is natural to regard link prediction as a matrix completion task. Denote E^k the set of node pairs corresponding to known entries in A that can be utilized in the matrix completion task. In most studies, $E^k = E^T$, while we should be aware of that E^k can also contain some known absent links. The matrix completion problem can be formulated in line with supervised learning, as follows:

$$\min_{\vartheta} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \ell\left[a_{ij}, \tilde{a}_{ij}(\vartheta)\right] + \Omega(\vartheta), \qquad \text{(Equation 22)}$$

where ϑ is the parameter vector, $\tilde{a}_{ij}$ is the predicted value of the model, $\ell(\bullet,\bullet)$ is a loss function, and Ω is a regularization term preventing overfitting. The most frequently used loss functions are squared loss $\ell(a,b) = (a-b)^2$ and logistic loss $\ell(a,b) = \log(1+e^{-ab})$.

Matrix factorization is a very popular method for matrix completion, which has already achieved great success in a closely related domain, the design of recommender systems (Koren et al., 2009). We consider a simple network with symmetry A and assume $\tilde{A}$ can be approximately factorized as $\tilde{A} \approx UU^T$ with $U \in \mathbb{R}^{N\times d}$ and $d \ll N$; then, we need to solve the following optimization problem:

$$\min_{U} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \ell\left(a_{ij}, u_i^T u_j\right) + \Omega(U), \qquad \text{(Equation 23)}$$

where u_i and u_j are the ith and jth rows of U., respectively Notice that u_i^T is the transpose of u_i not the ith row of U^T. Though without topological interpretation, u_i can be treated as a lower-dimensional representation of node i, and matrix factorization can also be considered as a kind of matrix embedding algorithms (Qiu et al., 2019). If we adopt the squared loss function and the Forbenius norm for Ω, then we get a specific optimization problem:

$$\min_{U} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \left(a_{ij} - u_i^T u_j\right)^2 + \lambda \|U\|_F^2, \qquad \text{(Equation 24)}$$

where λ is a tradeoff parameter. Menon and Elkan (2011) suggested that one can directly optimize for AUC on the training set, given some known absent links. Accordingly, the objective function Equation (23) can be rewritten in terms of AUC as follows:

$$\min_{U} \frac{1}{|E_+^k||E_-^k|} \sum_{(i,j)\in E_+^k,\ (x,y)\in E_-^k} \ell\left(1, u_i^T u_j - u_x^T u_y\right) + \Omega(U), \qquad \text{(Equation 25)}$$

where E_+^k and E_-^k are sets of known present and known absent links, respectively. Cleary, $E_+^k \cap E_-^k = \varnothing$ and $E_+^k \cup E_-^k = E^k$. Menon and Elkan (2011) showed that the usage of AUC-based loss function can improve AUC value by around 10% comparing with the routine loss function like Equation (24).

The factorization in Equation (24) is easy to be extended to directed networks (Menon and Elkan, 2011), bipartite networks (Natarajan and Dhillon, 2014), temporal networks (Ma et al., 2018), and so on. For example, if A is asymmetry, we can replace $\tilde{A} \approx UU^T$ by $\tilde{A} \approx U\Lambda U^T$ with $\Lambda \in \mathbb{R}^{d\times d}$, and thus, Equation (24) can be extended to the following:

$$\min_{U,\Lambda} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \left(a_{ij} - u_i^T \Lambda u_j\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_\Lambda}{2}\|\Lambda\|_F^2, \qquad \text{(Equation 26)}$$

Analogously, for bipartite networks like gene-disease associations (Natarajan and Dhillon, 2014) and user-product purchases (Shang et al., 2010), if $A \in \mathbb{R}^{M\times N}$, then we can replace $\tilde{A} \approx UU^T$ by $\tilde{A} \approx WH^T$, where $W \in \mathbb{R}^{M\times d}$ and $H \in \mathbb{R}^{N\times d}$. Accordingly, we get the optimization problem for bipartite networks as follows:

$$\min_{W,H} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \left(a_{ij} - w_i^T h_j\right)^2 + \frac{\lambda_W}{2}\|W\|_F^2 + \frac{\lambda_H}{2}\|H\|_F^2, \quad \text{(Equation 27)}$$

where w_i and h_j are the ith and jth rows of W and H, respectively. More details can be found in Menon and Elkan (2011), Natarajan and Dhillon (2014), and Ma et al. (2018).

The explicit features of nodes, such as tags associated with users and products (Zhang et al., 2011), can also be incorporated in the matrix factorization framework. Menon and Elkan (2011) suggested a direct combination of explicit features and latent features learned from the observed topology. Denoting $x_i \in \mathbb{R}^s$ the vector of explicit features of node i, the predicted values $\tilde{a}_{ij}$ in Equation (22) are then replaced by

$$\tilde{a}_{ij} = u_i^T u_j + v^T x_i + v^T x_j, \quad \text{(Equation 28)}$$

where $v \in \mathbb{R}^s$ is a vector of parameters. Experiments showed that the incorporation can considerably improve the prediction accuracy (Menon and Elkan, 2011). Jain and Dhillon (2013) proposed a so-called inductive matrix completion (IMC) algorithm that uses explicit features to reduce the computational complexity. In IMC, the predicted value can be expressed as $\tilde{a}_{ij} = x_i^T QQ^T x_j$, where $x_i \in \mathbb{R}^s$ is the vector of i's explicit features, and $Q \in \mathbb{R}^{s \times t}$ is a low-rank matrix with small t, which describes the latent relationships between explicit features and topological structure. Q can be learned from observed links by the following optimization problem:

$$\min_{Q} \frac{1}{|E^k|} \sum_{(i,j)\in E^k} \left(a_{ij} - x_i^T QQ^T x_j\right)^2 + \lambda \|Q\|_F^2. \quad \text{(Equation 29)}$$

Notice that, in Equation (24), the number of parameters to be learned is Nd, while in Equation (29), we only need to learn st parameters. The numbers of latent features (d and t) could be more or less the same as they are largely dependent on the topological structure, while the number of nodes N is usually much larger than the number of explicit features s. Therefore, the computational complexity of IMC should be much lower than brute-force factorization methods. The original IMC is proposed for bipartite networks (Jain and Dhillon, 2013), which has already found successful applications in the design of recommender systems (Jain and Dhillon, 2013) and the prediction of gene- and RNA-disease associations (Natarajan and Dhillon, 2014; Lu et al., 2018; Chen et al., 2018).

Pech et al. (2017) argued that low rank is the most critical property in matrix completion. They assumed that the observed network can be decomposed into two parts as $A = A_B + A_E$, where A_B is called the backbone preserving the network organization pattern, and A_E is a noise matrix, in which positive and negative entries are spurious and missing links, respectively. Pech et al. (2017) considered only two simple properties: the low rank of A_B and the sparsity of A_E. Accordingly, A_B and A_E can be determined by solving the following optimization problem:

$$\min_{A_B, A_E} \text{rank}(A_B) + \lambda \|A_E\|_0 \text{ subject to } A = A_B + A_E, \quad \text{(Equation 30)}$$

where A_{E0} is the l_0-norm counting the number of nonzero entries of A_E. The predicted links can be obtained by sorting entries in A_B that correspond to zero entries in A. This method is straightforwardly named as low rank (LR) algorithm. Although being simple, LR performs better than well-known similarity-based algorithms reported in Zhou et al. (2009) and Cannistraci et al. (2013a), hierarchical structure model (Clauset et al., 2008), and stochastic block model (Guimerà and Sales-Pardo, 2009), while slightly worse than LOOP (Pan et al., 2016), structural perturbation model (Lü et al., 2015), and Cannistraci-Hebb automata (Muscoloni et al., 2020).

ENSEMBLE LEARNING

In an early survey (Lü and Zhou, 2011), we noticed the low stability of individual link predictors and thus suggested ensemble learning as a powerful tool to integrate them. Ensemble learning is a popular method in machine learning, which constructs and integrates a number of individual predictors to achieve better algorithmic performance (Zhou, 2012). Roughly speaking, ensemble learning techniques can be divided into two classes: the "parallel ensemble" where individual predictors do not strongly depend on each other and can be implemented simultaneously (e.g., bagging (Breiman, 1996) and random forests (Breiman, 2001)) and the "sequential ensemble" where the integration of individual predictors has to be implemented in a sequential way (e.g., boosting (Freund and Schapire, 1997) and stacking (Wolpert, 1992)). In the following, we will, respectively, introduce how parallel ensemble and sequential ensemble are applied to link prediction.

Given an observed network, an individual link predictor will produce a rank of all unobserved links. Pujari and Kanawati (2012) proposed an aggregation approach on ranks resulted from individual algorithms. If there are R ranks produced by R individual predictors, an unobserved link e's Borda count $B_k(e)$ in the kth rank can be defined as the number of links ranked ahead of e (there are many variants of Borda count and here we use the simplest one). Pujaji and Kanawati used a weighted aggregation to obtain the final score of any unobserved link e, as follows:

$$B(e) = \sum_{k=1}^{R} w_k B_k(e), \quad \text{(Equation 31)}$$

where w_k is set to be proportional to the precision of the kth predictor trained by the observed network. Clearly, smaller $B(e)$ indicates higher existence likelihood. In addition to rank aggregation, a similar weighting technique can also be applied in integrating likelihood scores. If every unobserved link is assigned a score (higher score indicates higher existence likelihood) by each predictor, then the final score of any unobserved link e can be defined in a weighted form as follows:

$$S(e) = \sum_{k=1}^{R} w_k S_k(e), \quad \text{(Equation 32)}$$

where $S_k(e)$ is the score from the kth predictor. Different from rank aggregation, $S_k(\bullet)$ $(k = 1, 2, \cdots, R)$ should be normalized before the weighted sum to ensure scores from different predictors are comparable. An alternative aggregation method is the ordered weighted averaging (OWA) (Yager, 1988), where the R predictors are ordered according to their importance to the final prediction, as $S^{(1)}, S^{(2)}, \cdots, S^{(R)}$, and then the final score of any unobserved link e is

$$S(e) = \sum_{k=1}^{R} w_k S^{(k)}(e), \quad \text{(Equation 33)}$$

where $\sum_{k=1}^{R} w_k = 1$ and $w_1 \geq w_2 \geq \cdots \geq w_R \geq 0$. Without prior information, the most usual way to determine the weights is using the maximum entropy method, which maximizes $-\sum_{k=1}^{R} w_k \ln w_k$ subject to $\sum_{k=1}^{R} w_k = 1$ and $\eta = \frac{1}{R-1}\sum_{k=1}^{R-1}(R-k)w_k$, where η is a tunable parameter measuring the extent to which the ensemble (33) is like an *or* operation. If η is very large, then $w_1 \approx 1$ and $w_k \approx 0$ $(k \geq 2)$, that is to say, only the first predictor works. He et al. (2015) applied OWA to aggregate nine local similarity indices. These indices are ordered according to their normalized values (irrelevant to their qualities), which is essentially unreasonable. Therefore, although He et al. (2015) reported considerable improvement, later experiments (Wu et al., 2019; Zhang et al., 2020) indicated that the method in He et al. (2015) does not work well because the position of a predictor is irrelevant to its quality. In contrast, if the order is relevant to the predictors' qualities (e.g., according to their precisions trained by the target network), OWA will bring in remarkable improvement compared with individual predictors (Wu et al., 2019). As some link prediction algorithms scale worse than $\mathbb{O}(N)$, Duan et al. (2017) argued that to solve smaller problems multiple times is more efficient than to solve a single large problem. They considered a latent factor model (similar to the one described by Equation (23), with complexity $\mathbb{O}(Nd^2)$) and developed several ways for the bagging decomposition, such as bagging with random nodes together with their immediate neighbors and bagging preferring dense components. They showed that those bagging techniques can largely reduce computational complexity without sacrificing prediction accuracy. Considering the family of stochastic block models (Guimerà and Sales-Pardo, 2009), Valles-Catala et al. (2018) showed that the integration (via Markov Chain Monte Carlo sampling according to Bayesian rules) of individually less plausible models can result in higher predictive performance than the single most plausible model.

Boosting is a typical sequential ensemble algorithm that trains a base learner from initial training set and adjusts weights of instances (the wrongly predicted instances will be enhanced while the easy-to-be-predicted instances will lose weights) in the training set to train the next learner. Such operation will continue until reaching some preseted conditions. The most representative boosting algorithm is AdaBoost (Freund and Schapire, 1997), which is originally designed for binary classification and thus can be directly applied in link prediction. AdaBoost is an additive model as

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \quad \text{(Equation 34)}$$

where h_t is the tth base learner, α_t is a scalar coefficient, and T is a preseted terminal time. H aims to minimize the expected value of an exponential loss function:

$$\ell(H, W) = \mathbb{E}_{x\sim W}\left[e^{-f(x)H(x)}\right], \quad \text{(Equation 35)}$$

where $f(x)$ denotes the true class of the instance x and W is the original weight distribution. Without specific requirements, we usually set $W(x) = 1/m$ for every instance x where m is the number of instances. We set $W_1 = W$ and learn h_1 from W_1, and then, α_1 is determined by minimizing $\ell(\alpha_1 h_1, W_1)$. The weight of an instance x in the second step is updated as follows:

$$W_2(x) = \begin{cases} \frac{1}{Z_2} W_1(x) e^{-\alpha_1}, & h_1 = f(x) \\ \frac{1}{Z_2} W_1(x) e^{\alpha_1}, & h_1 \neq f(x) \end{cases}, \quad \text{Equation (36)}$$

where Z_2 is the normalization factor. Obviously, if the instance x can be correctly classified, its weight will decrease; otherwise, its weight will increase. Such process iterates until reaching the terminal time T. When applying AdaBoost in link prediction, we need to be aware of the following three issues. (i) The base learner should be sensitive to W_t so that we cannot use unsupervised algorithms or supervised algorithms insensitive to W_t. (ii) In addition to positive instances (observed links), negative instances should be sampled from unobserved links. Though it introduces some noise, the influence is ignorable if the network is sparse. (iii) The negative instances should be undersampled to keep the data balanced. Comar et al. (2011) proposed the so-called LinkBoost algorithm, which is an extension of AdaBoost to link prediction with a typical matrix factorization model being the base learner. Instead of undersampling negative instances, they suggest a cost-sensitive loss function which penalizes the misclassifying links as nonlinks about N times heavier than misclassifying nonlinks as links. They further considered a degree-sensitive loss function that penalizes more for misclassification of links between low-degree nodes than high-degree nodes.

Stacking (Wolpert, 1992) is another powerful approach in sequential ensemble. It trains a group of primary learners from the initial training set and uses the outputs of primary learners as input features to train the secondary learner that provides the final prediction. If both primary learners (i.e., input features) and training instances are directly generated by the same training set, the risk of overfitting will be very high. Therefore, the original training set D, usually containing similar numbers of positive and negative instances for data balance, is divided into J sets with same size as $D_1, D_2, \cdots, D_J$. Denoting $h_r^{(j)}$ the primary learner using the rth algorithm and trained from the jth fold of the training set $\overline{D_j} = D \backslash D_j$, for each instance $x_i \in D_j$, its feature vector is $z_i = (z_{i1}, z_{i2}, \cdots, z_{iR})$, where $z_{ir} = h_r^{(j)}(x_i)$ and R is the number of primary algorithms. This J-fold division ensures all features of any instance x are obtained by primary learners trained without x. Some scientists have already used similar techniques (e.g., using various regressions to integrate results from primary predictors and other features (Zhang et al., 2020; Zhang, 2017; Fire et al., 2013)), but they are not aware of stacking model and did not employ any measures to avoid overfitting. Li et al. (2020) proposed a stacking model for link prediction, which use logistic regression and XGBoost to learn 4 similarity indices. Their method is inspiring, but they only considered 4 primary predictors and tested on two very small networks with some experimental results (e.g., the AUC values of CN index) far different from well-known results, and thus, the reported results and conclusion are questionable. Ghasemian et al. (2020) proposed a stacking model that considers 203 primary link predictors on 550 disparate networks. Using a standard supervised random forest algorithm (Breiman, 2001) as the secondary learner, Ghasemian et al. (2020) argued that the stacking model is remarkably superior to individual predictors for real networks and can approach to the theoretical optima for synthetic networks with known highest prediction accuracies. In addition, they showed that social networks are more predictable than biological and technological networks. However, a recent large-scale experiment (Muscoloni and Cannistraci, 2021) suggested that the above stacking model does not perform better than SPM (Lü et al., 2015) and Cannistraci-Hebb automata (Muscoloni et al., 2020). Wu et al. (2019) proposed an alternative sequential ensemble strategy called network reconstruction, which reconstructs network via one link prediction algorithm and predicts missing links by another prediction algorithm.

DISCUSSION

In this review, to improve the readability, we classify representative works in the last decade into five groups. Of course, some novel and interesting methods, such as evolutionary algorithm (Bliss et al., 2014), ant colony approach (Sherkat et al., 2015), structural Hamiltonian analysis (Pan et al., 2016), and

Table 1. Computational complexities of some representative link prediction algorithms

Algorithm	References	Complexity	Scalability
CN	Liben-Nowell and Kleinberg (2007)	$\mathbb{O}(Nk^2)$	High
RA	Zhou et al. (2009)	$\mathbb{O}(Nk^2)$	High
LP	Zhou et al. (2009)	$\mathbb{O}(Nk^3)$	Medium
CH	Muscoloni et al. (2018)	$\mathbb{O}(Nk^3)$	Medium
L3	Kovács et al. (2019)	$\mathbb{O}(Nk^3)$	Medium
SPM	Lü et al. (2015)	$\mathbb{O}(dN^2)$	Medium
DeepWalk	Perozzi et al. (2014)	$\mathbb{O}(\gamma N \log N)$	Medium
LINE	Tang et al. (2016)	$\mathbb{O}(tNk)$	Medium
NetSMF	Qiu et al. (2019)	$\mathbb{O}(Nd^2)$	Medium
IMC	Jain and Dhillon (2013)	$\mathbb{O}(Nkd^2)$	Medium
LR	Pech et al. (2017)	$\mathbb{O}(dN^2)$	Medium
HSM	Clauset et al. (2008)	$\mathbb{O}(e^N)$	Low
SBM	Guimerà and Sales-Pardo (2009)	$\mathbb{O}(e^N)$	Low
LOOP	Pan et al. (2016)	$\mathbb{O}(N^3)$	Low

N is the number of nodes, k is the average degree, γ is usually a large number depending on the number of random walks, the length of each walk, and the implemented deep learning model, t is the number of iterations, and d is the representation dimension or the preseted rank for matrix factorization and low-rank decomposition.

leading eigenvector control (Lee et al., 2021), do not belong to any of the above groups, and readers are encouraged to read other recent surveys (Wang et al., 2015; Martínez et al., 2016; Kumar et al., 2020) as complements of the present review.

Computational complexities of some representative algorithms mentioned in this perspective are reported in Table 1. Those algorithms are roughly categorized into three classes according to their computational complexities: the ones of "high" scalability can be applied to large-scale networks with millions of nodes, the ones of "medium" scalability can be applied to mid-sized networks with tens to hundreds of thousands of nodes, and the ones of "low" scalability can only deal with small networks with up to a few thousands of nodes using a common desktop computer. Given the size and sparsity of the target network, as well as the computational power, this table helps readers in finding suitable algorithms.

Very recently, a notable issue is the applications of neural networks in link prediction, which may be partially facilitated by the dramatic advances of deep learning techniques. Zhang and Chen (2017) trained a fully connected neural network on the adjacency matrices of enclosing subgraphs (with a fixed size) of target links. They applied a variant of the Weisfeiler-Lehman algorithm to determine the order of nodes in each adjacency matrix, ensuring that nodes with closer distances to the target link are ranked in higher positions. Zhang and Chen (2018) further proposed a novel framework based on graph neural networks, which can learn multiple types of information, including general structural features and latent and explicit node features. In this framework, a node's order in the enclosing subgraph can be determined only by its closeness to the target link and the subgraph size can be flexible. Wang et al. (2020) directly represented the adjacency matrix of a network as an image and then learned hierarchical feature representations by training generative adversarial networks. Some preliminary experimental results suggested that the performance of those methods (Zhang and Chen, 2017, 2018; Wang et al., 2020) is highly competitive to many other state-of-the-art algorithms. Despite of the promising results, at present, features and models are simply pieced together without intrinsic connections. The above pioneering works (Zhang and Chen, 2017, 2018; Wang et al., 2020) provide a good start but we still need in-depth and comprehensive analyses to push forward related studies.

Although most link prediction algorithms only account for structural information, attributes of nodes (e.g., expression levels of genes (Natarajan and Dhillon, 2014) and tags of citation and social networks (Zhang et al., 2011; Wang et al., 2019)) can be utilized to improve the prediction performance. It is easy to treat attributes as independent information additional to structural features and work out a method that directly combines the two, while what is lacking but valuable is to uncover nontrivial relationship between attributes and structural roles and then design more meaningful algorithms. Beyond explicit attributes, we should also pay

attention to dynamical information. It is known to us that limited time series obtained from some dynamical processes can be used to reconstruct network topology (Shen et al., 2014) while even a small fraction of missing links in modeling dynamical processes can lead to remarkable biases (Nicolaou and Motter, 2020); however, studies about how to make use of the correlations between topology and dynamics to predict missing links or how to take advantage of link prediction algorithms to improve estimates of dynamical parameters are rare.

By an elaborately designed model, Gu et al. (2017) showed that there is no ground truth in ranking influential spreaders even with a given dynamics. Peel et al. (2017) proved that there is no ground truth and no free lunch for community detection. The latter implies that no detection algorithm can be optimal on all inputs. Fortunately, we have ground truth in link prediction; however, extensive experiments (Ghasemian et al., 2020) also implicate that no known link predictor performs best or worst across all inputs. If link prediction is a no-free-lunch problem, then no single algorithm performs better or worse than any other when applied to all possible inputs. It raises a question that whether the study on prediction algorithms is valuable. The answer is of course YES (Guimerà, 2020) because we actually have free lunches as what we are interested in, the real networks, have far different statistics from those of all possible networks. As Ghasemian et al. (2020) argued that the ensemble models are usually superior to individual algorithms, a related question is whether the study on individual algorithm is valuable. The answer is still YES. Firstly, a recent large-scale experimental study (Muscoloni and Cannistraci, 2021) indicated that the performance of the stacking model is worse than elaborately designed individual algorithms, like SPM (Lü et al., 2015) and Cannistraci-Hebb automata (Muscoloni et al., 2020). Secondly, an individual algorithm could be highly cost-effective for its competitive performance and low complexity in time and space. Above all, individual algorithms, especially the mechanistic algorithms, may provide significant insights about network organization and evolution. In some real applications like friend recommendation, predictions with explanations are more acceptable (Barbieri et al., 2014), which cannot be obtained by ensemble learning. In addition, an alogical reason is that some elegant individual models (e.g., HSM (Clauset et al., 2008), SBM (Guimerà and Sales-Pardo, 2009), SPM (Lü et al., 2015), HYPERLINK (Kitsak et al., 2020), etc.) bring us inimitable esthetic perception that cannot be experienced elsewhere.

Along with fruitful algorithms proposed recently, the design of novel and effective algorithms for general networks is increasingly hard. We expect a larger fraction of algorithms in the future studies will be designed for networks of particular types (e.g., directed networks (Zhang et al., 2013), weighted networks (Zhao et al., 2015), multilayer networks (Bacco et al., 2017), temporal networks (Bu et al., 2019), hypergraphs and bipartite networks (Daminelli et al., 2015; Benson et al., 2018), networks with negative links (Leskovec et al., 2010; Tang et al., 2015), etc.) and networks with domain knowledge (e.g., drug-target interactions (Wu et al., 2018), disease-associated relations (Zeng et al., 2018), protein-protein interactions (Kovács et al., 2019; Lei and Ruan, 2013), criminal networks (Berlusconi et al., 2016), citation networks (Liu et al., 2019), academic social networks (Kong et al., 2019), knowledge graphs (Nickel et al., 2015), etc.). We should take serious consideration about properties and requirements of target networks and domains in the algorithm design, instead of straightforward (and thus less valuable) extensions of general algorithms. For example, if we attempt to recommend friends in an online social network based on link prediction (Aiello et al., 2012b), we need to consider how to explain our recommendations to improve the acceptance rate (Barbieri et al., 2014), how to use the acceptance/rejection information to promote the prediction accuracy (Wu et al., 2013), and how to avoid recommending bots to real users (Aiello et al., 2012a). These considerations will bring fresh challenges in link prediction.

Early studies often compare a very few algorithms on several small networks according to one or two metrics. Recent large-scale experiments (Mara et al., 2020; Ghasemian et al., 2020; Muscoloni et al., 2020; Muscoloni and Cannistraci, 2021; Zhou et al., 2021) indicated that the above methodology may result in misleading conclusions. Future studies ought to implement systematic analyses involving more synthetic and real networks, benchmarks, state-of-the-art algorithms, and metrics. Researchers can find benchmark datasets for networks from Open Graph Benchmark (OGB, ogb.stanford.edu), Pajek (mrvar.fdv.uni-lj.si/pajek), and Link Prediction Benchmarks (LPB, www.domedata.cn/LPB). If relevant results cannot be published in an article with limited space, they should be made public (better together with data and codes) in some accessible websites like GitHub, OGB, and LPB.

Lastly, we would like to emphasize that the soul of a network lies in its links instead of nodes; otherwise, we should pay more attention on set theory rather than graph theory. Therefore, in network science, link prediction is a paradigmatic and fundamental problem with long attractivity and vitality. Beyond an algorithm predicting missing and future links, link prediction is also a powerful analyzing tool, which has already

been utilized in evaluating and inferring network evolving mechanisms (Wang et al., 2012; Zhang et al., 2015; Zhang, 2017), testing the privacy-protection algorithms (as an attaching method) (Xian et al., 2021), evaluating and designing network embedding algorithms (Dehghan-Kooshkghazi et al., 2021; Gu et al., 2021), and so on. Though the last decade has witnessed plentiful and substantial achievements, the study of link prediction is just unfolding and more efforts are required toward a full picture of how links do emerge and vanish.

ACKNOWLEDGMENTS

I acknowledge Yan-Li Lee for valuable discussion and assistance. This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 11975071 and 61673086), the Science Strength Promotion Programmer of University of Electronic Science and Technology of China under Grant No. Y03111023901014006, and the Fundamental Research Funds for the Central Universities of China under Grant No. ZYGX2016J196.

AUTHOR CONTRIBUTIONS

T.Z. conceived and designed the project. T.Z. wrote the manuscript.

DECLARATION OF INTERESTS

The author declares no competing financial interests.

REFERENCES

Aiello, L.M., Deplano, M., Schifanella, R., and Ruffo, G. (2012a). People are strange when you're a stranger: Impact and influence of bots on social networks. In Proceedings of the international AAAI conference on web and social media (AAAI Press), pp. 10–17.

Aiello, L.M., Barrat, A., Schifanella, R., Cattuto, C., Markines, B., and Menczer, F. (2012b). Friendship prediction and homophily in social media. ACM Trans. Web *6*, 9.

Bacco, C. De, Power, E.A., Larremore, D.B., and Moore, C. (2017). Community detection, link prediction, and layer interdependence in multilayer networks. Phys. Rev. E *95*, 042317.

Barabasi, A.-L. (2016). Network Science (Cambridge University Press).

Barbieri, N., Bonchi, F., and Manco, G. (2014). Who to follow and why: link prediction with explanations. In Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 1266–1275.

Benson, A.R., Abebe, R., Schaub, M.T., Jadbabaie, A., and Kleinberg, J. (2018). Simplicial closure and higher-order link prediction. PNAS *115*, E11221–E11230.

Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., and Piccardi, C. (2016). Link prediction in criminal networks: a tool for criminal intelligence analysis. PLoS One *11*, e0154244.

Bliss, C.A., Frank, M.R., Danforth, C.M., and Dodds, P.S. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. J. Comput. Sci. *5*, 750–764.

Breiman, L. (1996). Bagging predictors. Mach. Learn. *24*, 123–140.

Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32.

Bu, Z., Wang, Y., Li, H.J., Jiang, J., Wu, Z., and Cao, J. (2019). Link prediction in temporal networks: integrating survival analysis and game theory. Inf. Sci. *498*, 41–61.

Cannistraci, C.V., Alanis-Lobato, G., and Ravasi, T. (2013a). From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Sci. Rep. *3*, 1613.

Cannistraci, C.V., Alanis-Lobato, G., and Ravasi, T. (2013b). Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. Bioinformatics *29*, i199.

Cao, S., Lu, W., and Xu, Q. (2016). Deep neural networks for learning graph representations. In Proceedings of the AAAI conference on artificial intelligence (AAAI Press), pp. 1145–1152.

Chen, X., Wang, L., Qu, J., Guan, N.N., and Li, J.Q. (2018). Predicting miRNA-disease association based on inductive matrix completion. Bioinformatics *34*, 4256–4265.

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

Comar, P.M., Tan, P.N., and Jain, A.K. (2011). Linkboost: a novel cost-sensitive boosting framework for community-level network link prediction. In Proceedings of the 11th international conference on data mining (IEEE Press), pp. 131–140.

Csermely, P., Korcsmáros, T., Kiss, H.J., London, G., and Nussinov, R. (2013). Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. Pharmacol. Ther. *138*, 333–408.

Cui, P., Wang, X., Pei, J., and Zhu, W. (2018). A survey on network embedding. IEEE Trans. Knowl. Data Eng. *31*, 833–852.

Daminelli, D., Thomas, J.M., Durán, C., and Cannistraci, C.V. (2015). Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. New J. Phys. *17*, 113037.

Davis, J., and Goadrich, M. (2006). The relationship between precision–recall and ROC curves. In Proceedings of the 23rd international conference on machine learning (ACM Press)), pp. 233–240.

Dehghan-Kooshkghazi, A., Kamiński, B., Prałat, Ł., and Théberge, F. (2021). Evaluating Node embeddings of complex networks. arXiv: 2102.08275.

Ding, H., Takigawa, I., Mamitsuka, H., and Zhu, S. (2014). Similarity-based machine learning methods for predicting drug–target interactions: a brief review. Brief. Bioinform. *15*, 734–747.

Duan, L., Ma, S., Aggarwal, C., Ma, T., and Huai, J. (2017). An ensemble approach to link prediction. IEEE Trans. Knowl. Data Eng. *29*, 2402–2416.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognit. Lett. *27*, 861–874.

Fire, M., Tenenboim-Chekina, L., Puzis, R., Lesser, O., Rokach, L., and Elovici, Y. (2013). Computationally efficient link prediction in a variety of social networks. ACM Trans. Intell. Syst. Technol. *5*, 10.

Freund, Y., and Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. *55*, 119–139.

Garcia-Perez, G., Aliakbarisani, R., Ghasemi, A., and Serrano, M.A. (2020). Precision as a measure of predictability of missing links in real networks. Phys. Rev. E *101*, 052318.

Ghasemian, A., Galstyan, A., Airoldi, E.M., and Clauset, A. (2020). Stacking models for nearly optimal link prediction in complex networks. PNAS *117*, 23393–23400.

Grover, A., and Leskovec, J. (2016). node2vec: scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 855–864.

Gu, J., Lee, S., Saramäki, J., and Holme, P. (2017). Ranking influential spreaders is an ill-defined problem. EPL *118*, 68002.

Gu, W., Gao, F., Li, R., and Zhang, J. (2021). Learning universal network representation via link prediction by graph convolutional neural network. J. Soc. Comput. *2*, 43–51.

Guimerà, R., and Sales-Pardo, M. (2009). Missing and spurious interactions and the reconstruction of complex networks. PNAS *106*, 22073–22078.

Guimerà, R. (2020). One model to rule them all in network science. PNAS *117*, 25195–25197.

Hand, D.J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach. Learn. *77*, 103–123.

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

He, Y.-L., Liu, J.N.K., Hu, Y.-X., and Wang, X.-Z. (2015). OWA operator based link prediction ensemble for social network. Expert Syst. Appl. *42*, 21–50.

Holme, P., Liljeros, F., Edling, C.R., and Kim, B.J. (2003). Network bipartivity. Phys. Rev. E *68*, 056107.

Holme, P., and Saramäki, J. (2012). Temporal networks. Phys. Rep. *519*, 97–125.

Jain, P., and Dhillon, I.S. (2013). Provable inductive matrix completion. arXiv: 1306.0626.

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

Kitsak, M., Voitalov, I., and Krioukov, D. (2020). Link prediction with hyperbolic geometry. Phys. Rev. Res. *2*, 043113.

Kong, X., Shi, Y., Yu, S., Liu, J., and Xia, F. (2019). Academic social networks: modeling, analysis, mining and applications. J. Netw. Comput. Appl. *132*, 86–103.

Kontoyiannis, I., Algoet, P.H., Suhov, Y.M., and Wyner, A.J. (1998). Nonparametric entropy estimation for stationary processes and random fields, with applications to English text. IEEE Trans. Inf. Theor. *44*, 1319–1327.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer *40*, 30–37.

Koutra, D., Kang, U., Vreeken, J., and Faloutsos, C. (2015). Summarizing and understanding large graphs. Stat. Anal. Data Mining *8*, 183–202.

Kovács, I.A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.K., Kishore, N., Hao, T., et al. (2019). Network-based prediction of protein interactions. Nat. Commun. *10*, 1240.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguna, M. (2010). Hyperbolic geometry of complex networks. Phys. Rev. E *82*, 036106.

Kumar, A., Singh, S.S., Singh, K., and Biswas, B. (2020). Link prediction techniques, applications, and performance: a survey. Physica A *553*, 124289.

Lee, Y.-L., and Zhou, T. (2021). Collaborative filtering approach to link prediction. Physica A *578*, 126107.

Lee, Y.-L., Dong, Q., and Zhou, T. (2021). Link prediction via controlling the leading eigenvector. Appl. Math. Comput. *411*, 126517.

Lei, C., and Ruan, J. (2013). A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. Bioinformatics *29*, 355–364.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In Proceedings of the 19th international conference on world wide web (ACM Press), pp. 641–650.

Li, K., Tu, L., and Chai, L. (2020). Ensemble-model-based link prediction of complex networks. Comput. Netw. *166*, 106978.

Liben-Nowell, D., and Kleinberg, J. (2007). The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. *58*, 1019–1031.

Lichtenwalter, R.N., Lussier, J.T., and Chawla, N.V. (2010). New perspectives and methods in link prediction. In Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 243–252.

Lichtenwalter, R.N., and Chawla, N.V. (2012). Link prediction: fair and effective evaluation. In Proceedings of the 2012 IEEE/ACM international conference on advances in social networks analysis and mining (IEEE Press), pp. 376–383.

Liu, H., Kou, H., Yan, C., and Qi, L. (2019). Link prediction in paper citation network to construct paper correlation graph. EURASIP J. Wirel. Commun. Netw. *2019*, 233.

Lu, C., Yang, M., Luo, F., Wu, F.-X., Li, M., Pan, Y., Li, Y., and Wang, J. (2018). Prediction of lncRNA-disease associations based on inductive matrix completion. Bioinformatics *34*, 3357–3364.

Lü, L., Jin, C.-H., and Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. Phys. Rev. E *80*, 046122.

Lü, L., and Zhou, T. (2011). Link prediction in complex networks: a survey. Physica A *390*, 1150–1170.

Lü, L., Medo, M., Yeung, C.-H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. Phys. Rep. *519*, 1–49.

Lü, L., Pan, L., Zhou, T., Zhang, Y.-C., and Stanley, H.E. (2015). Toward link predictability of complex networks. PNAS *112*, 2325–2330.

Ma, X., Sun, P., and Wang, Y. (2018). Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks. Physica A *496*, 121–136.

Mara, A.C., Lijffijt, J., and Bie, T. De (2020). Benchmarking network embedding models for link prediction: are We making progress? In Proceedings of the 7th IEEE international conference on data science and advanced analytics (IEEE Press)), pp. 138–147.

Martínez, V., Berzal, F., and Cubero, J.C. (2016). A survey of link prediction in complex networks. ACM Comput. Surv. *49*, 69.

Menon, A.K., and Elkan, C. (2011). Link prediction via matrix factorization. In Proceedings of the joint European conference on machine learning and knowledge discovery in databases (Springer), pp. 437–452.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv: 1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. *26*, 3111–3119.

Muscoloni, A., Thomas, J.M., Ciucci, S., Bianconi, G., and Cannistraci, C.V. (2017). Machine learning meets complex networks via coalescent embedding in the hyperbolic space. Nat. Commun. *8*, 1615.

Muscoloni, A., Abdelhamid, I., and Cannistraci, C.V. (2018). Local-community network automata modeling based on length-three-paths for prediction of complex network structures in protein interactomes, food web and more. bioRxiv. https://doi.org/10.1101/346916.

Muscoloni, A., and Cannistraci, C.V. (2018a). A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. New J. Phys. *20*, 052002.

Muscoloni, A., and Cannistraci, C.V. (2018b). Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction. New J. Phys. *20*, 063022.

Muscoloni, A., and Cannistraci, C.V. (2018c). Minimum curvilinear automata with similarity attachment for network embedding and link prediction in the hyperbolic space. arXiv: 1802.01183..

Muscoloni, A., Michieli, U., and Cannistraci, C.V. (2020). Adaptive network automata modeling of complex networks. Preprint: 202012.0808.

Muscoloni, A., and Cannistraci, C.V. (2021) Short note on comparing stacking modelling versus Cannistraci-Hebb adaptive network automata for link prediction in complex networks. Preprints: 202105.0689.

Natarajan, N., and Dhillon, I.S. (2014). Inductive matrix completion for predicting gene-disease associations. Bioinformatics *30*, i60.

Newman, M.E.J. (2018). Networks (Oxford University Press).

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs. Proc. IEEE *104*, 11–33.

Nicolaou, Z.G., and Motter, A.E. (2020). Missing links as a source of seemingly variable constants in complex reaction networks. Phys. Rev. Res. *2*, 043135.

Ou, M., Cui, P., Zhang, J., and Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM

SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 1105–1114.

Pan, L., Zhou, T., Lü, L., and Hu, C.-K. (2016). Predicting missing links and identifying spurious links via likelihood analysis. Sci. Rep. *6*, 22955.

Papadopoulos, F., Kitsak, M., Serrano, M.A., Boguna, M., and Krioukov, D. (2012). Popularity versus similarity in growing networks. Nature *489*, 537–540.

Papadopoulos, F., Psomas, C., and Krioukov, D. (2014). Network mapping by replying hyperbolic growth. IEEE/ACM Trans. Netw. *23*, 198–211.

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

Pech, R., Hao, D., Lee, Y.-L., Yuan, Y., and Zhou, T. (2019). Link prediction via linear optimization. Physica A *528*, 121319.

Peel, L., Larremore, D.B., and Clauset, A. (2017). The ground truth about metadata and community detection in networks. Sci. Adv. *3*, e1602548.

Peixoto, T.P. (2018). Reconstructing networks with unknown and heterogeneous errors. Phys. Rev. X *8*, 041011.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 701–710.

Provost, F., Fawcett, T., and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In Proceedings of the 15th international conference on machine learning (Morgan Kaufmann Publishers), pp. 445–453.

Pujari, M., and Kanawati, R. (2012). Supervised rank aggregation approach for link prediction in complex networks. In Proceedings of the 21st international conference on world wide web (ACM Press), pp. 1189–1196.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, C., Wang, K., and Tang, T. (2019). NetSMF: large-scale network embedding as sparse matrix factorization. In Proceedings of the world wide web conference (ACM Press), pp. 1509–1520.

Roweis, S.T., and Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding. Science *290*, 2323–2326.

Saito, T., and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One *10*, e0118432.

Shang, M.-S., Lü, L., Zhang, Y.-C., and Zhou, T. (2010). Empirical analysis of web-based user-object bipartite networks. EPL *90*, 48006.

Shen, Z., Wang, W.-X., Fan, Y., Di, Z., and Lai, Y.-C. (2014). Reconstructing propagation networks with natural diversity and identifying hidden sources. Nat. Commun. *5*, 4323.

Sherkat, E., Rahgozar, M., and Asadpour, M. (2015). Structural link prediction based on ant colony approach in social networks. Physica A *419*, 80–94.

Song, C., Qu, Z., Blumm, N., and Barabási, A.-L. (2010). Limits of predictability in human mobility. Science *327*, 1018–1021.

Squartini, T., Caldarelli, G., Cimini, G., Gabrielli, A., and Garlaschelli, D. (2018). Reconstruction methods for networks: the case of economic and financial systems. Phys. Rep. *757*, 1–47.

Swamidass, S.J., Azencott, C.A., Daily, K., and Baldi, P.A. (2010). CROC stronger than ROC: measuring, visualizing and optimizing early retrieval. Bioinformatics *26*, 1348–1356.

Sun, J., Feng, L., Xie, J., Ma, X., Wang, D., and Hu, Y. (2020). Revealing the predictability of intrinsic structure in complex networks. Nat. Commun. *11*, 574.

Tang, J., Chang, S., Aggarwal, C., and Liu, H. (2015). Negative link prediction in social media. In Proceedings of the 8th ACM international conference on web search and data mining (ACM Press), pp. 87–96.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2016). LINE: large-scale information network embedding. In Proceedings of the 24th international conference on world wide web (ACM Press), pp. 1067–1077.

Tang, D., Du, W., Shekhtman, L., Wang, Y., Havlin, S., Cao, X., and Yan, G. (2020). Predictability of real temporal networks. Natl. Sci. Rev. *7*, 929–937.

Tenenbaum, J.B., Silva, V. De, and Langford, J.C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science *290*, 2319–2323.

Valles-Catala, T., Peixoto, T.P., Sales-Pardo, M., and Guimera, R. (2018). Consistencies and inconsistencies between model selection and link prediction in networks. Phys. Rev. E *97*, 062316.

Wang, W.-Q., Zhang, Q.-M., and Zhou, T. (2012). Evaluating network models: a likelihood analysis. EPL *98*, 28004.

Wang, Y., Wang, L., Li, Y., He, D., Chen, W., and Liu, T.Y. (2013). A theoretical analysis of NDCG ranking measures. In Proceedings of the 26th annual conference on learning theory (COLT Press), pp. 25–54.

Wang, P., Xu, B., Wu, Y., and Zhou, X. (2015). Link prediction in social networks: the state-of-the-art. Sci. China Inf. Sci. *58*, 1–38.

Wang, Q., Gao, J., Zhou, T., Hu, Z., and Tian, H. (2016a). Critical size of ego communication networks. EPL *114*, 58004.

Wang, D., Cui, P., and Zhu, W. (2016b). Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (ACM Press), pp. 1225–1234.

Wang, Z., Wu, Y., Li, Q., Jin, F., and Xiong, W. (2016c). Link prediction based on hyperbolic mapping with community structure for complex networks. Physica A *450*, 609–623.

Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, M., Zhang, F., Xie, X., and Guo, M. (2018). GraphGAN: graph representation learning with generative adversarial nets. In Proceedings of the AAAI conference on artificial intelligence (AAAI Press), pp. 2508–2515.

Wang, J., Zhang, Q.-M., and Zhou, T. (2019). Tag-aware link prediction algorithm in complex networks. Physica A *523*, 105–111.

Wang, X.-W., Chen, Y., and Liu, Y.-Y. (2020). Link prediction through deep generative model. iScience *23*, 101626.

Wolpert, D.H. (1992). Stacked generalization. Neural Netw. *5*, 241–259.

Wu, S., Sun, J., and Tang, J. (2013). Patent partner recommendation in enterprise social networks. In Proceedings of the 6th ACM international conference on web search and data mining (ACM Press), pp. 43–52.

Wu, Z., Li, W., Liu, G., and Tang, Y. (2018). Network-based methods for prediction of drug-target interactions. Front. Pharmacol. *9*, 1134.

Wu, M., Wu, S., Zhang, Q., Xue, C., Kan, H., and Shao, F. (2019). Enhancing link prediction via network reconstruction. Physica A *534*, 122346.

Xian, X., Wu, T., Qiao, S., Wang, X.-Z., Wang, W., and Liu, Y. (2020). NetSRE: link predictability measuring and regulating. Knowl.-Based Syst. *196*, 105800.

Xian, X., Wu, T., Liu, Y., Wang, W., Wang, C., Xu, G., and Xiao, Y. (2021). Towards link inference attack against network structure perturbation. Knowl.-Based Syst. *218*, 106674.

Xu, P., Yin, L., Yue, Z., and Zhou, T. (2019). On predictability of time series. Physica A *523*, 345–351.

Yager, R.R. (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. Syst. Man Cybern. *18*, 183–190.

Yang, Y., Lichtenwalter, R.N., and Chawla, N.V. (2015). Evaluating link prediction methods. Knowl. Inf. Syst. *45*, 751–782.

Zeng, X., Liu, L., Lü, L., and Zou, Q. (2018). Prediction of potential disease-associated microRNAs using structural perturbation method. Bioinformatics *34*, 2425–2432.

Zhang, Z.-K., Zhou, T., and Zhang, Y.-C. (2011). Tag-aware recommender systems: a state-of-the-art survey. J. Comput. Sci. Technol. *26*, 767–777.

Zhang, Q.-M., Lü, L., Wang, W.-Q., and Zhou, T. (2013). Potential theory for directed networks. PLoS One *8*, e55437.

Zhang, Q.-M., Xu, X.-K., Zhu, Y.-X., and Zhou, T. (2015). Measuring multiple evolution mechanisms of complex networks. Sci. Rep. *5*, 10350.

Zhang, J. (2017). Uncovering mechanisms of co-authorship evolution by multirelations-based link prediction. Inf. Process. Manag. *53*, 42–51.

Zhang, M., and Chen, Y. (2017). Weisfeiler-Lehman neural machine for link prediction. In Proceedings of the 23rd ACM SIGKDD

international conference on knowledge discovery and data mining (ACM Press), pp. 575–583.

Zhang, M., and Chen, Y. (2018). Link prediction based on graph neural networks. In Proceedings of the 32nd international conference on neural information processing systems (ACM Press), pp. 5171–5181.

Zhang, Q., Tong, T., and Wu, S. (2020). Hybrid link prediction via model averaging. Physica A *556*, 124772.

Zhao, J., Miao, L., Yang, J., Fang, H., Zhang, Q.-M., Nie, M., Holme, P., and Zhou, T. (2015). Prediction of links and weights in networks by reliable routes. Sci. Rep. *5*, 12261.

Zhou, T., Lü, L., and Zhang, Y.-C. (2009). Predicting missing links via local information. Eur. Phys. J. B *71*, 623–630.

Zhou, T., Lee, Y.-L., and Wang, G. (2021). Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. Physica A *564*, 125532.

Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms (CRC Press).

Zhu, Y.-X., Lü, L., Zhang, Q.-M., and Zhou, T. (2012). Uncovering missing links with cold ends. Physica A *391*, 5769–5778.