Review on Graph Feature Learning and Feature Extraction Techniques for Link Prediction

Ece C. Mutlu
ecemutlu@knights.ucf.edu
University of Central Florida
Department of Industrial Engineering

Toktam A. Oghaz
toktam@cs.ucf.edu
University of Central Florida
Department of Computer Science

ABSTRACT

Studying networks to predict the emerging interactions is a common research problem for both fields of network science and machine learning. The problem of predicting future or missing relationships in networks is called link prediction. Machine learning studies have mostly approached to this problem as a clustering or a classification task. A few obstacles might be involved in approaching network datasets through machine learning models, including undefined euclidean distance, extracting proper features, unbalanced data classes due to the sparsity of real networks, or embedding graphs to a low dimensional vector space while preserving the structure to study networks. Extensive studies have examined these problems from different aspects and proposed methods some of which might work very well for a specific application but not as a global solution. In this survey, we review the general-purpose techniques at the heart of link prediction problem, which can be combined with domain-specific heuristic methods in practice. To the best of our knowledge, this survey is the first comprehensive study which considers all of the mentioned challenges about studying networks and approaching them through machine learning models. We provide a diverse study on feature extraction techniques for network datasets based on similarity metrics, maximum likelihood methods, probabilistic methods and graph representation learning. Our other contributions include proposing a taxonomy to classify link prediction methods and continue with introducing valuable network dataset collections to study the problem of link prediction. Our final contribution is discussing and proposing a few models, including a multi-stream feature learning model to exploit the benefits of local and quasi-local network extraction techniques combined with graph representation learning.

KEYWORDS

Networks, link prediction, machine learning, knowledge learning, similarity metrics

ACM Reference Format:
Ece C. Mutlu and Toktam A. Oghaz. 2018. Review on Graph Feature Learning and Feature Extraction Techniques for Link Prediction. In *Proceedings of* . ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/1122445.1122456

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Complex networks have been extensively studied in the context of understanding diffusion of information in social networks, interactions between people, structural similarity of proteins and architecture of business relations between people, corporations or countries etc. This emerging "connectedness" fascinated researchers to investigate complex networks thoroughly. Social networks, as we are all familiar with, may be a prime example of complex networks. Social networks are constructed by putting together human-human interactions irrespective of their regional distances, different cultures, and even sometimes different languages. The use of social networks facilitates receiving news from around the world, communicating with friends, following scientific developments, and more. Another example of a complex network is the information network, which is also called a "knowledge network" [82] and has a similar structure to social networks. The most common example for information networks is the citation network, in which authors are connected via their scientific publications and co-citation interactions [35]. Biological networks, on the other hand, might provide another example for complex networks, which represent protein-protein interactions, metabolic pathways or genetic interactions between organisms. Individuals and the different relations between them in a network structure can simply act as a graph composed of a set of nodes (vertices) and edges (links). Such graphs can be defined as G = $\langle V, E\rangle$ where V is the set of vertices and E is the set of edges in the graph [47]. For a dynamic graph of complex networks, the set of vertices and edges change over time as new users are introduced to the network and new links emerge through new connections. Graphs of complex networks might contain a substantial number of communities in which each strong and dense interconnections help to distinguish communities while they are themselves connected through weak connections [13].

To provide a few visualization examples for complex networks, Figure 1a demonstrates the well-known network of Zachary's karate club. This figure shows the connectedness of 34 members of the karate club interacting outside the club context [118], and is colored based on the connections of two central people (members 1 and 34). The matrix is formed by linking associations between the nodes, and it is called "adjacency matrix" (Figure 1b). This matrix provides information about whether links exist between the Zachary's karate club members. In Figure 1b, the links are displayed by the color yellow, whereas blue colored area illustrates the nonexistence of links between members. Since this network is very sparse and small, one can easily follow all the connections between the individuals. To exemplify the visualization of a dense graph of social networks, Figure 2 displays one of the ego-network structures of the SNAP Facebook dataset [63]. As can be observed in this figure, the colors either refer to the number of connections in the network, known as the "degree" (Figure 2a), or "closeness" (Figure 2b) which is measured by taking the shortest paths.

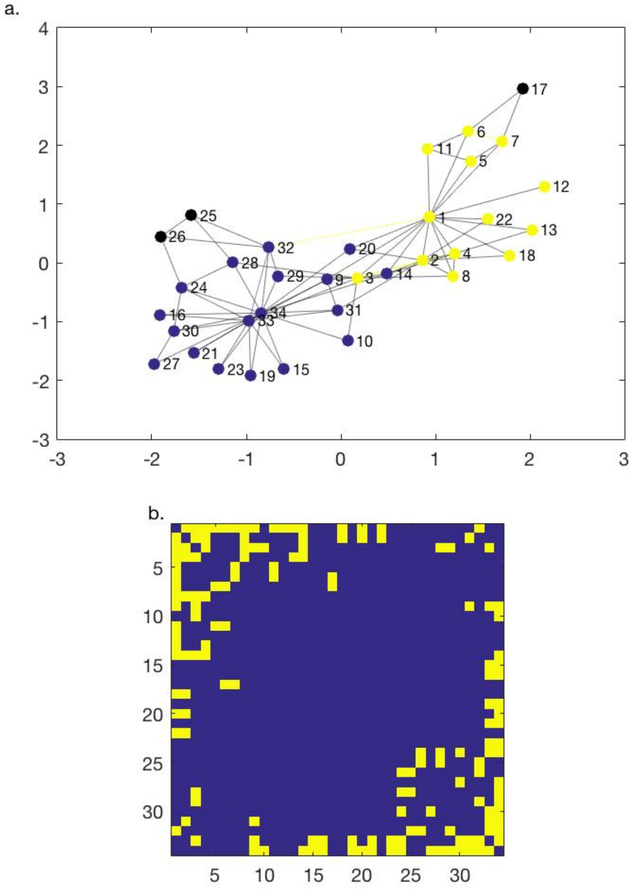

Figure 1: Two visualizations for the Zachary's karate club data. a. The visualization of the relationships between 34 members in the karate club outside of the club (for two central nodes of numbers 1 and 34), b. The adjacency matrix of the relationships between members of the karate club. Links are colored by yellow.

The oldest studies in network science are based on the random graphs [37] proposed by Erdős and Rényi, in which n edges are connected randomly out of $n(n-1)/2$ number of possible edges with probability p. An extensive effort has been made on random graphs, which demonstrate the common properties of the networks and their probabilistic distributions, and it fueled novel research ideas for the future studies for a long time. [6, 17, 36, 41, 56]. Later studies shifted their focus to real networks (not generated randomly) and explained their formation and evolution. Studies on computational network analysis mainly comprise of statistical analyses of complex networks [28, 77, 92], community detection and node classification [38, 61, 89], evolution of the dynamics of networks over time [31, 32, 58, 111], information diffusion and cascade analyses [9, 42, 97, 116], data mining [29, 96, 103] and graph visualization [18, 24, 78, 112] etc. One of the most interesting and long-standing challenges is the problem of link prediction in complex networks. This challenge aims to make inferences about the existing links between the nodes, understand the structure and the formation of the networks to predict the not-yet-existing connections between the pairs of entities. Link prediction applications include online recommendation systems, route recommendations based on traffic patterns, and patterns of disease epidemics, and the diffusion of information in complex networks [66, 75].

One of the main obstacles in the challenge of link prediction is that there is a trade-off between the amount of information (nodes, links, and features) to be analyzed and the complexity of the method used for the analysis. This problem becomes apparent especially when studying real-world networks containing thousands of nodes and interactions [75]. Furthermore, network datasets pose the problem of an imbalance resulting from the sparsity of networks [75].

Link prediction have been mostly investigated through unsupervised graph representation and feature learning methods based on node (local) or path (global) similarity metrics calculated for neighboring nodes. However, the task of link prediction can also be overcome by the use of supervised machine learning algorithms. Machine learning models for the task of link prediction might i) exploit the similarity metrics as the input features ii) embed the nodes into a low dimensional vector space while preserving the topological structure of the graph iii) combine the information derived from i or ii with the node attributes available from the dataset. Link prediction models rely on the hypothesis that nodes with more similarities are more likely to connect [75]. Graph feature learning techniques, on the other hand, include the examination of graph topology and structural features to calculate score functions based on pairwise similarity metrics. Common neighbors, preferential attachment, Jaccard, Katz and Adamic Adar are some of the widely used similarity metrics which measure the likelihoods of edge associations in graphs. While these methods may seem dated, they are far from being obsolete. Despite the fact that they do not discover the attributes for graphs, they have remained popular for years due to being simple, interpretable and scalable [121]. These methods provide the features on which machine learning models can learn upon.

The rest of the paper is structured as follows. First, we look into the preliminaries and describe the problem, then we list our contributions in this survey. In section 2 a review over techniques for similarity metrics and their definitions is provided. We present maximum likelihood techniques for link prediction in section 3 and continue with probabilistic methods in section 4. Section 5 devotes to graph embedding methods and representation learning. A discussion over a few supervised link prediction models is available in section 6. Section 7 includes literature combining multi sources for the problem of link prediction. In section 8, a few network datasets are introduced. Finally, in section 9 we discuss and review some methods and propose distinct models for future study. The appendix includes the diagram for the proposed taxonomy and supplementary materials.

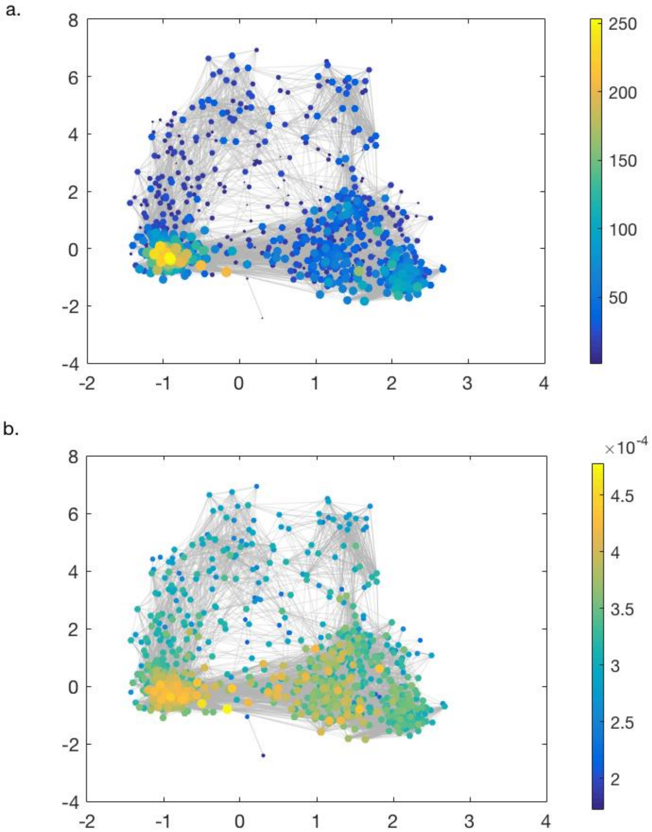

Figure 2: Ego network of node 107 of SNAP Facebook dataset colored with respect to a. degree centrality b. closeness centrality

1.1 Preliminaries and Problem Description

A complex network (graph) $G = \langle V, E \rangle$ can be defined as the set of entities called "nodes (vertices)" V and interaction between pairs of entities called "edges (links)" V at a particular time t. The main idea behind applying the feature extraction or learning methods onto link prediction problem is the use present information of the existing edges to predict future or missing edges which will emerge at time $t^{'} > t$. Although some discussed methods in next sections can solve the link prediction problem of directed graphs mainly, most methods we considered in this survey address the problem of link prediction for undirected graphs. The difference between the link prediction problem for directed and undirected graphs arises from the additional information required for directed ones. This information refers to the origin of the associated link for directed graphs in which $\langle x, y \rangle$ means that the relation is directed from x to y. However, in undirected graphs the edges have no orientation and the relation between them is reciprocal. The important feature of undirected graphs is the identity of the pair orders $\langle x, y \rangle$ and $\langle y, x \rangle$ [110]. It should be noted that self interactions of nodes are not allowed and not taken into account [73]. Since the connections between nodes are the main concern of link prediction approaches, the other set of nodes connected to node $x \in V$ is called as the "neighbors of node x" and denoted as $\Gamma(x) \in V$. While $\Gamma(x)$ stands for the neighbor nodes connected to the node x, the number of links or edges connected to the node is represented by $|\Gamma(x)|$. Different link prediction algorithms necessitate training and test sets to compare their performances like every machine learning ; however, one cannot know the future links at $t^{'}$ by considering the current graph structure. Therefore, a fraction of links from the current network structure is deleted (Figure 3.c), and taken as the test set (true positive), whereas the remaining fraction of edges in the network is used as the training set. A reliable link prediction approach should provide higher probabilities for the edges belong to the set of true positives comparing to the set of nonexistent edges (true negatives) [109].

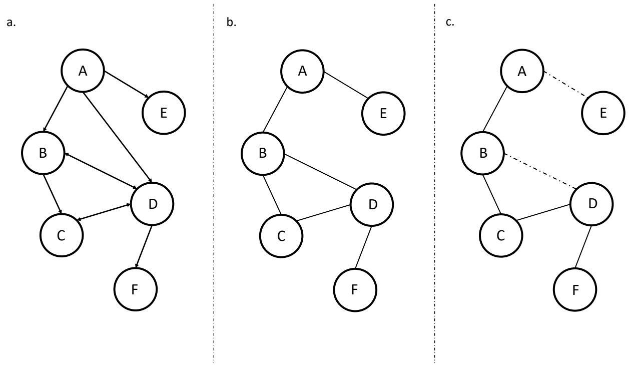

Figure 3: Imaginary representation of a. directed whole graph b. undirected whole graph c. undirected training graph

1.2 Summary of Contributions

Our contribution in this survey includes, firstly, presenting a comprehensive study on diverse and thorough link prediction techniques which is not provided in other surveys as a whole. Since the solution of link prediction problem is not a trivial task, a number of methods were derived all different from each other in terms of their philosophy. Although survey studies up to now provide a large spectrum of solutions, they do not consider unsupervised/supervised feature extraction techniques, feature based/graph based methods and learning models simultaneously. We present our work by providing an overall review on feature extraction techniques based on similarity metrics, maximum likelihood methods, probabilistic methods, and graph representation learning models.

Another contribution of this survey is proposing a general taxonomy for the methods studying the problem of link prediction. A scheme of the proposed taxonomy is displayed in Figure 4. The methods are categorized in accordance with their approaches to extract the network features. We split the available models into two categories of Feature Extraction Methods and Feature Learning Methods. The first branch consists of models studying network features through Similarity Based Methods, Likelihood Based Methods, and Probabilistic Methods. The second branch contains techniques

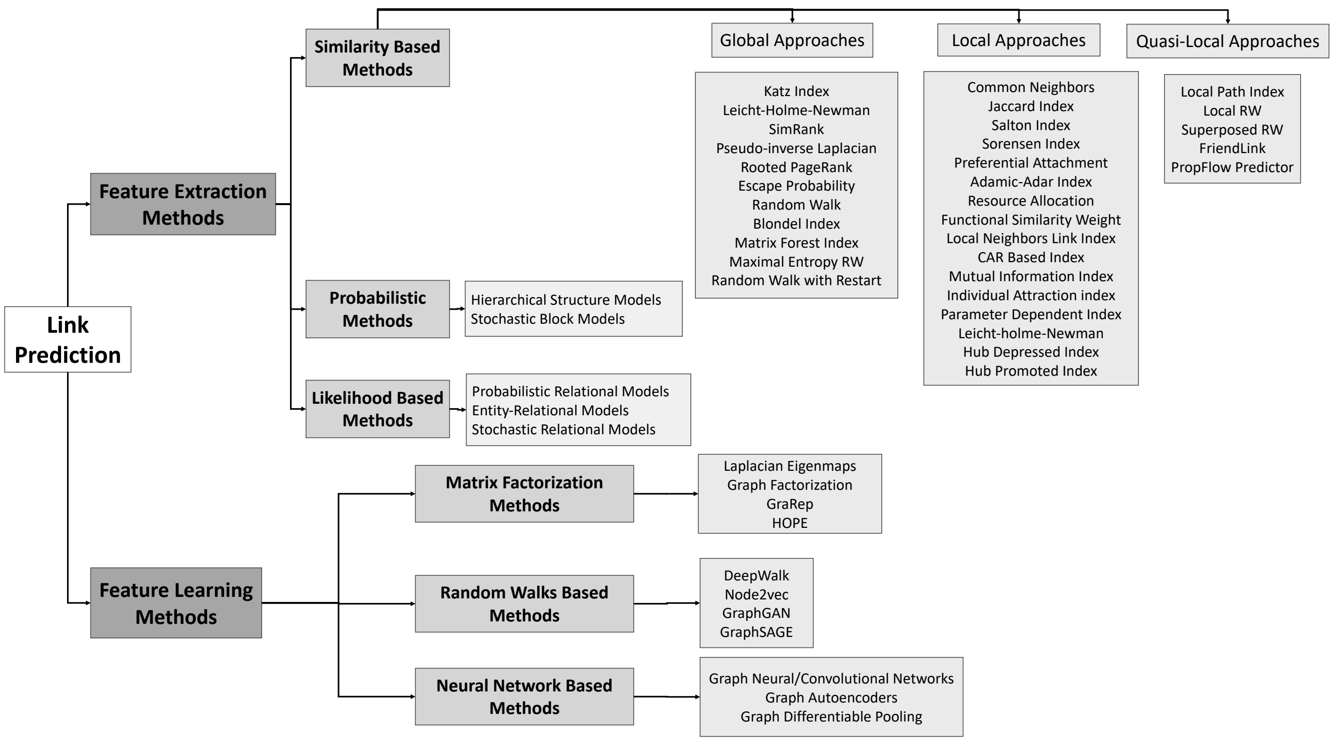

Figure 4: The proposed taxonomy for the feature extraction techniques and feature learning methods for link prediction studies.

which apply a node embedding to the graph and learn the graph representations. These models include Matrix Factorization Methods, Random Walk Based Methods, and Neural Network Based Methods. All of the studied techniques provide features which can be fed to supervised/unsupervised machine learning algorithms to approach the link prediction problem through a clustering or a classification task.

2 SIMILARITY BASED METHODS

The most generic framework used for link prediction purposes is computing the similarity between nodes based on very simple idea: if two nodes are more similar, they are more likely to be linked in the future. Based on this hypothesis similarity between unconnected node pairs (x and y) are assigned $s_{(x,y)}$ and ranked; not-yet-existing links can be predicted with high similarity score. These methods use the topology structure of the network by considering specific node pairs. Based on the systematic structure of the metrics, they can be investigated under three categories: global, local and quasi-local approaches.

For clarity of the further explanations, let us give common notations used in these approaches. Suppose that lowercase letters denote nodes; x and y are the main nodes we are trying to assign a similarity score between them. $\Gamma(x)$ is used to define the set of neighbors of node x and $|\Gamma(x)|$ is for the number of neighbors of node x, i.e. $z \in \Gamma(x) \cap \Gamma(y)$ mean z is either the neighbor to node x or node y. Let A represents adjacency matrix and $|E|$ the number of edges in the network.

2.1 Local Similarity Based Approaches

Local similarity based approaches are based on either the idea that if node pairs have common neighbors structures or one of them already has a significantly higher degree, they will probably form a link in the future. Because they use only local topological information based on neighborhood-related structures rather than considering the whole network structure, they are faster than the global similarity based approaches. Many studies showed also their superior performance especially on the dynamic networks [68]; however, they are restricted to compute the similarity of the all possible combination of the node pairs since they only rank similarity between close nodes which have distance less than two.

2.1.1 Common Neighbors (CN). : CN is one of the most widespread information retrieval metric for link prediction task due to its high efficiency in spite of its simplicity. The idea behind CN is that the probability of being linked for two nodes in the future is affected by the number of their common nodes, i.e. two nodes will highly probably establish a link if they have more shared nodes. The score of this metric can be defined as follows:

$$s^{CN}_{(x,y)} = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

It should be noted that, resulting score using CN is not normalized, and only shows the relative similarity of different node-pairs by considering shared nodes between them. Newman used CN to show that the probability of collaboration between two scientists in the future can be estimated by their previous common collaborators [81].

2.1.2 *Jaccard Index (JC).*

: The metric not only takes number of common nodes into account as in CN, but also normalizes it by considering the total set of number of shared and non-shared neighbors. The equation of this score proposed by Jaccard [53] is:

$$s^{JC}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2}$$

2.1.3 *Salton Index (SL).*

: SI is the metric which is known as cosine similarity in the literature. It is simply the ratio of number of shared neighbors of x and y to the square root of inner-product of their degrees [91].

$$s^{SL}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)|.|\Gamma(y)|}} \tag{3}$$

Wagner & Leydesdorff [106] showed that SI is an efficient metric when constructional pattern of relations are aimed to be visualized.

2.1.4 *Sørensen Index (Sø).*

: The index very similar to JI is generated to make a comparison between different ecological samples [94].

$$s^{S}_{(x,y)} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|} \tag{4}$$

The difference of using summation of the degrees instead of degrees of their union makes SøI less outlier sensitive when it is compared with Jaccard Index [76].

2.1.5 *Preferential Attachment Index (PA).*

: This metric is the result of the study of Barabasi & Albert in which new nodes joining to the network are proved to be connected highly probably with an existing node that has higher degrees rather than a node that has lower degrees [10].

$$s^{PA}_{(x,y)} = |\Gamma(x)|.|\Gamma(y)| \tag{5}$$

2.1.6 *Adamic-Adar Index (AA).*

: The metric AA is motivated by the necessity of the comparison of two web-pages by Lada Adamic and Eytan Adar [4], is simply use the idea of giving more weight to the relatively fewer common neighbors.

$$s^{AA}_{(x,y)} = \sum_{z \in |\Gamma(x) \cap \Gamma(y)|} \frac{1}{log|\Gamma(z)|} \tag{6}$$

Although it has similarities between CN, the important difference is that shared neighbors of two nodes are penalized by the logarithm of their degrees. It should be noted that while the other metrics include the two nodes (x and y) and/or their degrees in their equations, AA relates common neighbors (z) to these two nodes (x and y).

2.1.7 *Resource Allocation Index (RA).*

: Motivated by physical process of resource allocation, a very similar metric to AA is developed by Zhou et al. [122].

$$s^{RA}_{(x,y)} = \sum_{z \in |\Gamma(x) \cap \Gamma(y)|} \frac{1}{|\Gamma(z)|} \tag{7}$$

The difference in the denominator k_z^{-1} of RA rather than $logk_z{}^{-1}$ in AA penalizes the contribution of common neighbors more. Many studies show that this discrepancy is very insignificant, and resulting performances of these two metrics are very similar when the average degree of the network is low; however, RA is more superior when the average degree is high [108].

2.1.8 *Hub Promoted Index (HP).*

: The index is proposed for assessing the similarity of the substrates in metabolic networks [88] and defined as following formula:

$$s^{HP}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{min(|\Gamma(x)|, |\Gamma(y)|)} \tag{8}$$

HPI value is simply determined by the ratio of common neighbors of both x and y to the minimum of degrees of nodes of x and y. Here, link formation between lower degree nodes and the hubs is more promoted while the formation of the connection between hub nodes are prevented [75].

2.1.9 *Hub Depressed Index (HD).*

: The totally opposite analogy is also considered by Lü and Zhou [73]. Here, link formation between lower degree nodes and that between hubs are promoted; however, connection beween hub nodes and lower degree nodes are prevented.

$$s^{HD}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{max(|\Gamma(x)|, |\Gamma(y)|)} \tag{9}$$

2.1.10 *Leicht-Holme-Newman Index (LHN1).*

: The index, very similar to SI, is defined as the ratio of number of shared neighbors of x and y to the product of their degrees, the latter is the expected value of the number of paths of length between them [62].

$$s^{LHN1}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|.|\Gamma(y)|} \tag{10}$$

The difference in denominator shows that SI always assigns higher score than LHNI.

2.1.11 *Parameter Dependent Index (PD).*

: Zhou et al. proposed a new metric to improve the prediction accuracy for not only popular links but also unpopular links. PD can be defined as:

$$s^{PD}_{(x,y)} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|.|\Gamma(y)|^{\lambda}} \tag{11}$$

where, λ is free parameter [123]. One can easily recognizes that PD is degraded to CN, SL and LHN1 when $\lambda = 0$ $\lambda = 0.5$ and $\lambda = 1$, respectively.

2.1.12 *The Individual Attraction Index (IA).*

: Dong et al. [30] proposed an index which relates not only the common neighbors of the nodes individually but also the effect of the sub-network created by those.

$$s^{IA}_{(x,y)} = \sum_{z\in|\Gamma(x)\cap\Gamma(y)|} \frac{e_z}{|\Gamma(z)|} \tag{12}$$

where e_z is the number of links among node z with nodes x and y, and their common neighbors. Since IA considers link between all common neighbors, the algorithm is very time-consuming, thus, Simple Individual Attraction Index (SIA) is also proposed in the same study.

$$s^{SIA}_{(x,y)} = \sum_{z\in|\Gamma(x)\cap\Gamma(y)|} \frac{1}{k_z} x \frac{e+2}{|\Gamma(x)\cap\Gamma(y)|} \tag{13}$$

Here, e is the average number of links among node z with nodes x and y, and their common neighbors.

2.1.13 The Mutual Information Index (MI). : This method examines the link prediction problem using information theory, and measures the likelihood by conditional self-information when their common neighbors are known [99].

$$s^{MI}_{(x,y)} = -I(e_{x,y}|\Gamma(x)\cap\Gamma(y)) \tag{14}$$

where $e_{x,y}$ is the conditional self-information of the existence of a link between node x and y given the set of their common neighbors. The smaller $-I(e_{x,y}|\Gamma(x)\cap\Gamma(y))$ means the higher likelihood to be linked, and it can be derived from the property of the self-information as:

$$I(e_{x,y}|\Gamma(x)\cap\Gamma(y)) = I(e_{x,y}) - I(e_{x,y};\Gamma(x)\cap\Gamma(y)) \tag{15}$$

If all the link between common neighbors are assumed to be independent of each other, then

$$I(e_{x,y};\Gamma(x)\cap\Gamma(y)) = \sum_{z\in|\Gamma(x)\cap\Gamma(y)|} I(e_{x,y};z) \tag{16}$$

Here, $I(e_{x,y})$ is the self-information of that node pair of x and y is connected while $I(e_{x,y};\Gamma(x)\cap\Gamma(y))$ is is the mutual information between the event that node pair conditioned on the common neighbors, and can be calculated as:

$$I(e_{x,y}) = -log2(1 - \prod_{i=1}^{|\Gamma(y)|} \frac{|E| - |\Gamma(x) - i + 1|}{|E| - i + 1}) \tag{17}$$

$$I(e_{x,y};z) = \frac{1}{|\Gamma(z)|(|\Gamma(z)|-1)} \sum_{u,v\in\Gamma(z):u\neq v} I(e_{u,v}) - I(e_{u,v}|z) \tag{18}$$

Therefore, the conditional self-information of nodes x and y being connected is derived as [75]:

$$I(e_{x,y}|z) = log2 \frac{|\{e_{x,y} : x, y \in \Gamma(z), e_{x,y} \in E\}|}{\frac{1}{2}|\Gamma(z)|(|\Gamma(z)|-1)} \tag{19}$$

2.1.14 CAR Based Index (CB). : When a node interacts with another neighbor node, it is called as first-level neighborhood; whereas, the interaction between first-level neighbor node and its neighbor node is called second-level neighborhood for seed node. According to Cannistraci [21], researchers mostly consider first-level neighborhood because second-level neighborhood is very noisy; however, it also carries important information about the topology of the network. Therefore, CB filters these noises and considers to nodes interlinked with neighbors mostly. The similarity metric can be calculated as follows:

$$s^{CB}_{(x,y)} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} 1 + \frac{\Gamma(x)+\Gamma(y)+\Gamma(z)}{2} \tag{20}$$

2.1.15 Functional Similarity Weight (FSW). : This index is first used by Chou et al. in order to understand the similarity of physical or biochemical characteristics proteins [25]. Their motivation is based on the Czekanowski-Dice distance used in [20] to estimate functional similarity of proteins. The score can be defined as:

$$s^{FSW}_{(x,y)} = (\frac{2|\Gamma(x)\cap\Gamma(y)|}{|\Gamma(x)-\Gamma(y)| + 2|\Gamma(x)\cap\Gamma(y)| + \lambda(x,y)})^2 \tag{21}$$

Here, $\lambda(x,y)$ is used to penalize the nodes which have very few common neighbor, and defined as:

$$\lambda(x,y) = max(0, \Gamma_{avg} - (|\Gamma(x)-\Gamma(y)|) + (|\Gamma(x)\cap\Gamma(y)|)) \tag{22}$$

where Γ_{avg} is the average number of neighbours that each nodes has in the network.

2.1.16 Local Neighbors Link Index (LNL). : Motivated by the cohesion between common neighbors and predicted nodes, both attribute and topological features is examined in [113].

$$s^{LNL}_{(x,y)} = \sum_{z\in|\Gamma(x)\cap\Gamma(y)|} w(z) \tag{23}$$

where $w(z)$ is weight of node z, defined by:

$$w(z) = \frac{\sum_{u\in\Gamma(x)\cup x}\delta(z,u) + \sum_{v\in\Gamma(y)\cup y}\delta(z,y)}{|\Gamma(z)|} \tag{24}$$

Here, $\delta(a,b)$ is the boolean variable which represents whether there exist a link between a and b, and is equal to 1 when there is link between a and b, otherwise equals to 0.

2.1.17 Local Affinity Structure Index (LAS). : LAS shows the affinity relationship between a pair of nodes and their common neighbors. The hypothesis used here is that more affinity of two nodes and their common neighbor increases the probability to be linked [98].

$$s^{LAS}_{(x,y)} = \frac{|\Gamma(x)+\Gamma(y)|}{|\Gamma(x)|} + \frac{|\Gamma(x)+\Gamma(y)|}{|\Gamma(y)|} \tag{25}$$

2.2 Global Similarity Based Approaches

Global similarity based approaches, on the contrary of local ones, use the whole topology of the network to rank similarity between node pairs; therefore, they are not limited to measure the similarity between nodes which are locating far away from each other. Although considering the whole topology of network gives more flexibility in link prediction analysis, it also increases the time complexity of the algorithm. Based on the characteristics of the method used, they can be classified as path-based methods, in which ensemble of all paths between node pairs are used, and random walk based methods, in which transition probabilities between two nodes to represent the how far random walker traveled are used.

2.2.1 *Katz Index (KI).*

: The metric defined by Katz in [57] not only considers the path between specific neighbor nodes, rather, sums over the sets of paths and exponentially damped by length to be counted more intensively with shorter paths.

$$s^{KI}_{(x,y)} = \sum_{i=1}^{\infty} \beta^{i}.|A^{\langle i \rangle}_{xy})| \tag{26}$$

Here, β is free parameter ($\beta > 0$) and called the "damping factor". One can realize that KI yields very similar score when β is low enough, because the paths which have higher lengths contribute less, and similarity index is simply determined by the shorter paths [73].

In the case of $\beta < \frac{1}{\lambda^A_1}$, where λ^A_1 is the largest eigenvalue of adjacency matrix, the similarity matrix can be written as follows:

$$S^{KI} = (I - \beta A)^{-1} - I \tag{27}$$

where I is the identity matrix.

2.2.2 *Global Leicht-Holme-Newman Index (GLHN).*

: The idea behind GLHN is very similar to that of KI, since it also gives high similarity to the nodes if number of paths between these corresponding nodes are high [62].

$$S^{GLHN} = \lambda (I - \beta A)^{-1} \tag{28}$$

where β and λ are free parameters, and smaller value of β gives more importance to the shorter paths, and vice versa.

2.2.3 *SimRank (SR).*

This index computes the similarity starting from the hypothesis "two objects are similar if they are related to similar objects.", and is recursively defined [55]. It is equal to 1 when $x = y$, otherwise,

$$s^{SR}_{(x,y)} = \gamma . \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} s^{SR}_{(a,b)}}{|\Gamma(x)|.|\Gamma(y)|} \tag{29}$$

where $\gamma \in [0, 1]$ is called decay factor, and controls how fast the effect of neighbor nodes (a and b) reduces as they move away the original nodes (x,y). SR can be explained in terms of random walk process, that is, $s^{SR}_{(x,y)}$ measures how long the two random walkers are expected to meet on a particular node, starting with the x and y nodes. Its applicability is constrained on large networks due to its computational complexity [67, 108].

2.2.4 *Pseudo-inverse of the Laplacian Matrix (PLM).*

: Using Laplacian matrix ($L = D - A$) rather than Adjacency matrix (A) gives an alternative representation of a graph, where D is the unit diagonal matrix [95]. The Moore-Penrose pseudoinverse of the Laplacian matrix, represented by L^{+} can be used in the calculation of proximity measures [39]. Since PLM is calculated as inner product cosine similarity, it is also called "cosine similarity time" in the literature [108].

$$s^{PLM}_{(x,y)} = \frac{L^{+}_{(x,y)}}{\sqrt{L^{+}_{(x,x)} L^{+}_{(y,y)}}} \tag{30}$$

2.2.5 *Hitting Time (HT) and Average Commute Time (ACT).*

: HT is defined as the average number of steps to be taken by random walker to reach node y, starting from x. Because HT is not a symmetric metric, one may consider to ACT, which is defined as the average number of steps to be taken by random walker starting from x to reach node y, and that from y to reach node x.

$$s^{HT}_{(x,y)} = 1 + \sum_{z \in \Gamma(x)} P_{x,z} s^{HT}_{(z,y)} \tag{31}$$

Here, $P_{i,j} = D^{-1}A$, where A is the adjacency matrix and $D_{i,j} = 0$ and $D_{i,i} = \sum_j A_{i,j}$ [108].

$$s^{ACT}_{(x,y)} = s^{HT}_{(x,y)} + s^{HT}_{(y,x)} \tag{32}$$

For the sake of computational simplicity, ACT can be computed in closed form by using the pseudo-inverse of the Laplacian matrix of the graph as follows [39]:

$$s^{ACT}_{(x,y)} = m(L^{+}_{(x,x)} + L^{+}_{(y,y)} - 2L^{+}_{(x,y)}) \tag{33}$$

One challenge of HT and ACT is that it gives very small proximity measures when terminal node has high stationary probability, π_y regardless of the identity of starting node. This problem can be solved by normalizing the scores as $-s^{HT}_{(x,y)}.\pi_y$ and $-(s^{HT}_{(x,y)}.\pi_y + s^{HT}_{(y,x)}.\pi_x$, respectively [68].

2.2.6 *Rooted PageRank (RPR).*

: PageRank (PR) is the metric used by Google Search to determine the relative importance of the webpages by treating links as a vote. The recursively defined PR on $G(V, E)$ can be obtained for single node as follows:

$$s^{PR}_{(x)} = \frac{1 - \beta}{|V|} + \beta \sum_{z \in \Gamma^{-1}(x)} \frac{s^{PR}_{(z)}}{|\Gamma(x)|} \tag{34}$$

where, β is the damping factor. Personalized PR (PRP) can be obtained by inner product of the two PR values of the nodes as:

$$s^{PRP}_{(x,y)} = s^{PR}_{(x)}.s^{PR}_{(y)} \tag{35}$$

RPR, on the other hand, defines that the rank of a node is proportional to the likelihood that it can be reached through random walk [19, 108].

$$s^{RPR}_{(x,y)} = (1 - \lambda)(1 - \lambda P_{x,y})^{-1} \tag{36}$$

Here, $P_{i,j} = D^{-1}A$, where A is the adjacency matrix and $D_{i,j} = 0$ and $D_{i,i} = \sum_j A_{i,j}$. It should be noted that, one can calculate PR by averaging the columns of RPR [93].

2.2.7 Escape Probability (EP). : The metric, which can be derived from RPR, measures the likelihood that the random walk, starting from node x, visits node y before coming back to the node x again [104]. If we define $Q(x, y)$ to be equal to $(1-\lambda D^{-1}A)^{-1} = s^{RPR}_{(x,y)}/(1-\lambda)$, the equation of EP can be written as follows [93]:

$$s^{EP}_{(x,y)} = \frac{Q(x,y)}{Q(x,x).Q(y,y) - Q(x,y).Q(y,x)} \tag{37}$$

2.2.8 Random Walk (RW). : Random walk, introduced by mathematician Karl Pearson, is nothing more than the following Markov chain: First, starting node is selected and moved to the randomly selected neighbor. Then new starting node is considered as previous terminal node and walk is repeated [85]. The probability vector of reaching a node starting from node x is defined as follows:

$$\vec{p_x}(t) = M^T \vec{p_x}(t-1) \tag{38}$$

where M is the matrix called as "transition probability matrix", and equals to $A_{i,j}/\sum_k A_{i,k}$, where A is the adjacency matrix [72].

Suppose that α denotes the probability that node x randomly moves toward any neighbor node, so, $1-\alpha$ represents the probability that random walker turns back to the node x. The closed form of the solution at steady state is as follows:

$$\vec{p_x} = (1-\alpha)(I - \alpha M^T)^{-1}\vec{s_x} \tag{39}$$

Here, $\vec{s_x}$ represents seed vector in which elements of $\vec{s_x}^x$ are 1, the others are equal to 0. Similarity metric between nodes x and y, $s^{RW}_{(x,y)}$, can be defined as:

$$s^{RW}_{(x,y)} = \vec{p_x}^y \tag{40}$$

2.2.9 Random Walk with Restart (RWR). : Since RW does not yield symmetric matrix, the metric of RWR, very similar to RPR, looks for the probability that random walker starting from node x visits node y and come back to initial state node x at steady state.

$$s^{RW}_{(x,y)} = \vec{p_x}^y + \vec{p_y}^x \tag{41}$$

2.2.10 Maximal Entropy Random Walk (MERW). : The basic MERW algorithm, based on maximum uncertainty principle, were used due to necessity of defining uniform path distribution in Monte Carlo simulations even in 1980s [52], however, its application on stochastic models are very recent [34]. The main purpose here is to maximize the entropy of the random walk which can be defined as follows:

$$\lim_{l\to\infty} \frac{-\sum_{A^l_{xy}\in A^l} p(A^l_{xy}) \ln p(A^l_{xy})}{l} \tag{42}$$

Here, $p(A^l_{xy})$ is the multiplication of iterative transition matrices $(M_{xz}.M_{zt}...M_{ty})$, and those can be calculated as follows:

$$M_{ij} = \frac{A_{ij}}{\lambda}\frac{\psi_j}{\psi_i} \tag{43}$$

where A is the adjacency matrix, and ψ is the normalized eigenvector with normalization constant λ [65, 75].

2.2.11 The Blondel Index (BI). : The index is proposed by Blondel et al. to measure the similarity of "automatic extraction of synonyms in a monolingual dictionary" [16]. Although BI used for understanding the similarity between two different graphs, Martinez et al. shows its iteratively computed from can also bi used to understand the similarity of two nodes in a single graph [75].

$$S(t) = \frac{AS(t-1)A^T + A^T S(t-1)A}{||AS(t-1)A^T + A^T S(t-1)A||} \tag{44}$$

where A is adjacency matrix and $||M||$ is the Frobenius matrix form, which can be calculated as follows:

$$||M_{mxn}|| = \sqrt{\sum_{i=1}^{m}\sum_{i=1}^{m} M^2_{i,j}} \tag{45}$$

The similarity metric is obtained when S(t) is converged $s^{BI}_{(x,y)} = S_{x,y}(t=c)$, where $t = c$ denotes the steady state level.

2.3 Quasi-Local Similarity Based Approaches

Trade-off between the efficiency of using the information of whole network topological structure in global approaches and the less time complexity of the algorithms in local approaches emerged a balanced method which is called quasi-local similarity based approaches. These approaches are also limited in the calculation of similarities between arbitrary node pairs; however, they give an opportunity to compute the similarity between a node and its neighbors of neighbors. Although some of them still consider to the whole topology of network, their time complexity is still below than that of global approaches.

2.3.1 The Local Path Index (LPI). : The index, very similar to well known approaches of KI and CN, considers local path with a wider perspective by using the information of not only the nearest neighbor but also the next 2 and 3 nearest neighbor [71, 122].

$$S^{LP} = A^2 + \alpha A^3 \tag{46}$$

where A is adjacency matrix and α is free parameter to adjust the relative importance of of the neighbors within length 2 distances and length 3 distances. The metric can be also extended for higher orders as:

$$S^{LP^{(n)}} = \sum_{i=2}^{n} \alpha^{i-2} A^i \tag{47}$$

Since the high complexity in higher order LP, only neighbors within length 3 distances are preferred more. One can easily realize that the similarity matrix degenerates to CN when $n = 2$, and may give very similar result with KI at low α values without necessitating the inverse transform . Similarity between two nodes can be found as $s^{LP}_{(x,y)} = S^{LP}_{x,y}$.

2.3.2 Local (LRW) and Superposed Random Walks (SRW). : Although algorithms based on random walks perform well, sparsity and the amount of data are still challenging problem for these algorithms; therefore, Liu and Lü proposed LRW which does not concentrates the stationary , instead, number of iterations is fixed [71].

$$s_{(x,y)}^{LRW}(t) = \frac{|\Gamma(x)|}{2|E|}\vec{p_y^x}(t) + \frac{|\Gamma(y)|}{2|E|}\vec{p_x^y}(t) \tag{48}$$

Since superposing all the random walkers starting from same points may help to prevent sensitive dependence of LRW to the nodes further away, SRW is proposed as:

$$s_{(x,y)}^{SRW}(t) = \sum_{l=1}^{t} s_{(x,y)}^{LRW}(l) \tag{49}$$

where t denotes the time steps.

2.3.3 *Third-Order Resource Allocation Based on Common Neighbor Interactions (RACN).*

: Motivated by RA index, Zhang et el. proposed RACN in which resource of nodes are allocated to the neighbors [75, 120].

$$s_{(x,y)}^{RACN} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma(z)|} + \sum_{e_{i,j}\in E,\, |\Gamma(j)|<|\Gamma(i)|} (\frac{1}{|\Gamma(i)|} - \frac{1}{|\Gamma(j)|}) \tag{50}$$

where $i \in \Gamma(x)$ and $j \in \Gamma(j)$. The superiority of the RACN to the original RA has been shown in different datasets [121].

2.3.4 *FriendLink Index (FL).*

: The similarity of two nodes is determined according to their normalized counts of paths between them with varying length l.

$$s_{(x,y)}^{FL} = \sum_{i=1}^{l} \frac{1}{i-1} \frac{|A_{x,y}^{i}|}{\prod_{j=2}^{i}(|V|-j)} \tag{51}$$

where $|V|$ is the number of vertices in graph. The metric is favorable due to its high performance and speed [84].

2.3.5 *PropFlow Predictor Index (PFP).*

: PFP is a metric which is inspired by Rooted PageRank, and simply equals to the probability that the success of random walk started as node x and terminates at node y not more than l steps [69]. This restricted random walk selects links based on weights, denoted as ω [108].

$$s_{(x,y)}^{PFP} = s_{(a,x)}^{PFP} \frac{\omega_{xy}}{\sum_{k\in\Gamma(x)}\omega_{xy}} \tag{52}$$

The most important superiority of PFP is its widespread use in directed, undirected, weighted, unweighted, sparse or dense networks.

3 MAXIMUM LIKELIHOOD METHODS

Maximum likelihood approaches assume networks to have a known structure. Based on the structure of the network, they fit a statistical model and then compute the probability of non-observed links to exist. It should be noted that maximum likelihood methods are very time-consuming and generally are not very accurate.

3.1 Hierarchical Structure Model

[26] proposed a method in which they consider networks having a hierarchical structure. In fact, many networks have a hierarchical structure, e.g. protein interaction networks, metabolic networks, etc. [87]. The method represents the network by a dendrogram with |N| leaves and |N-1| internal nodes. Each leaf is a node from the original network and each internal node represents the relationship of the descendent nodes in dendrogram. In a representative dendrogram, the likelihood of dendrogram with the set of internal node probabilities is calculated as follows:

$$L(D,\{p_r\}) = \prod_{r\in D} p_r^{E_r}(1-p_r)^{L_rR_r-E_r} \tag{53}$$

In the equation above, D is the dendrogram, each internal node r is associated with a probability p_r and the connecting probability of a pair of nodes (leaves) is equal to $p_{r'}$ while r' is the lowest common ancestor of these two nodes. L_r and R_r are number of leaves in the left and right subtrees rooted at r, respectively. E_r is the number of links in the network connecting nodes that have internal node n as their lowest common ancestor in D.

If dendrogram D is fixed, it is pr that maximizes the likelihood function for each r is calculated by:

$$\bar{p_r} = \frac{E_r}{L_rR_r} \tag{54}$$

So, the likelihood of the dendrogram is:

$$L(D) = \prod_{r\in D}\left[(1-\bar{p_r})^{1-\bar{p_r}}\bar{p_r}^{\bar{p_r}}\right]^{L_rR_r} \tag{55}$$

The algorithm to find the missing links then is as follows [73]: 1) finding a set of dendrograms representing the network. 2) for each pair of nodes i,j which are not connected in the network, calculate the mean connecting probability by averaging the corresponding probability $< p_{ij} >$ over all sampled dendrograms. 3) sorting the nodes based on $< p_{ij} >$. The higher the value, the more likely the link exists.

3.2 Stochastic Block Model

When the network can not be represented as a hierarchical structure, another approach will be taken. This new approach assumes that nodes are in communities or blocks. Then, the probability of the existence of an edge between two nodes is related to the block they belong to. We first determine a partition M in which all nodes are assigned to one group. The connecting probability for two nodes in groups α and β is shown by $Q_{\alpha\beta}$. $l_{\alpha\beta}$ shows the number of edges between nodes in groups α and β, and $r_{\alpha\beta}$ is number of pairs of node that one node is in α and the other one is in β. The likelihood of the network structure is calculated by [46]:

$$L(A|M) = \prod_{\alpha\leq\beta} Q_{\alpha\beta}^{l_{\alpha\beta}}(1-Q_{\alpha\beta})^{r_{\alpha\beta}-l_{\alpha\beta}} \tag{56}$$

The optimal $Q_{\alpha\beta}$ is:

$$Q_{\alpha\beta}^{*} = \frac{l_{\alpha\beta}}{r_{\alpha\beta}} \tag{57}$$

And finally the reliability of a link using Bayes theorem [12] is [46]:

$$R_{xy} = L(A_{xy}=1|A) = \frac{\int_{\Omega} L(A_{xy}=1|M)L(A|M)p(M)dM}{\int_{\Omega} L(A|M')p(M')dM'} \tag{58}$$

here, ω is the set of possible partitions. Due to the fast growth of ω when number of nodes increase, this stochastic block model is not appropriate for large netwroks.

4 PROBABILISTIC METHODS

Probabilistic models aim at observing the relational structure of a network and predicting the weight or value of links in the network. Three main probabilistic models for link prediction are: probabilistic relational models (PRM), stochastic relational models (SRM) and probabilistic entity relationship models [73]. Whether the network is directed or undirected, different models are used.

To understand the notations and underlying concepts of probabilistic models of link prediction, one should be familiar with relational algebra notations, relational bayesian networks (RBN) [54] and relational markov networks (RMN)[102].

4.1 Probabilistic Relational Models

PRM [40] defines joint probability distribution over the attributes of a relational dataset [73]. A schema includes a set of classes and a set of relations. Each entity in a schema, contains some attributes, and the value of each attribute is limited to a predefined domain. This is defined as a Schema. A skeleton structure σ of a relational schema is a partial specification of an instance of the schema [40]. [73] gives student-course selection system as an example of skeleton graph. Students and courses are the two types of nodes. Each student can have four attribute: grade, age, sex, and department (the same applies for course). The relation of this skeleton is selection (i.e. students select a course).

The conditional probability distribution (CPD) of a variable x given its parents is as follows [73]:

$$\prod_{t\in T}\prod_{X_i^t\in X^t}\prod_{v:T(v)=t} p(x_{v_i}^t|pa_{x_{v_i}^t})\prod_{e:T(e)=t} p(x_{e_i}^t|pa_{x_{e_i}^t}) \tag{59}$$

Here $T(v)$ is type of node, T is the types set, $x_{v_i}^t$ is attribute value of node v_i with type t, and pa_x denotes the parents of node x [73].

The disadvantage of this model is that it is computationally difficult and naively avoids cycles. On the other hand, Relational Markov Networks use undirected graph. They are easy to learn discriminatively and they can include cycles.

4.2 Probabilistic Entity-Relationship Models

A probabilistic entity-relationship model is based on entity-relationship (ER) model [51]. entity-relationship model is an abstraction of database structure. The most important and widely used entity-relationshil model is directed acyclic probabilistic entity-relationship model (DAPER). six classes make a DAPER: Entity classes, Relationship classes, Attribute classes, Arc classes, Local distribution classes, and Constraint classes. These models are capable of performing better than the other models when the relational structure is uncertain, and are more expensive than PRMs. .

4.3 Stochastic Relational Models

The key concept of stochastic relational models, is 'a stochastic entity-wise process which is caused by interplay of multiple entity-wise Gaussian processes (GP)' [117]. Stochastic Relational Models are discriminative [117].

It is assumed that links r are by a latent relational function $t : U \times V \longrightarrow \mathbb{R}$ and $p(r_{i,n}|t_{i,n})$ where $r_{i,n}$ is each link and $t_{i,n}$ is its latent value. Also, θ_Σ and θ_ω are the hyperparameters for GP kernel function on U and GP kernel function on V, respectively. If $\mathbb{I}$ be index set of entity pairs, the marginal likelihood would be:

$$p(R_{\mathbb{I}}|\theta) = \int \prod_{(i,n))} p(r_{i,n}|t_{i,n})p(t|\theta)dt \tag{60}$$

first the θ values are estimated by maximizing the evidence. Then the link for a new pair of entities can be predicted by marginalization over a posteriori $p(t|R_{\mathbb{I}}, \theta)$ [117].

5 FEATURE LEARNING METHODS

Automatic graph feature learning is feasible using graph embedding and representation learning models. These learning methods can be viewed as graph dimensionality reduction techniques which map the graph structure based on features best describing the relations between the nodes and embed the nodes to a low dimensional feature space [49]. Embedding algorithms try to preserve the structure of the embedded graph in the vector space by keeping the neighboring nodes closer to each other [43]. In contrast with classical methods which study node neighborhoods through calculating similarity based metrics, feature representation learning algorithms can learn features automatically preventing hand-engineered features [45].

Mapping the graph to a vector space is also known as encoding and reconstruction of the node neighborhood from the embedded graph is referred as decoding. Graph representation can be learned through supervised or unsupervised manners that both learn to optimize the graph embeddings [49]. This mapping can be defined for graph G = < V, E > as $f : v_i \longrightarrow y_i \in \mathbb{R}^d, \forall i \in [n]$ such that $d \ll | V |$ [43]. V refers to the set of vertices of graph G, E is defined as the set of edges in the graph, n denotes the total number of vertices, v_i is a sample node which has been embedded to d-dimensional vector space and the embedded node is represented by y_i.

Optimizing the graph mappings consists of joint optimization of the encoder and the decoder resulting in learning the graph transformation to low dimensional feature space [49]. Mostly, the Stochastic Gradient Descent algorithm is used for this optimization problem. The decoding function receives a set of node embeddings as input to decode the graph statistics or class information by reconstructing the node neighborhood. A pairwise decoder which map a pair of embedded nodes to a real value measurement of the proximities based on the original graph neighborhoods is defined as follows [49]:

$$DEC(DEC(v_i), DEC(v_j)) = DEC(y_i, y_j) \approx s_G(v_i, v_j) \tag{61}$$

where s_G refers to the class information or graph statistics for the two nodes v_i and v_j in the original graph G, y_i and y_j are the embeddings of these nodes respectively. The mapping by the decoder can be denoted as $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ [49].

Graph representation learning models and embedding algorithms are used in literature closely and interchangeably. However, embedding algorithms to a low dimensional vector space might refer to node embedding or embedding the whole structure of the graph which have diverse applications. In this work we refer to graph representation and embedding algorithms only as methods to embed each local topology of nodes to vector space that can reconstruct the neighborhood for the embedded node.

Generally, graph classifiers learn some parameters over the input embeddings which work as input features to the classifiers [43]. Studies on graph representation learning have mostly considered learning the structure of small graphs or subgraphs. The difficulty of learning structure of huge graphs like complex social networks arises from the large number of vertices and edges besides their evolving structure over time [7, 8, 64].

Diverse graph embedding techniques have become available through recent studies. These algorithms can be categorized into 1) Matrix Factorization Based Models, 2) Random Walk Based Models, and 3) Deep Neural Network Models. Link prediction studies based on these representation learning methods can be viewed as either generative models or discriminative models or combination of the two. Generative algorithms study the problem of link prediction through an optimization problem maximizing future edge likelihoods [107]. A few generative models consider both node and edge formations by directly learning over the adjacency matrix of the graph. An example is GraphRNN which studies adjacency matrices by applying a mapping function to the adjacency matrix to gain sequences of nodes and feed the sequences to a multilayer perceptron [115]. This generative model learns the distribution of the structure of the flattened graphs and employs BFS to limit the number of edge predictions[115].

5.1 Matrix Factorization Based Methods

The links between nodes can also be represented by using adjacency matrix, in which each row and column represent different nodes and the boolean variable denotes whether link exist between node pairs. In matrix factorization based methods, vector representation of the topology-related features form N-dimensional space, where N is the number of nodes in the network. The main purpose, here, is to reduce dimensionality of this space by preserving nonlinearity and locality. Thus, global structure of topology may be generally lost[100]. Singular value decomposition(SVD) was one of the commonly used method due to its feasibility in low-rank approximations [27, 83]. Here, link function, $L(.)$ is defined as $G \approx L(U \cap U^T)$, where $U \in \mathbb{R}^{nxk} \cap \in \mathbb{R}^{kxk}$, where n denotes number of nodes and k represents the number of latent variables in SVD. The similarity between node pairs $s(x, y)$ is defined by $L(u_x^T \cap u_y^T)$. Since, the methods based on graph embedding techniques using inner product decoders aimed to follow and improve one of the earliest dimensionality reduction technique, Laplacian Eigenmaps, we will look into details these methods.

5.1.1 Laplacian Eigenmaps. The old algorithm was proposed by Belkin and Niyoki [14], first constructs graph using ϵ or K nearest neighbor [27], then loss function is minimized using the weight of node pairs. The decoder in encoder-decoder framework of Laplacian Eigenmaps can be identified as:

$$DEC(z_i, z_j) = |z_i - z_j|_2^2 \tag{62}$$

where the weights of the loss function is defined pairs computed by the similarity between node pairs in the graph:

$$\mathcal{L} = \sum_{(v_i, v_j) \in D} DEC(z_i, z_j).W(v_i, v_j) \tag{63}$$

5.1.2 Graph Factorization. Graph factorization uses the same loss function given in Laplacian Eigenmaps, and optimized it using stochastic gradient descent. Therefore, it is also very efficient in the existence of large networks [100]. Its aim is to distribute the framework to partition the vector space and minimize the number of neighboring of nodes rather than edges[5].

5.1.3 GraRep. GraRep also uses the same loss function given in Laplacian Eigenmaps, and optimized it using stochastic gradient descent. Therefore, it is also very efficient in the existence of large networks [100]. While model reducing the dimension of the vector space, it also integrates global topological structure information into learning [22].

5.1.4 HOPE. An inner product based method which preserves the asymmetric transitivity for directed graph embeddings. This property seems crucial to capture the graph structure while factorizing the graph vertices to the vector space. This property also comes in handy for decoding the embedded graph features [83]. To approximate the high order proximities in this model, a Singular Value Decomposition is applied on the proximity matrix and the optimized vector representations are constructed using the singular values [83].

5.2 Random Walk Based Methods

Graphs exploration and sampling with random walks or search algorithms like Bridth First Search (BFS) and Depth First Search (DFS) have been used to investigate node features including node centrality and node similarity [43].The importance of exploring graphs with search algorithms is more obvious for huge graphs including graphs of social networks to decrease the complexity by limiting node and edge options. Representations with BFS provide information about similarity of nodes in case of their roles in the network, for instance being a hub [45]. In contrary, representations with DFS can provide information about the communities that nodes belong to. These algorithms have been recently applied along with generative models to introduce edges and nodes directly to the network [107]. Generative algorithms study the problem of link prediction through an optimization problem maximizing future edge likelihoods.

5.2.1 DeepWalk. This method approaches the graph representation learning as a natural language problem [86]. By applying random walks to a set of random vertices through a stream of short walks with a specific length, the nodes' neighborhood and community information would be available. The SkipGram optimization model which is mostly designed for language processing is employed for the objective function to train and learn the graph representations.

5.2.2 Node2vec. This method extends DeepWalk to approach graph embedding by a combination of BFS and DFS search algorithm to not only learn node features, but also edge embeddings [45]. Node2vec learns the representation of the edges through the embedding of pairs of nodes by 2nd order random walk and applies bagging over the embedded features of the individual nodes.

5.2.3 GraphGAN. In [107] an online edge generator adds new edges to the structure of the graph based on random walk starting from a single node, while a discriminator is dealing with the problem of link prediction learning over the features of the authentic edges. GraphGAN bridges the disciminative and generative models for network evolution. However, this model does not consider introducing new nodes to the structure of the network and only studies edges.

5.2.4 GraphSAGE. Supervised learning representation of evolving graphs based on the aggregated local feature information from the neighboring nodes is addressed in [48]. The evolution of the network is considered for both links and nodes in the graph. Starting from a node, GraphSAGE samples a uniform number of immediate neighboring nodes to collect their local features and map them to feature vectors. It concatenates the node's current representation at depth equal to 1 to the same feature vector and the process continues until a defined depth K is met. Then the results are fed into a fully connected neural network to learn the aggregator's weights. The aggregator architectures is either Mean aggregator, LSTM, or Pooling[48].

5.3 Neural Network Based Methods

5.3.1 Graph Neural Networks. The introduction of neural networks, specially convolutional neural networks to graph structures have led to extract features from complex graphs flexibly. The features for these models include the information from the topology of the network aggregated by the node attributes available from the data domain [49]. The idea behind these models is that the structure of the local neighborhood can be learned through the aggregated feature information instead of exploring the whole graph.

In [50] the problem of link prediction is studied using a combination of two convolutional neural networks for the graph network of molecules. The molecules are represented having a hierarchical structure for their internal and external interactions. The graph structure transformation to a low dimensional vector space is obtained from an internal convolutional layer which is randomly initialized for each node representation and trained by backpropagation. The external conovlutional layer receives the embedded nodes as input to learn over the external graph representations. Finally, the link prediction algorithm consists of a multilayer neural network which was accepting the final representations to predict the molecule-molecule interactions by a softmax function.

5.3.2 Graph Autoencoders. Autoencoders consist of an encoder-decoder structure in which the encoder learns to embed the graph into a low dimensional vector space by preserving the structural information, the decoder learns to decode the embedded information of the graph and output the studied labels. This output might contain community belonging label or positive/negative link prediction class labels [49]. The neural network based architecture of autoencoders results in extracting the complex features of the graphs. In contrast with the factorization based models which encode each node directly to a single representation in the vector space, autoencoders learn the graph structures using neural network architectures and reduce the graph dimensionality in accordance with the number of channels of the autoencoder hidden layers. These models also outperform the factorization based node mappings as a result of being able to embed the nodes into sequences with diverse length [23]. This benefits the autoencoders to not only achieve high performances for testing over the unseen node embeddings, but also aggregate the node attributes to improve their prediction accuracy more [49].

LINE. This model is a combination of two encoder-decoder structures to study and optimize first and second node proximities in the vector space [49].

DNGR. This method embeds the node local neighborhood information using a random surfing method and studies single embeddings through autoencoders than pairwise transformations. The neural network based architecture of autoencoders results in extracting the complex features of the graphs and include nonlinearity [23].

SDNE. Is a representation learning model which is very similar to DNGR by a few differences in accordance with the similarity based metric to study the graph, objective function optimization, and the encoder-decoder implementation details.

5.3.3 Graph Differentiable Pooling. Employing neural networks to learn the structure of complex graphs is getting more popular. However, the hierarchical structure of the graph cannot be learned through these models. This problem is well addressed by DIFFPOOL [114] which is a pooling method to learn complex graph representations. Applying this pooling process in conjunction with neural networks works as a mapping approach for nodes to soft clusters which will be fed into a convolutional layer as the inputs [114]. This model facilitates the learning graph representations of complex structures and their deep features which are beneficial for the problems of graph classification and link prediction.

6 SUPERVISED LINK PREDICTION

Introduction of supervised learning algorithms to the problem of link prediction led to the state-of-the-art models achieving high prediction performances [33]. These models view the problem of link prediction as a classification task. To approach the link prediction problem, supervised models are supposed to tackle a few challenges. These challenges include the unbalanced data classes resulting from the sparsity property of real networks and calculation of the neighborhood similarity metrics to use as informative independent features [49]. Learning over the similarity measurements as features which represent the structure of the network along with the node attributes according to the data domain result in supervised learning approaches to be powerful classifiers. Another option to approach link prediction problem with supervised algorithms is through learning the representation of the features through an objective function which maximizes the prediction accuracy [45].

Numerous literature have approached link prediction problem through classification models. Support Vector Machines, K-nearest

Table 1: Network dataset collections

Collection	Description
SNAP[63]	A collection of more than 90 network datasets by Stanford Network Analysis Platform. With biggest dataset consisting of 96 million nodes.
BioSNAP[74]	More than 30 Bio networks datasets by Stanford Network Analysis Platform
KONECT[59]	This collection contains more than 250 network datasets of various types, including social networks, authorship networks, interaction networks, etc.
PAJEK[11]	This collection contains more than 40 datasets of various types.
Network Repository[90]	A huge collection of more than 5000 network datasets of various types, including social networks,
Uri ALON[60]	A collection of complex networks datasets by Uri Alon Lab.
NetWiki[79]	More than 30 network datasets collection of various types.
WOSN 2009 Data Sets[105]	A collection of facebook data provided by social computing group
Citation Network Dataset[101]	A collection of citation network dataset extracted from DBLP, ACM, and other sources.
Grouplens Research[44]	A movie rating network dataset.
ASU social computing data repository[119]	A collection of 19 network datasets of various types: cheminformatics, economic networks, etc.
CNetS[3]	A collection of 11 datasets.
Nexus network repository[2]	A reository collection of network datasets by igraph.
SocioPatterns[1]	A collection of 10 network datasets collected by SocioPatterns interdisciplinary research collaboration.
Mark Newman[80]	A collection of Network datasets by Mark Newman.

Neighbors, Logistic Regression, Ensemble Learning and Random Forrest, Multilayer Perceptron, Radial Basis Function network, and Naive Bayes are just a few supervised learning methods extensively used for link prediction. In [7] a comparison between a few of these supervised models is reported and surprisingly, SVM with RBF kernel was very successful in case of high accuracy and low squared error.

The problem of link prediction for complex networks might be investigated as in the field of computer vision in which the data inputs are images. In [109] The adjacency matrix of the network is represented as a binary image and for the training set and constructing the set of true positives, random perturbations to the image of the adjacency matrix is applied. A generative adversarial network is trained over the perturbed images to generate fake images of the adjacency matrices as inputs to the discriminator network, while the discriminator network is trained to predict the missing links through distinguishing between the real and fake images of the adjacency matrices and minimize the link prediction error.

7 MULTI-SOURCE LINK PREDICTION

Link prediction problem for graphs of networks have been recently studied through frameworks which concurrently extract the features through multiple processes or from multiple resources. Multi-source link prediction techniques seek to combine the advantages each of the separate models can provide. These algorithms might include weights to be assigned according to the contribution of different link prediction techniques. Studying link prediction for multi-layer and multiplex (multi-relational) networks is a basic example of multi-source models. Multiplex networks have multi-layer network structures in which different layers share the same set of nodes [70].

Another example for multi-source link prediction models is CMA-ES [15] which is designed by a linear combination of 16 node similarity based metrics from the local and quasi-local feature extraction categories. These metrics contribute in the problem of link prediction with different weights. The weights for each of the similarity based metrics are found by an evolutionary strategy which optimizes the influence of each of the techniques through minimizing the overall link prediction error.

8 DATASETS

A challenging part of most link prediction studies is implementation and validation of the proposed methods and models. Dataset collection is a time-consuming and labor-intensive work. While some studies build their own dataset, majority of researchers prefer to use an existing dataset. Some popular collections of network datasets which might be used in link prediction studies are introduced in table 1.

9 DISCUSSION AND FUTURE WORK

Studying complex networks to predict the emerging links or missed associations is feasible through a variety of approaches discussed above. Feature extraction techniques, which provide information about node neighborhoods, contribute to the task of link prediction differently from the models extracting global features. Node similarities result in higher chance of connection and community belongings, and increase the probability of link emergence as well. On the one hand, a few models have been proposed to aggregate the benefits from a combination of classical methods. On the other hand, the plentiful unsupervised methods available to extract features make it laborious to pick the appropriate technique for a specific domain.

Machine learning approaches can present solutions to the aforementioned problems by contributing to the task of link prediction through multiple ways. Getting advantage from the feature extraction techniques, supervised learning models can approach the task of link prediction as a classification method and combine node attributes to the similarity based metrics as the model inputs. Additionally, machine learning models can come in handy to pick the right combination of features by optimizing an objective function. Graph embedding and representation learning algorithms can provide a combined solution to the mentioned problems by preparing a low dimensional space which preserves the structural features besides the global similarities. Moreover, learning the graph representations lead to automatic selection of features which maximize the prediction accuracy and prevent hand-engineered features.

Although the discussed models provide solutions to the task of link prediction, approaching the huge graphs of complex networks by the available unsupervised models or the machine learning algorithms are not time efficient. We believe that exploiting the advantages of multiple link prediction methods and approaching this problem through concurrent algorithms is the solution. Concurrent learning of network features by multiple processes or from different sources might incorporate the benefits of all the reviewed models into a single framework.

REFERENCES

[1] [n. d.]. SocioPAttern Research Collaboration. http://www.sociopatterns.org/datasets/.

[2] 2015. Nexus network repository. https://igraph.org/r/doc/nexus.html.

[3] 2018. Center for Complex Networks and Systems Research. http://cnets.indiana.edu/resources/data-repository/.

[4] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.

[5] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. 2013. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 37–48.

[6] William Aiello, Fan Chung, and Linyuan Lu. 2000. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. Acm, 171–180.

[7] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*.

[8] Ana Paula Appel, Renato LF Cunha, Charu C Aggarwal, and Marcela Megumi Terakado. 2018. Temporally Evolving Community Detection and Prediction in Content-Centric Networks. *arXiv preprint arXiv:1807.06560* (2018).

[9] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 519–528.

[10] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[11] Vladimir Batagelj and Andrej Mrvar. 2006. Pajek datasets. http://http://vlado.fmf.uni-lj.si/pub/networks/data/.

[12] Thomas Bayes, Richard Price, and John Canton. 1763. An essay towards solving a problem in the doctrine of chances. (1763).

[13] Punam Bedi and Chhavi Sharma. 2016. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 6, 3 (2016), 115–135.

[14] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*. 585–591.

[15] Catherine A Bliss, Morgan R Frank, Christopher M Danforth, and Peter Sheridan Dodds. 2014. An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science* 5, 5 (2014), 750–764.

[16] Vincent D Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review* 46, 4 (2004), 647–666.

[17] Béla Bollobás. 1998. Random graphs. In *Modern graph theory*. Springer, 215–252.

[18] Bobby-Joe Breitkreutz, Chris Stark, and Mike Tyers. 2003. Osprey: a network visualization system. *Genome biology* 4, 3 (2003), R22.

[19] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30, 1-7 (1998), 107–117.

[20] Christine Brun, François Chevenet, David Martin, Jérôme Wojcik, Alain Guénoche, and Bernard Jacq. 2003. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome biology* 5, 1 (2003), R6.

[21] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. 2013. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific reports* 3 (2013), 1613.

[22] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 891–900.

[23] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep Neural Networks for Learning Graph Representations.. In *AAAI*. 1145–1152.

[24] Nicholas A Christakis and James H Fowler. 2009. Social network visualization in epidemiology. *Norsk epidemiologi Norwegian journal of epidemiology* 19, 1 (2009), 5.

[25] Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong. 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics* 22, 13 (2006), 1623–1630.

[26] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* 453, 7191 (2008), 98.

[27] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2017. A survey on network embedding. *arXiv preprint arXiv:1711.08752* (2017).

[28] Agostino Di Ciaccio and Giovanni Maria Giorgi. 2013. Statistical analysis of social networks. *Rivista Italiana di Economia Demografia e Statistica* 67, 3/4 (2013), 103–110.

[29] Pedro Domingos. 2005. Mining social networks for viral marketing. *IEEE Intelligent Systems* 20, 1 (2005), 80–82.

[30] Yuxiao Dong, Qing Ke, Bai Wang, and Bin Wu. 2011. Link prediction based on local information. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, 382–386.

[31] Sergey N Dorogovtsev and Jose FF Mendes. 2002. Evolution of networks. *Advances in physics* 51, 4 (2002), 1079–1187.

[32] Sergei N Dorogovtsev and José FF Mendes. 2013. *Evolution of networks: From biological nets to the Internet and WWW*. OUP Oxford.

[33] Liang Duan, Shuai Ma, Charu Aggarwal, Tiejun Ma, and Jinpeng Huai. 2017. An ensemble approach to link prediction. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2402–2416.

[34] J Duda. 2012. *Extended maximal entropy random walk*. Ph.D. Dissertation. Ph. D. dissertation, Jagiellonian University, 2012.[Online]. Available: http://www.fais. uj. edu. pl/documents/41628/d63bc0b7-cb71-4eba-8a5a-d974256fd065.

[35] Leo Egghe and Ronald Rousseau. 1990. *Introduction to informetrics: Quantitative methods in library, documentation and information science*. Elsevier Science Publishers.

[36] Paul Erdos and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[37] P. ErdÃűs and A. Rényi. 1959. On random graphs I. *Publ. Math. Debrecen* 6 (1959), 290–297.

[38] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[39] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering* 19, 3 (2007), 355–369.

[40] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. 1999. Learning probabilistic relational models. In *IJCAI*, Vol. 99. 1300–1309.

[41] Edgar N Gilbert. 1959. Random graphs. *The Annals of Mathematical Statistics* 30, 4 (1959), 1141–1144.

[42] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1019–1028.

[43] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151 (2018), 78–94.

[44] grouplens. [n. d.]. Movielens rating dataset. https://grouplens.org/datasets/movielens/.

[45] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[46] Roger Guimerà and Marta Sales-Pardo. 2009. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National*

Academy of Sciences 106, 52 (2009), 22073–22078.

[47] Alireza Hajibagheri, Gita Sukthankar, and Kiran Lakkaraju. 2016. A holistic approach for link prediction in multiplex networks. In *International Conference on Social Informatics*. Springer, 55–70.

[48] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[49] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[50] Shonosuke Harada, Hirotaka Akita, Masashi Tsubaki, Yukino Baba, Ichigaku Takigawa, Yoshihiro Yamanishi, and Hisashi Kashima. 2018. Dual Convolutional Neural Network for Graph of Graphs Link Prediction. *arXiv preprint arXiv:1810.02080* (2018).

[51] David Heckerman, Chris Meek, and Daphne Koller. 2007. Probabilistic entity-relationship models, PRMs, and plate models. *Introduction to statistical relational learning* (2007), 201–238.

[52] JH Hetherington. 1984. Observations on the statistical iteration of matrices. *Physical Review A* 30, 5 (1984), 2713.

[53] Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc Vaudoise Sci Nat* 37 (1901), 547–579.

[54] Manfred Jaeger. 1997. Relational bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 266–273.

[55] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 538–543.

[56] Michał Karoński. 1982. A review of random graphs. *Journal of Graph Theory* 6, 4 (1982), 349–389.

[57] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[58] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2010. Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*. Springer, 337–357.

[59] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13 Companion)*. ACM, New York, NY, USA, 1343–1350. https://doi.org/10.1145/2487788.2488173

[60] Usi Alon lab. 2006. COLLECTION OF COMPLEX NETWORKS. http://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks.

[61] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Physical review E* 78, 4 (2008), 046110.

[62] Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. 2006. Vertex similarity in networks. *Physical Review E* 73, 2 (2006), 026120.

[63] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[64] Jundong Li, Kewei Cheng, Liang Wu, and Huan Liu. 2018. Streaming link prediction on dynamic attributed networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 369–377.

[65] Rong-Hua Li, Jeffrey Xu Yu, and Jianquan Liu. 2011. Link prediction: the power of maximal entropy random walk. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 1147–1156.

[66] Xiaoyi Li, Nan Du, Hui Li, Kang Li, Jing Gao, and Aidong Zhang. 2014. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 289–297.

[67] David Liben-Nowell. 2005. *An algorithmic approach to social networks*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[68] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.

[69] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. 2010. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 243–252.

[70] Xueming Liu, H Eugene Stanley, and Jianxi Gao. 2016. Breakdown of interdependent directed networks. *Proceedings of the National Academy of Sciences* 113, 5 (2016), 1138–1143.

[71] Linyuan Lü, Ci-Hang Jin, and Tao Zhou. 2009. Similarity index based on local paths for link prediction of complex networks. *Physical Review E* 80, 4 (2009), 046122.

[72] Linyuan Lü and Tao Zhou. 2010. Link prediction in weighted networks: The role of weak ties. *EPL (Europhysics Letters)* 89, 1 (2010), 18001.

[73] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390, 6 (2011), 1150–1170.

[74] Sagar Maheshwari Marinka Zitnik, Rok Sosič and Jure Leskovec. 2018. BioSNAP Datasets: Stanford Biomedical Network Dataset Collection. http://snap.stanford.edu/biodata.

[75] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2017. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)* 49, 4 (2017), 69.

[76] Bruce McCune, James B Grace, and Dean L Urban. 2002. *Analysis of ecological communities*. Vol. 28. MjM software design Gleneden Beach, OR.

[77] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 29–42.

[78] James Moody, Daniel McFarland, and Skye Bender-deMoll. 2005. Dynamic network visualization. *American journal of sociology* 110, 4 (2005), 1206–1241.

[79] Peter Mucha and Mason Porter. 2013. Netwiki Shared Data. http://netwiki.amath.unc.edu/SharedData/SharedData.

[80] Mark Newman. 2013. Mark Newman Network Datasets Collection. http://www-personal.umich.edu/~mejn/netdata.

[81] Mark EJ Newman. 2001. Clustering and preferential attachment in growing networks. *Physical review E* 64, 2 (2001), 025102.

[82] Mark EJ Newman. 2003. The structure and function of complex networks. *SIAM review* 45, 2 (2003), 167–256.

[83] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1105–1114.

[84] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. Fast and accurate link prediction in social networking systems. *Journal of Systems and Software* 85, 9 (2012), 2119–2132.

[85] Karl Pearson. 1905. The problem of the random walk. *Nature* 72, 1867 (1905), 342.

[86] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[87] Erzsébet Ravasz and Albert-László Barabási. 2003. Hierarchical organization in complex networks. *Physical Review E* 67, 2 (2003), 026112.

[88] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. 2002. Hierarchical organization of modularity in metabolic networks. *science* 297, 5586 (2002), 1551–1555.

[89] Jörg Reichardt and Stefan Bornholdt. 2006. Statistical mechanics of community detection. *Physical Review E* 74, 1 (2006), 016110.

[90] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. http://networkrepository.com

[91] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).

[92] Tom AB Snijders. 2001. The statistical evaluation of social network dynamics. *Sociological methodology* 31, 1 (2001), 361–395.

[93] Han Hee Song, Tae Won Cho, Vacha Dave, Yin Zhang, and Lili Qiu. 2009. Scalable proximity estimation and link prediction in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 322–335.

[94] Thorvald Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* 5 (1948), 1–34.

[95] Daniel A Spielman. 2007. Spectral graph theory and its applications. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*. IEEE, 29–38.

[96] Jaideep Srivastava. 2008. Data mining for social network analysis. In *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*. IEEE, xxxiii–xxxiv.

[97] Stefan Stieglitz and Linh Dang-Xuan. 2013. Emotions and information diffusion in social mediaâĂŤsentiment of microblogs and sharing behavior. *Journal of management information systems* 29, 4 (2013), 217–248.

[98] Qingshuang Sun, Rongjing Hu, Zhao Yang, Yabing Yao, and Fan Yang. 2017. An improved link prediction algorithm based on degrees and similarities of nodes. In *Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on*. IEEE, 13–18.

[99] Fei Tan, Yongxiang Xia, and Boyao Zhu. 2014. Link prediction in complex networks: a mutual information perspective. *PloS one* 9, 9 (2014), e107056.

[100] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[101] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD'08*. 990–998.

[102] Ben Taskar, Pieter Abbeel, Ming-Fai Wong, and Daphne Koller. 2007. Relational markov networks. *Introduction to statistical relational learning* (2007), 175–200.

[103] Mike Thelwall, David Wilkinson, and Sukhvinder Uppal. 2010. Data mining emotion in social network communication: Gender differences in MySpace. *Journal of the American Society for Information Science and Technology* 61, 1 (2010), 190–199.

[104] Hanghang Tong, Christos Faloutsos, Christos Faloutsos, and Yehuda Koren. 2007. Fast direction-aware proximity for graph mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 747–756.

[105] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the Evolution of User Interaction in Facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*.

[106] Caroline S Wagner and Loet Leydesdorff. 2005. Mapping the network of global science: comparing international co-authorships from 1990 to 2000. *International journal of Technology and Globalisation* 1, 2 (2005), 185–208.

[107] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2017. Graphgan: Graph representation learning with generative adversarial nets. *arXiv preprint arXiv:1711.08267* (2017).

[108] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. 2015. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* 58, 1 (2015), 1–38.

[109] Xu-Wen Wang, Yize Chen, and Yang-Yu Liu. 2018. Link Prediction through Deep Learning. *bioRxiv* (2018), 247577.

[110] Stanley Wasserman and Katherine Faust. 1994. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press.

[111] Joshua S Weitz, Philip N Benfey, and Ned S Wingreen. 2007. Evolution, interactions, and biological networks. *PLoS biology* 5, 1 (2007), e11.

[112] Mingrui Xia, Jinhui Wang, and Yong He. 2013. BrainNet Viewer: a network visualization tool for human brain connectomics. *PloS one* 8, 7 (2013), e68910.

[113] Juan Yang, Lixin Yang, and Pengye Zhang. 2015. A New Link Prediction Algorithm Based on Local Links. In *International Conference on Web-Age Information Management*. Springer, 16–28.

[114] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning withDifferentiable Pooling. *arXiv preprint arXiv:1806.08804* (2018).

[115] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*. 5694–5703.

[116] Kai Yu, Wei Chu, Shipeng Yu, Volker Tresp, and Zhao Xu. 2007. Stochastic relational models for discriminative link prediction. In *Advances in neural information processing systems*. 1553–1560.

[117] Kai Yu, Wei Chu, Shipeng Yu, Volker Tresp, and Zhao Xu. 2007. Stochastic Relational Models for Discriminative Link Prediction. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman (Eds.). MIT Press, 1553–1560. http://papers.nips.cc/paper/2998-stochastic-relational-models-for-discriminative-link-prediction.pdf

[118] Wayne W Zachary. 1977. An information flow model for conflict and fission in small groups. *Journal of anthropological research* 33, 4 (1977), 452–473.

[119] R. Zafarani and H. Liu. 2009. Social Computing Data Repository at ASU. http://socialcomputing.asu.edu

[120] Jianpei Zhang, Yuan Zhang, Hailu Yang, and Jing Yang. 2014. A link prediction algorithm based on socialized semi-local information. *Journal of Computational Information Systems* 10, 10 (2014), 4459–4466.

[121] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. *arXiv preprint arXiv:1802.09691* (2018).

[122] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B* 71, 4 (2009), 623–630.

[123] Yu-Xiao Zhu, Linyuan Lü, Qian-Ming Zhang, and Tao Zhou. 2012. Uncovering missing links with cold ends. *Physica A: Statistical Mechanics and its Applications* 391, 22 (2012), 5769–5778.