Chapter 3

Link Prediction Techniques, Applications, and Performance: A Survey

In this chapter [1], we exhibit a review of previous methodologies shedding light on link prediction with the point of convergence mostly on social network graphs. We order these methodologies into several categories; one category of those calculates a similarity score between pairs of vertices in which higher scored pairs are assumed to have links between them. Another category of algorithms is based on probabilistic approaches in which Bayesian and relational models have been used. Dimensionality reduction approaches consisting of embedding and factorization-based methods have grouped into one, and some other approaches also have been studied.

Recently, numerous methodologies of link prediction have been implemented. These methods can be grouped into several categories, like similarity-based, probabilistic models, learning-based models, etc.

[1] Published in Physica A: Statistical Mechanics and its Applications

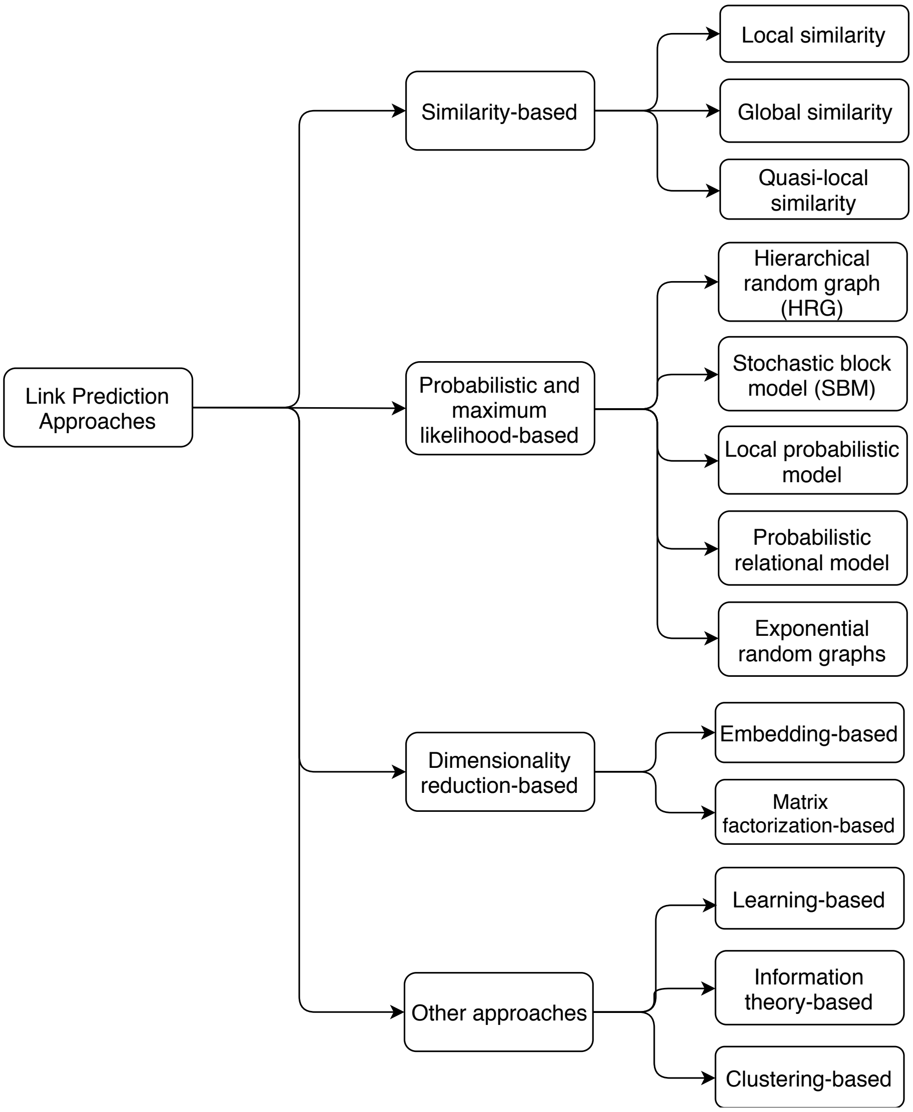
Link Prediction Approaches
Similarity-based
Probabilistic and maximum likelihood-based
Dimensionality reduction-based
Other approaches
Local similarity
Global similarity
Quasi-local similarity
Hierarchical random graph (HRG)
Stochastic block model (SBM)
Local probabilistic model
Probabilistic relational model
Exponential random graphs
Embedding-based
Matrix factorization-based
Learning-based
Information theory-based
Clustering-based

FIGURE 3.1: Taxonomy of Link Prediction Approaches

3.1 Similarity-based methods

Similarity-based metrics are the simplest one in link prediction, in which for each pair x and y, a similarity score $S(x,y)$ is calculated. The score $S(x,y)$ is based on the structural or node's properties of the considered pair. The non-observed links (i.e., $U - E^T$) are assigned scores according to their similarities. The pair of nodes having a higher score represents the predicted link between them. The similarity measures between every pair can be calculated using several properties of the network, one of which is structural property. Scores based on this property can be grouped in several categories like local and global, node-dependent and path-dependent, parameter-dependent and parameter-free, and so on.

3.1.1 Local similarity indices

Local indices are generally calculated using information about common neighbors and node degree. These indices consider immediate neighbors of a node. Examples of such indices contains common neighbor [56], preferential attachment [57], Adamic/Adar [58], resource allocation [59], etc.

3.1.1.1 Common neighbors (CN)

In a given network or graph, the size of common neighbors for a given pair of nodes x and y is calculated as the size of the intersection of the two nodes neighborhoods [56].

$$S(x,y) = |\Gamma(x) \cap \Gamma(y)|, \tag{3.1}$$

where $\Gamma(x)$ and $\Gamma(y)$ are neighbors of the node x and y respectively. The likelihood of the existence of a link between x and y increases with the number of common neighbors between them. In a collaboration network, Newman calculated this quantity and demonstrated that the probability of collaboration between two nodes depends upon the common neighbors of the selected nodes. Kossinets and Watts [60, 61] investigated a large social network and recommended that two students are more likely to be friends who are having numerous common friends. It has been observed that the common neighbor approach performs well on most real-world networks and beats other complex methods.

3.1.1.2 Jaccard coefficient (JC)

This metric [62] is similar to the common neighbor. Additionally, it normalizes the above score, as given below.

$$S(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \tag{3.2}$$

i.e., the Jaccard coefficient is defined as the probability of selection of common neighbors of pairwise vertices from all the neighbors of either vertex. The pairwise Jaccard score increases with the number of common neighbors between the two vertices considered. Liben-Nowell et al. [17] demonstrated that this similarity metric performs worse as compared to Common Neighbors.

3.1.1.3 Adamic/Adar index (AA)

Adamic and Adar [58] presented a metric to calculate a similarity score between two web pages based on shared features, which are further used in link prediction after some modification by Liben-Nowell et al. [17].

$$S(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}, \tag{3.3}$$

where k_z is the degree of the node z. It is clear from the equation that more weights are assigned to the common neighbors having smaller degrees. This is also intuitive in the real-world scenario, for example, a person with more number of friends spend less time/resource with an individual friend as compared to the less number of friends.

3.1.1.4 Preferential attachment (PA)

The idea of preferential attachment is applied to generate a growing scale-free network. The term growing represents the incremental nature of nodes over time in the network.

The likelihood incrementing new connection associated with a node x is proportional to k_x, the degree of the node. Preferential attachment score between two nodes x and y can be computed as [57]

$$S(x,y) = k_x.k_y. \tag{3.4}$$

This index shows the worst performance on most networks, as reported in the result section. The simplicity (as it requires the least information for the score calculation) and the computational time of this metric are the main advantages. Also, it can be used in a non-local context as it requires only degree as information and not the common neighbors. In assortative networks, the performance of the PA improves, while very bad for disassortative networks. In other words, PA shows better results if larger degree nodes are densely connected, and lower degree nodes are rarely connected.

In a supervised learning framework, Hasan et al. [63] showed that aggregate functions (e.g., sum, multiplication, etc.) over feature values of vertices could be applied to compute link feature value. In the above equation, summation can also be used instead of multiplication as an aggregate function, and in fact, it has been proved to be quite useful. [63] showed the preferential attachment with aggregate function "sum" performs well for the link prediction in coauthorship network.

3.1.1.5 Resource allocation Index (RA)

The original dynamics of this similarity index is originated from Ou et al. [64] work published in "Physical Review E" on resource allocation dynamics on complex networks. Consider two non-adjacent vertices x and y. Suppose node x sends some resources to y through the common nodes of both x and y then the similarity between the two vertices is computed in terms of resources sent from x to y. This is expressed mathematically as [59]

$$S(x,y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{k_z}. \tag{3.5}$$

This similarity measure and the Adamic/Adar are very similar to each other, as shown by the equations 3.5 and 3.3, respectively. The difference is that the RA index heavily penalizes to higher degree nodes compared to the AA index. Prediction results of these indices become almost the same for smaller average degree networks. This index shows good performance on heterogeneous networks with a high clustering coefficient, especially on transportation networks (e.g., USAir97 as reported in the result section).

3.1.1.6 Cosine similarity or Salton index (SI)

In a vector space, document similarities can be computed using the Salton index [65], also known as Cosine similarity. This similarity index between two records (documents) is measured by calculating the Cosine of the angle between them. The metric is all about the orientation and not magnitude. The Cosine similarity [65] can be computed as

$$S(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{(k_x.k_y)}}. \tag{3.6}$$

3.1.1.7 Sorensen index

This index [66] of similarity was applied mainly to the ecological data samples and given by Thorvald Sorensen in 1948. It is very similar to the Jaccard index, as we can observe in the equation 3.7. McCune et al. show that it is more robust than Jaccard against the outliers [67].

$$S(x,y) = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k_x + k_y}. \tag{3.7}$$

3.1.1.8 CAR-based common neighbor index (CAR)

CAR-based indices are presented based on the assumption that the link existence between two nodes is more likely if their common neighbors are members of a local community (local-community-paradigm (LCP) theory) [68]. In other words, the likelihood existence

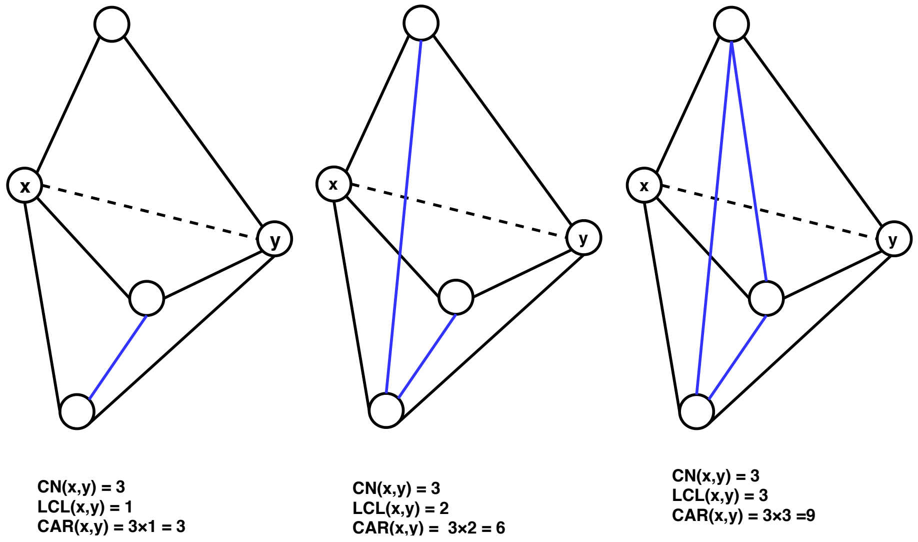

FIGURE 3.2: *CAR Index* $=$ (*Number of CNs*) $\times$ (*Number of LCLs*)

increases with the number of links among the common neighbors (local community links (LCLs)) of the seed node pair as described in Figure 3.2.

$$\begin{aligned} S(x,y) &= CN(x,y) \times LCL(x,y) \\ &= CN(x,y) \times \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{|\gamma(z)|}{2}, \end{aligned} \tag{3.8}$$

where $CN(x,y) = |\Gamma(x) \cap \Gamma(y)|$ is number of common neighbors $LCL(x,y)$ refers to the number of local community links which are defined as the links among the common neighbors of seed nodes x and y [68]. $\gamma(z)$ is the subset of neighbors of node z that are also common neighbors of x and y.

3.1.1.9 Hub promoted index (HPI)

Ravasz et al. [15] published a paper on a cellular organization in metabolic networks. They show that the metabolic networks are composed of several small and highly connected topological modules and are combined into larger and less cohesive hierarchical structures. The number of such modules and their degree of clustering follow the power law. This similarity index promotes the formation of links between the sparsely connected nodes and hubs. It also tries to prevent links formation between the hub nodes. This similarity metric can be expressed mathematically as

$$S(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min(k_x, k_y)}. \tag{3.9}$$

3.1.1.10 Hub depressed index (HDI)

This index is the same as the previous one but with the opposite goal as it avoids the formation of links between hubs and low degree nodes in the networks. The Hub depressed index promotes the links evolution between the hubs as well as the low degree nodes. The mathematical expression for this index [15] is given below.

$$S(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max(k_x, k_y)}. \tag{3.10}$$

3.1.1.11 Local naive Bayes-based common neighbors (LNBCN)

The above similarity indices are somehow based on common neighbors of the node pair where each of the which are equally weighted. This method [69] is based on the Naive Bayes theory and arguments that different common neighbors play different role in the network and hence contributes differently to the score function computed for non-observed node pairs.

$$S(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} [\log(\frac{C(z)}{1-C(z)}) + \log(\frac{1-\rho}{\rho})], \tag{3.11}$$

where $C(z)$ is node clustering coefficient and ρ is the network density expressed as

$$\rho = \frac{m}{n(n-1)/2}.$$

3.1.1.12 Leicht-Holme-Newman local index (LHNL)

Leicht et al. [70] presented a paper on vertex similarity in networks. Their work is based on the concept of self-similarity, i.e., two vertices are similar to each other if their corresponding neighbors are self-similar to themselves. This score is defined by the ratio of the path of length two that exits between two vertices and the expected path of the same length between them.

$$S(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{k_x.k_y}. \tag{3.12}$$

3.1.1.13 Node clustering coefficient (CCLP)

This index [35] is also based on the clustering coefficient property of the network in which the clustering coefficients of all the common neighbors of a seed node pair are computed and summed to find the final similarity score of the pair. Mathematically, this index can be expressed as follows

$$S(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} C(z), \tag{3.13}$$

where

$$C(z) = \frac{t(z)}{k_z(k_z-1)}$$

is clustering coefficient of the node z and $t(z)$ is the total triangles passing through the node z.

3.1.1.14 Node and link clustering coefficient (NLC)

This similarity index [71] is based on the basic topological feature of a network called ”Clustering Coefficient” [11, 12]. The clustering coefficients of both nodes and links are incorporated to compute the similarity score.

$$S(x,y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{|\Gamma(x)\cap\Gamma(z)|}{k_z - 1} \times C(z) + \frac{|\Gamma(y)\cap\Gamma(z)|}{k_z - 1} \times C(z), \tag{3.14}$$

3.1.2 Global similarity indices

Global indices are computed using entire topological information of a network. The computational complexities of such methods are higher and seem to be infeasible for large networks.

3.1.2.1 Katz index

This index [55] can be considered as a variant of the shortest path metric. It directly aggregates over all the paths between x and y and dumps exponentially for longer paths to penalize them. It can be expressed mathematically as

$$S(x,y) = \sum_{l=1}^{\infty} \beta^l |paths_{x,y}^{<l>}| = \sum_{l=1}^{\infty} \beta^l (A^l)_{x,y}, \tag{3.15}$$

where, $paths_{x,y}^{<l>}$ is considered as the set of total l length paths between x and y, β is a damping factor that controls the path weights and A is the adjacency matrix. For the convergence of above equation,

$$\beta < \frac{1}{\lambda_1},$$

where λ_1 is the maximum eigen value of the matrix A. If 1 is added to each element of the diagonal of the resulting similarity matrix S, this expression can be written in matrix terms as

$$S = \beta AS + I, \tag{3.16}$$

where I is the identity matrix of the proper dimension. The similarity between all pairs of nodes can be directly computed using the closed-form by rearranging for S in the previous expression and subtracting the previously added 1 to the elements in the diagonal. Katz score for each pair of nodes in the network is calculated by finding the similarity matrix as

$$S = (I - \beta A)^{-1} - I. \tag{3.17}$$

The computational complexity of the given metric is high, and it can be roughly estimated to be cubic complexity which is not feasible for a large network.

3.1.2.2 Random walk with restart (RWR)

Let α be a probability that a random walker iteratively moves to an arbitrary neighbor and returns to the same starting vertex with probability $(1-\alpha)$. Consider q_{xy} to be the probability that a random walker who starts walking from vertex x and located at the vertex y in steady-state. Now, this probability of walker to reach the vertex y is expressed mathematically as [72]

$$\vec{q_x} = \alpha P^T \vec{q_x} + (1-\alpha)\vec{e_x}, \tag{3.18}$$

where $\vec{e_x}$ is the seed vector of length $|V|$ (i.e., the total number of vertices in the graph). This vector consists of zeros for all components except the elements x itself. The transition matrix P can be expressed as

$$P_{xy} = \begin{cases} \frac{1}{k_x} & \text{if x and y are connected,} \\ 0 & \text{otherwise.} \end{cases}$$

Simplifying the above equation we get,

$$\vec{q}_x = (1-\alpha)(I-\alpha P^T)^{-1}\vec{e}_x. \tag{3.19}$$

Since this similarity is not symmetric, the final score between the node pair (x,y) can be computed as

$$S(x,y) = q_{xy} + q_{yx}. \tag{3.20}$$

It is clear from the equation 3.19 that matrix inversion is required to solve, which is quite expensive and prohibitive for large networks. A faster version of this index is implemented in [72].

3.1.2.3 Shortest path

Lots of algorithms [73–75] are available to compute the shortest path between a vertex pair in a graph that applies to a different scenario. Liben-Nowell et al. [17] provided the shortest path with its negation as a metric to link prediction. The inverse relation between the similarity and length of the shortest path is captured by the following mathematical equation given below [17].

$$S(x,y) = -|d(x,y)|, \tag{3.21}$$

where Dijkstra algorithm [73] is applied to efficiently compute the shortest path $d(x,y)$ between the node pair (x,y). The prediction accuracy of this index is low compared to most local indices that make room for the consideration of indirect path in link prediction.

Several paths of different lengths can exist between a vertex pair, the similarity between such pair is computed by several other methods like Katz index, local path index, etc.

3.1.2.4 Leicht-Holme-Newman global index (LHNG)

This global index, proposed by Leicht et al. [70], is based on the principle that two nodes are similar if either of them has an immediate neighbor, which is similar to the other node. This is a recursive definition of similarity where a termination condition is needed. The termination condition is introduced in terms of self-similarity, i.e., a node is similar to itself. Thus, the similarity score equation consists of two terms: first, the neighbor similarity, and the second, self-similarity, as given below.

$$S(x,y) = \phi \sum_z A_{x,z} S_{z,y} + \psi \delta_{x,y}. \tag{3.22}$$

Here, the first term is neighborhood similarity and the second term is self-similarity. ϕ and ψ are free parameters that make a balance between these two terms. In matrix form [18, 45]

$$\begin{aligned} S &= \phi AS + \psi I = \psi (I - \phi A)^{-1} \\ &= \psi (I + \phi A + \phi^2 A^2 + \ldots) \end{aligned} \tag{3.23}$$

When the free parameter $\psi = 1$, this index resembles to the Katz index [55]. Moreover, we note that $A^1(x,y)$, $A^2(x,y)$, etc, represent number of paths of length 1, 2, and so on respectively. After some calculation, the final similarity score can be expressed as given below [70].

$$S = 2m\lambda_1 D^{-1} (I - \frac{\alpha}{\lambda_1} A)^{-1} D^{-1}, \tag{3.24}$$

where D is the diagonal matrix, and β is dumping factor that penalizes the longer path contribution. Dropping the constant term $2m\lambda_1$ and rearranging the equation 3.24, it becomes

$$DSD = \frac{\beta}{\lambda_1} A(DSD) + I. \tag{3.25}$$

The equation 3.25 solved by iterating this equation repeatedly with the initial value of $(DSD) = 0$ and converges normally in 100 iterations as claimed by the authors [70].

3.1.2.5 Cosine based on L^+ (Cos^+)

Laplacian matrix is extensively used as an alternative representation of graphs in spectral graph theory [76]. This matrix can be defined as $L = D - A$, where, D is the diagonal matrix consisting of the degrees of each node of the matrix and A is the adjacency matrix of the graph. The Pseudoinverse of the matrix defined by Moore-Penrose is represented as L^+ and each entry of this matrix is used to represent the similarity score [77] between the two corresponding nodes. The most common way to compute this Pseudoinverse is by computing the singular value decomposition (SVD) of the Laplacian matrix $[(L = \mathscr{U}\Sigma\mathscr{V}^{\mathscr{T}})$, where $\mathscr{U}$ and $\mathscr{V}$ are left and right singular vectors of SVD] as follows

$$L^+ = \mathscr{V}\Sigma^+\mathscr{U}^{\mathscr{T}}, \tag{3.26}$$

Σ^+ is obtained by taking the inverse of each nonzero element of the Σ. Further, the similarity between two nodes x and y can be computed using any inner product measure such as Cosine similarity given as

$$S(x,y) = \frac{L^+_{x,y}}{\sqrt{L^+_{x,x}L^+_{y,y}}}. \tag{3.27}$$

3.1.2.6 Average commute time (ACT)

This index is based on the random walk concept. A random walk is a Markov chain [78, 79] which describes the movements of a walker. ACT is first coined by Gob̈el and Jagers [80] and applied in link prediction by [81]. It defined as the average number of movements/steps required by a random walker to reach the destination node y, and come back to the starting node x. If $m(x,y)$ be the number of steps required by the walker to reach y from x, then the following expression captures this concept.

$$n(x,y) = m(x,y) + m(y,x). \tag{3.28}$$

This equation can be simplified using the Pseudoinverse of the Laplacian matrix L^+ [77, 82] as

$$n(x,y) = |E|(l^+_{xx} + l^+_{yy} - 2l^+_{xy}), \tag{3.29}$$

where l^+_{xy} denotes the (x,y) entry of the matrix L^+. Pseudoinverse of the Laplacian, L^+ can be computed as [77]

$$L^+ = (L - \frac{ee^T}{n})^{-1} + \frac{ee^T}{n}, \tag{3.30}$$

where e is a column vector consisting of 1's. The square root of the equation 3.29 is called Euclidean commute time distance (ECTD) [77], so smaller value of this equation will represent higher similarity. The final expression representing this similarity index is thus, given by the squared reciprocal of the equation 3.29 and by ignoring the constant term $|E|$.

$$S(x,y) = \frac{1}{l^+_{xx} + l^+_{yy} - 2l^+_{xy}}. \tag{3.31}$$

3.1.2.7 Matrix forest index (MF)

This index is based on the concept of spanning tree which is defined as the subgraph that spans total nodes without forming any cycle. The spanning tree may contain total or less number of links as compared to the original graph. Chebotarev and Shamis proposed a theorem called matrix-forest theorem [83] which states that the number of spanning tree in a graph is equal to the cofactor of any entry of Laplacian matrix of the graph. Here, the term forest represents the union of all rooted disjoint spanning trees. The similarity between two nodes x and y can be computed with the equation 3.32 given below.

$$S = (I + L)^{-1}, \tag{3.32}$$

where $(I+L)_{(x,y)}$ is the number of spanning rooted forests (x as root) consisting of both the nodes x and y. Moreover, this quantity is equal to the cofactor of $(I+L)_{(x,y)}$.

3.1.2.8 SimRank (SR)

SimRank [84] is a measure of structural context similarity and shows object-to-object relationships. It is not domain-specific and recommends to apply in directed or mixed networks. The basic assumption of this measure is that two objects are similar if they are related to similar objects. SimRank computes how soon two random walkers meet each other, starting from two different positions. The measure is computed recursively using the equation 3.33.

$$S(x,y) = \begin{cases} \frac{\alpha}{k_x k_y} \sum_{i=1}^{k_x} \sum_{j=1}^{k_y} S(\Gamma_i(x), \Gamma_j(y)) & \text{x} \neq \text{y} \\ 1 & \text{x = y} \end{cases} \tag{3.33}$$

where, $\alpha \in (0,1)$ is a constant. $\Gamma_i(x)$ and $\Gamma_j(y)$ are the i^{th} and j^{th} elements in the neighborhood sets $\Gamma(x)$ and $\Gamma(y)$ respectively. Initially, $S(x,y) = A(x,y)$, i.e., $S(x,x) = 1$ and $S(x,y) = 0$ for $x \neq y$. This measure can be represented in matrix form as

$$S(x,y) = \alpha W^T S W + (1-\alpha) I, \tag{3.34}$$

where W is the transformation matrix and computed by normalizing each column of adjacency matrix A as $W_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n}}$.

The computational complexity of this measure is high for a large network, and to reduce its time, the authors [84] suggest pruning recursive branches after radius l. The time required to compute this score for each pair is $O(k^{2l+2})$, and total time is $O(n^2 k^{2l+2})$.

3.1.2.9 Rooted PageRank (RPR)

The idea of PageRank [40] was originally proposed to rank the web pages based on the importance of those pages. The algorithm is based on the assumption that a random walker randomly goes to a web page with probability α and follows hyper-link embedded in the page with probability $(1-\alpha)$. Chung et al. [85] used this concept incorporated with a random walk in link prediction framework. The importance of web pages, in a random walk, can be replaced by stationary distribution. The similarity between two vertices x and y can be measured by the stationary probability of y from x in a random walk where the walker moves to an arbitrary neighboring vertex with probability α and returns to x with probability $(1-\alpha)$. Mathematically, this score can be computed for all pair of vertices as

$$RPR = (1-\alpha)(I-\alpha\hat{N})^{-1}, \tag{3.35}$$

where $\hat{N} = D^{-1}A$ is the normalized adjacency matrix with the diagonal degree matrix $D[i,i] = \sum_j A[i,j]$.

3.1.3 Quasi-local indices

Quasi-local indices have been introduced as a trade-off between local and global approaches or performance and complexity, as shown in Table 3.1. These metrics are as efficient to compute as local indices. Some of these indices extract the entire topological information of the network. The time complexities of these indices are still below compared to the global approaches. Examples of such indices include local path index, local random walk index [81], local directed path (LDP) [86], etc.

3.1.3.1 Local path index (LP)

With the intent to furnish a good trade-off between accuracy and computational complexity, the local path-based metric is considered [53]. The metric is expressed mathematically as

$$S^{LP} = A^2 + \varepsilon A^3, \tag{3.36}$$

where ε represents a free parameter. Clearly, the measurement converges to common neighbor when $\varepsilon = 0$. If there is no direct connection between x and y, $(A^3)_{xy}$ is equated to the total different paths of length 3 between x and y. The index can also be expanded to generalized form

$$S^{LP} = A^2 + \varepsilon A^3 + \varepsilon^2 A^4 + \ldots + \varepsilon^{(n-2)} A^n, \tag{3.37}$$

where n is the maximal order. Computing this index becomes more complicated with the increasing value of n. The LP index [53] outperforms the proximity-based indices, such as RA, AA, and CN.

3.1.3.2 Path of length 3 (L3)

Georg Simmel, a German sociologist, first coined the concept “triadic closure” and made popular by Mark Granovetter in his work [87] “The Strength of Weak Ties”. The authors [54] proposed a similarity index in protein-protein interaction (PPI) network, called path of length 3 (or $L3$) published in the Nature Communication. They experimentally show that the triadic closure principle (TCP) does not work well with PPI networks. They showed the paradoxical behavior of the TCP (i.e., the path of length 2), which does not follow the structural and evolutionary mechanism that governs protein interaction. The TCP predicts well to the interaction of self-interaction proteins (SIPs), which are very small (4%) in PPI networks and fails in prediction between SIP and non SIP that amounts to 96%. They showed that the $L3$ index performs well in such conditions and give mathematical expression to compute this index [54] as

$$S(x,y) = \sum_{u,v} \frac{a_{x,u}.a_{u,v}.a_{v,y}}{\sqrt{k_u.k_v}}. \tag{3.38}$$

Recently, Pech et al. [88] in Physica A, proposed a work that models the link prediction as a linear optimization problem. They introduced a theoretical explanation of how direct count of paths of length 3 significantly improves the prediction accuracy. Meanwhile, some more studies [89, 90] focusing the length of path have been proposed in the literature. Muscoloni et al. [89] incorporate the concept of local community paradigm (LCP) with 2 and 3 length paths and introduced new similarity indices viz., Cannistraci-Hebb indices $CH2-L2$ and $CH2-L3$ corresponding to them. These indices are based on the common neighbor‘s rewards to internal local community links (iLCL) and penalization to external local community links (eLCL) [89]. The mathematical expression to compute these two similarity indices are as follows.

$$S^{CH2-L2}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1+c_z}{1+o_z}, \tag{3.39}$$

where c_z are total number of neighbors of z which are also members of $(\Gamma(x) \cap \Gamma(y))$ and o_z are those neighbors counting that are not in $(\Gamma(x) \cap \Gamma(y))$, also not x or y.

$$S^{CH2-L3}(x,y) = \sum_{z1 \in \Gamma(x), z2 \in \Gamma(y)} \frac{a_{z1,z2}\sqrt{(1+\tilde{c_{z1}})(1+\tilde{c_{z2}})}}{\sqrt{(1+\tilde{o_{z1}})(1+\tilde{o_{z2}})}}. \tag{3.40}$$

Here, $a_{z1,z2}$ is 1 when there is link between $z1$ and $z2$, 0, otherwise. $\tilde{c_{z1}}$ is the number of links between $z1$ and the set of intermediate nodes on all 3 length paths between x and y. Similarly, $\tilde{o_{z1}}$ is the number of links between $z1$ and nodes that are not in the set of intermediate nodes of any 3 length path between x and y, also not x or y.

3.1.3.3 Similarity based on local random walk and superposed random walk (LRW and SRW)

Liu and Lü [81] proposed new similarity measures by exploiting the random walk concept on graphs with limited walk steps. They defined node similarity based on random walks of lower computational complexity compared to the other random walk based methods [72, 81]. Given a random walker, starting from the node x, the probability of reaching the random walker to the node y in t steps is

$$\vec{\pi}_x(t) = P^T \vec{\pi}_x(t-1), \tag{3.41}$$

where $\vec{\pi}_x(0)$ is a column vector with x^{th} element as 1 while others are 0's and P^T is the transpose of the transition probability matrix P. P_{xy} entry of this matrix defines the probability of a random walker at node x will move to the next node y. It is expressed as $P_{xy} = \frac{a_{xy}}{k_x}$, where a_{xy} is 1 when there is a link between x and y and 0, otherwise. The authors computed the similarity score (LRW) between two nodes based on the above concept as

$$S^{LRW}(x,y) = \frac{k_x}{2|E|}\pi_{xy}(t) + \frac{k_y}{2|E|}\pi_{yx}(t). \tag{3.42}$$

This similarity measure focus on only few steps covered by the random walker (hence quasi-local) and not the stationary state compared to other approaches [72, 81].

Random walk based methods suffer from the situation where a random walker moves far away with a certain probability from the target node whether the target node is closer or not. This is an obvious problem in social networks that show a high clustering index i.e., clustering property of the social networks. This degrades the similarity score between the two nodes and results in low prediction accuracy. One way to counter this problem is that continuously release the walkers at the starting point, which results in a higher similarity between the target node and the nearby nodes. By superposing the contribution of each

TABLE 3.1: Comparison of similarity-based approaches

Properties	Local Indices	Global Indices	Quasi-local Indices
Nature	Simple	Complex	Moderate
Features employed	Local neighborhood	Entire network	More local neighborhood
Computational complexity	Low	high	Moderate
Parallelization	Easy	More complex	Moderate
Implementation	Feasible for large networks	Feasible for small networks	Feasible for large networks

walker (walkers move independently), SRW is expressed as

$$S^{SRW}(x,y)(t) = \sum_{l=1}^{t} S^{LRW}(l), \tag{3.43}$$

Remarks Similarity-based approaches mostly focus on the structural properties of the networks to compute the similarity score. Local approaches consider, in general, neighborhood information (direct neighbors or neighbors of neighbor), which take less time for computation. This is the property that makes the local approaches feasible for massive real-world network datasets. Global approaches consider the entire structural information of the network; that is why time required to capture this information is more than local and quasi-local approaches. Also, sometimes, entire topological information may not be available at the time of computation, especially in a decentralized environment. So, parallelization over the global approaches may not possible or very complex compared to the local and quasi-local approaches. Quasi-local approaches extract more structural information than local and somehow less information compared to the global. Table 3.1 shows a simple comparison among similarity-based approaches to link prediction.

3.2 Probabilistic and maximum likelihood models

For a given network $G(V,E)$, the probabilistic model optimizes an objective function to set up a model that is composed of several parameters. Observed data of the given network can be estimated by this model nicely. At that point, the likelihood of the presence of a non-existing link (i, j) is evaluated using conditional probability $P(A_{ij} = 1|\theta)$. Several probabilistic models [91–93] and maximum likelihood models [1, 2] have been proposed in the literature to infer missing links in the networks. The probabilistic models normally require more information like node or edge attribute knowledge in addition to structural information. Extracting these attribute information is not easy; moreover, the parameter tuning is also a big deal in such models that limit their applicability. Maximum likelihood methods are complex and time-consuming, so these models are not suitable for real large networks. Some seminal probabilistic and maximum likelihood models are tabulated in the Table 3.2 [3].

3.2.1 Local probabilistic model for link prediction

Wang et al. [91] proposed a local probabilistic model for link prediction in an undirected network. They employed three different types of features viz., topological, semantic, and co-occurrence probability features extracted from different sources of information. They presented an idea of a central neighborhood set derived from the local topology of the considered node-pair, which is relevant information for the estimation of a link between them. They computed non-derivable frequent itemsets (i.e., those itemsets whose occurrence statistics can not be derived from other itemset patterns) from the network events log data, which is further used as training data for the model. An event corresponds to a publication of a paper (i.e., authors' interactions in the paper is a an event, and a set

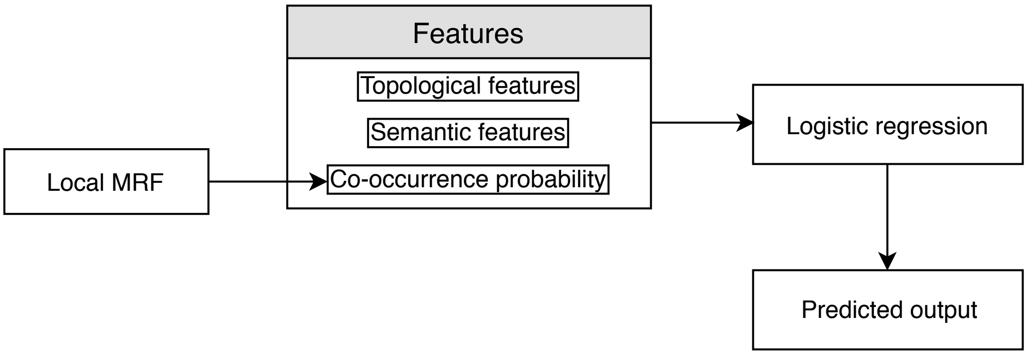

FIGURE 3.3: Local probabilistic model for link prediction [3]

of such events is the event log) in the Coauthorship network. The event log consists of transactional[2]. data upon which frequent itemset mining approaches [94–98] are applied. The model [91] is shown in Figure 3.3, which considers the following approach given below.

First, the central neighborhood set between x and y is calculated based on local event log data. One of the usual ways to find the central neighborhood set is to find the shortest path between two vertices of specified length, and the vertices are lying on this path can be included in the required set. There can be more than one shortest path between two vertices, so more neighborhood sets can be possible. Neighborhood sets of shorter lengths and more frequent (frequency score is used when more shortest paths of the same length are available) are chosen for the central neighborhood set. The authors considered the shortest path up to length 4 since the nodes lying on the shorter length path are more relevant.

In the second step, for a given central neighborhood set, non-derivable frequent itemsets are used to learn the local probabilistic model. Calders et al. [99] proposed a depth-first search method to calculate non-derivable itemsets and the same algorithm used by the authors [91]. [Why non-derivable frequent itemsets? Pavlov et al. [100] first introduced the concept of frequent itemset to construct an MRF [101]. They argued that a $\mathscr{K}$-itemset and its support represents a $\mathscr{K}$-way statistics, which can be viewed as a constraint on the true underlying distribution that generates the data. Given a set of itemset constraints, a maximum entropy distribution satisfying all these constraints is selected as the estimate for the true underlying distribution. This maximum entropy distribution is equivalent to an MRF. Since the number formed links are very few compared to all possible links in a sparse network, the authors [91] used a support threshold of one to extract all frequent itemsets. Theses extracted itemsets are large in number that results in expensive learning for the MRF. To reduce this cost, only non-derivable itemsets are extracted]. They find all

[2] Typically, social networks are the results of evolution of chronological sets of events (e.g., authors participation in the Coauthorship networks). A transaction dataset consists of such events as described by [91]

such itemsets that lie entirely within the central neighborhood set. Using these itemsets [102], a Markov random field is learned.

In the last step, the iterative scaling algorithm [91] is used to learn a local MRF for the given central neighborhood set. This process continues overall itemset constraints and continuously updates the model until the model converges. Once the model learning process is over, one can infer the co-occurrence probability by computing the marginal probability over the constructed model. The Junction tree inference algorithm [103] is used to infer co-occurrence probability. The algorithm to induce co-occurrence probability feature for a pair of vertices can be found in [91].

3.2.2 Probabilistic relational model for link prediction (PRM)

Existing works show that node attributes play a significant role to improve the link prediction accuracy. However, no generic framework is available to incorporate node and link attributes and hence, not applicable to all scenarios. To this end, the probabilistic model is a good and concrete solution that provides a systematic approach to incorporate both node and link attributes in the link prediction framework. Pioneering works on PRM include Getoor et al. [47] study on directed networks, Taskar et al. [104] study on undirected networks, Jennifer Neville work on [92] for both networks, etc. [47] published in JMLR is based on Relational Bayesian network (RBN) where relation links are directed and [104] published in NIPS is based on Relational Markov network (RMN) where relational links are undirected.

PRM was originally designed for attribute prediction in relational data, and it later extended to link prediction task [47, 92, 104]. The authors employed the attribute prediction framework to link prediction. This casting can be understood with the following example [27]. Consider the problem of link prediction in a coauthorship network. Non-relational frameworks of link prediction consider only one entity type "person" as node and one relationship; however, relational frameworks (PRMs) include more entity types like article, conference venue, institution, etc. Each entity can have attributes like a person (attributes: name, affiliation institute, status (student, professor)), article (attributes: publication year, type (regular, review)), etc. Several relational links may possible among these entities like advisor-advisee/research scholar relation between two persons, author relationship between person and paper entities, and paper can be related to the conference venue with publish relationship. Moreover, relationships (links) among these entities can also have attributes viz., exists (if there is a link between the two involved entities), or not-exist (no link between the involved entities). This way, the link prediction can be reduced to an attribute prediction framework/model.

During the model training, a single link graph is constructed that incorporates above heterogeneous entities and relationships among them. Model parameters are estimated discriminatively to maximize the probability of the link existence and other parameters with the given graph attribute information. The learned model is then applied using probabilistic inference to predict missing links. More details can be explored in [47, 92, 104].

3.2.3 Hierarchical structure model (HSM) [1]

These models are based on the assumption that the structures of many real networks are hierarchically organized, where nodes are divided into groups, which are further subdivided into subgroups and so forth over multiple scales. Some representative work [1] systematically encodes such structures from network data to build a model that estimates model parameters using statistical methods. These parameters are then used in estimating the link formation probability of unobserved links.

Some studies suggest that many real networks, like biochemical networks (protein interaction networks, metabolic networks, or genetic regulatory networks), Internet domains, etc. are hierarchically structured. In hierarchical networks, vertices are divided into groups, which are further sub-divided into subgroups and so forth over multiple

TABLE 3.2: Probabilistic and maximum likelihood models for link prediction

Model	Network types	Characteristics	References
Hierarchical structure model (HSM)	Hierarchical networks	High accuracy for HSM and low for non-HSM structure	Clauset et al. [1]
Stochastic block model (SBM)	Noisy networks	Good at predicting spurious and missing links	Guimera et al. [2] , Natalie Stanley et al. [105], Toni Valles-Catala et al. [106]
Parametric model	Dynamic networks	Extracts only topological features and performs better than structural methods	Kashima and Abe [107]
Non-parametric model	Dynamic networks	Explicitly clusters links instead of nodes	Sinead A. Williamson [108]
Local probabilistic model	Coauthorship networks	Combines co-occurrence features with topological and semantic features	Wang et al. [91]
Factor graph model	Heterogeneous social networks	Link prediction with aggregate statistics problem	Kuo et al. [109]
Affiliation model	Information and Social networks	soft-block assignment of each node	Jaewon Yang et al. [110]

scales [111]. Clauset et al. [1] proposed a probabilistic model that takes a hierarchical structure of the network into account. The model infers hierarchical information from the network data and further applies it to predict missing links.

The hierarchical structures are represented using a tree (binary), or dendrogram, where, the leaves (i.e., n) represent the number of total vertices in the network and each internal vertex out of $(n-1)$ corresponds to the group of vertices descended from it. Each internal vertex r is associated with a probability p_r, then the existing edge probability p_{xy} between two vertices x and y is given by $p_{xy} = p_r$ where, r is their lowest common ancestor. The hierarchical random graph is then, represented by the dendrogram D^* with the set of probability $\{p_r\}$ as $(D^*, \{p_r\})$. Now the learning task is to find the hierarchical random graph(s) that best estimates the observed real-world network data. Assuming all possible dendrograms to be equally likely, Bayes theorem says that the probability of the dendrogram $(D^*, \{p_r\})$ that best estimates the data is proportional to the posterior probability or likelihood, L from which the model generates the observed network and our goal is to maximize L. The likelihood of a hierarchical random graph $(D^*, \{p_r\})$ is computed using the following equation

$$L(D^*, \{p_r\}) = \prod_{r \in D^*} p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}, \tag{3.44}$$

where L_r and R_r are the left and right subtree rooted at r, and E_r is the number of links in the network whose endpoints have r as their lowest common ancestor in D^*. The above equation assumes the convention $0^0 = 1$. For a given dendrogram D^*, it is easy to compute the probability $\overline{p_r}$ that maximizes $L(D^*, \{p_r\})$ i.e.

$$\overline{p_r} = \frac{E_r}{L_r R_r}. \tag{3.45}$$

This can be understood with the following example illustrated in the Figure 3.4 Now, this model can be used to estimate the missing links of the network as follows. Sample a large number of dendrograms with probability proportional to their likelihood. Then, compute the mean connecting probability $\overline{p_{xy}}$ of each non-existing pair (x, y) by averaging the corresponding probability p_{xy} overall sampled dendrograms. Sort these vertices pairs scores in descending order and selects top-l links to be predicted.

3.2.4 Stochastic block model (SBM) [2]

Hierarchical structures may not represent most networks. A more general approach to represent these networks is block model [112, 113] where vertices are distributed (partitioned) into blocks or communities and the connecting probability between two vertices depends on blocks they belong to. Guimerà et al. [2] presented a novel framework where stochastic block model representation of a network is employed to find missing and spurious links. The authors compute the reliability of the existence of links

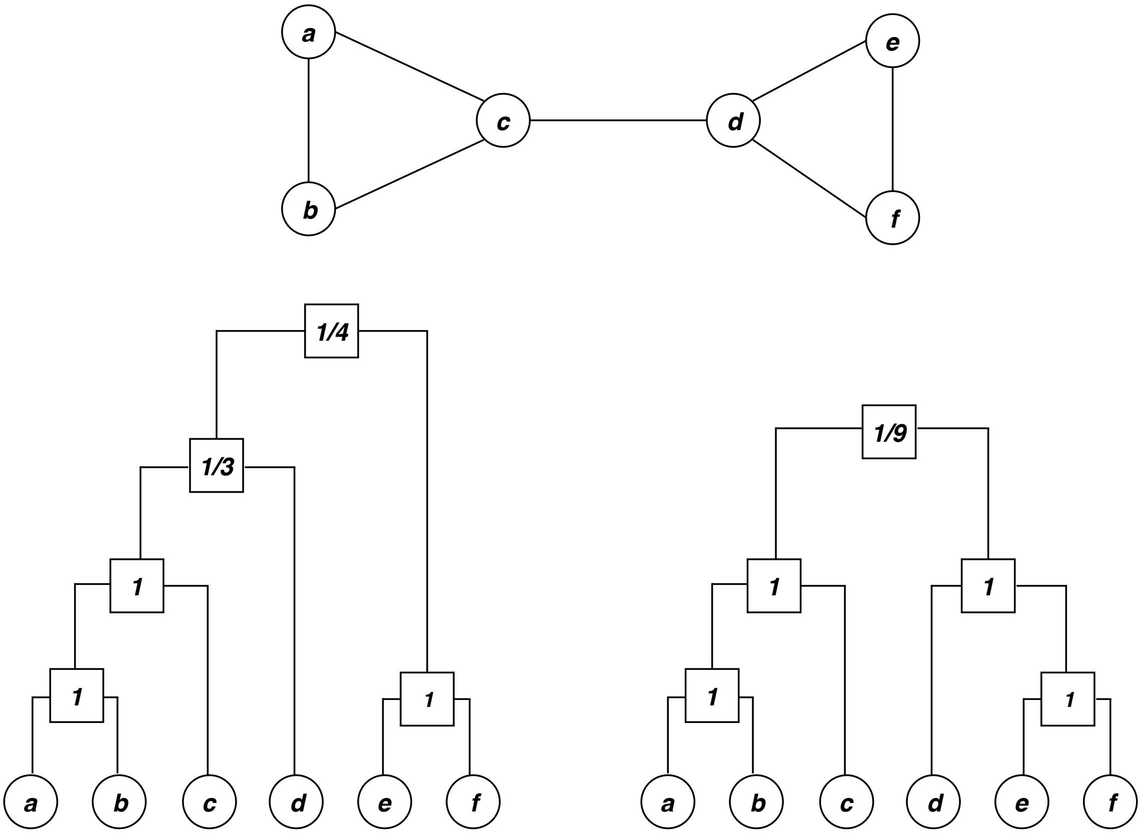

FIGURE 3.4: An illustrating example of HSM for a graph of 6 nodes and its two possible dendrograms as described in the paper [1]. The internal nodes of each dendrogram are labeled as the maximum likelihood probability $\overline{p_r}$, defined by the equation 3.45. The likelihoods of the left and the right dendrograms are $L(D_1) = (1/3)(2/3)^2.(1/4)^2(3/4)^6 = 0.00165$, and $L(D_2) = (1/9)(8/9)^8 = 0.0433$. Thus, the second (i.e., right) dendrogram is most probable as it divides the network in a balanced one at the first level.

given an observed network that is further used to find missing links (non-existing links with higher reliabilities) and spurious links (existing links with lower probabilities).

The link reliability R_{xy} between the two vertices x and y is [2]

$$R_{xy} = p_{BM}(A_{xy} = 1|A^o).$$

i.e. probability that the link truly exists given the observed network A^o, the block model BM.

Generally, complex networks are outcomes of combination of mechanisms, including modularity, role structure, and other factors. In SBM, partitioning vertices of network based on these mechanisms may result in different block models that capture different correlations (patterns) of the network. Assume that no prior knowledge of suitable models, the reliability is expressed as

$$R_{xy} = \frac{1}{Z} \sum_{P \in P^*} \left(\frac{l^o_{\sigma_x \sigma_y} + 1}{r^o_{\sigma_x \sigma_y} + 2}\right) \exp[-H(P)], \tag{3.46}$$

where the sum is over all possible partitions P^* of the network into groups, σ_x and σ_y are vertices x and y groups in partition P respectively. Moreover, $l^o_{\sigma_\alpha \sigma_\beta}$ and $r^o_{\sigma_\alpha \sigma_\beta}$ are the number of links and maximum possible links in the observed network between groups α and β. The function $H(P)$ is

$$H(P) = \sum_{\alpha \leq \beta} [\ln(r_{\alpha\beta}) + \ln \binom{r_{\alpha\beta}}{l^o_{\alpha\beta}}], \tag{3.47}$$

and $Z = \sum_{P \in P^*} \exp[-H(P)]$. Practically, solving equation 3.46, i.e., summing over all possible partitions is too complex even for a small network. However, the Metropolis algorithm [114] can be used to correctly sample the relevant partitions and obtain link reliability estimates.

The authors employed the link reliability concept to find missing links and to identify the spurious link in the networks with the following procedure. (i) Generate the observed network A^o by removing/adding some random links (for finding missing/spurious links) from/to the true network A^t. (ii) Compute the link reliability for non-observed links (i.e. non-existing + missing/spurious links). (iii) Arrange these links with their reliability score in decreasing order and decide the top-l links as desired ones (i.e., missing/spurious links).

Probabilistic and maximum likelihood methods extract useful features and valuable correlation among the data using hierarchical and stochastic block models, which result in significant improvements in prediction results as compared to some similarity-based methods. However, these are quite complex and time-consuming even on small datasets that limit their applicability on large scale real-world network datasets.

3.2.5 Exponential random graph model (ERGM) or P-star model

Exponential random graphs were first first studied by Holland and Leinhardt [115], further explored by [101], and practically used by several works [116–118]. ERGM is an ensemble model where one defines it as consisting of a set of all simple undirected graphs and specifies a probability corresponding to each graph in the ensemble. Properties of the ERGM is computed by averaging over the ensemble [117]. Pan et al. [118] also proposed a similar probabilistic framework (ERGM) to find missing and spurious links in the network. They employed predefined structural Hamiltonian for the score computation. The Hamiltonian is selected based on some organizing principle such that the observed network can have lower Hamiltonian than its randomized one. They defined the structure Hamiltonian by generalizing the 3-order loop to higher-order as

$$H(A) = -\sum_{l=3}^{\infty} \beta_l \ln(Tr(A^l)), \tag{3.48}$$

where A is the adjacency matrix of the network, β_l is temperature parameter. Here, the number of loops of length l starting and ending at the node i is $[A^l]_{ii}$. For undirected network, loops are counted several times when counting occurs for each involved node of the loop, also, for a given node it is counted twice (clockwise and anti-clockwise). Therefore, $Tr(A^l)$ counts approximated to $2l$ times the number of loops of length l that can be taken care of by the parameter β_l [118].

For large value of l, increment in $Tr(A^l)$ reaches to the leading eigen value λ_1 and small world phenomenon of a social network ensures to have l to a lower cut-off l_c.

$$H(A) = -\sum_{l=3}^{l_c} \beta_l \ln(\sum_{i=1}^{n} \lambda_1) \tag{3.49}$$

Note that the above equation is result of diagonalization of the adjacency matrix A^l as follows

$$(Tr(A^l) = Tr(\cup^T \Lambda^l \cup)$$

$$= Tr(\Lambda^l \cup^T \cup) = Tr(\Lambda^l) = \sum_{i=1}^{n} \lambda_i^l$$

Once, the structural Hamiltonian is defined to capture different parameters (higher order loops here), the probability of the appearance of the observed network $A^O = A - A^P$ in an ensemble $\mathscr{M}$ is

$$p(A^O) = \frac{1}{Z} \exp[-H(A^O)], \tag{3.50}$$

where, $Z = \sum_{P \in \mathscr{M}} \exp[-H(A^{'})]$ is the partition function. The parameters β_l are chosen to maximize the probability expressed in the above equation.

Now, the score of non-observed links can be computed by the conditional probability of the appearance of link (x, y)

$$S(x,y) = \frac{1}{Z_{xy}} \exp[-H(\tilde{A}(x,y))], \tag{3.51}$$

where $\tilde{A}(x,y)$ is the observed network by adding the link (x,y), and Z_{xy} is a normalization factor defined as follows [118]

$$Z_{xy} = exp[-H[\tilde{A}(x,y)]] + \exp[-H(A^O)].$$

Here, the prediction is based on the assumption that there is no significant change in the topological structure after adding the link (x,y) to the observed network and the parameter β_l for $\tilde{A}(x,y)$ is almost similar to that of the observed network A^O.

3.3 Dimension reduction frameworks for link prediction

The curse of dimensionality is a well-known problem in machine learning. Some researchers [119, 120] employ dimension reduction techniques to tackle the above problem and apply it in the link prediction scenario. Recently, many authors are working on network embedding and matrix decomposition techniques, which are also considered as dimension reduction techniques.

3.3.1 Embedding-based link prediction

The network embedding is considered as a dimensionality reduction technique in which higher D dimensional nodes (vertices) in the graphs are mapped to a lower d ($d << D$) dimensional representation (embedding) space by preserving the node neighborhood structures. In other words, find the embedding of nodes to a lower d-dimensions such that similar nodes (in the original network) have similar embedding (in the representation space). Figure 3.5 shows the structure of Zachary Karate club social network (left) and the representation of nodes in the embedding space using DeepWalk [4] (right). The nodes are colored based on the membership of their communities.

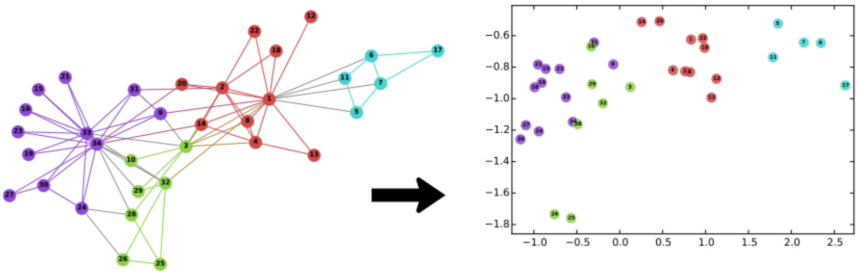

FIGURE 3.5: The Karate club network (left) and its representation in the embedding space with the DeepWalk [4] algorithm.

The main component of the network embedding is the encoding function or encoder f_{en} that map each node to the embedding space.

$$f_{en}(x) = z_x, \tag{3.52}$$

where z_x is the d-dimensional embedding of the node x. The embedding matrix is $Z \in \mathbb{R}^{d \times |V|}$, each column of which represents an embedding vector of a node. Now, a similarity function is $S(x,y)$ is defined that specifies how to model the vector (embedding) space

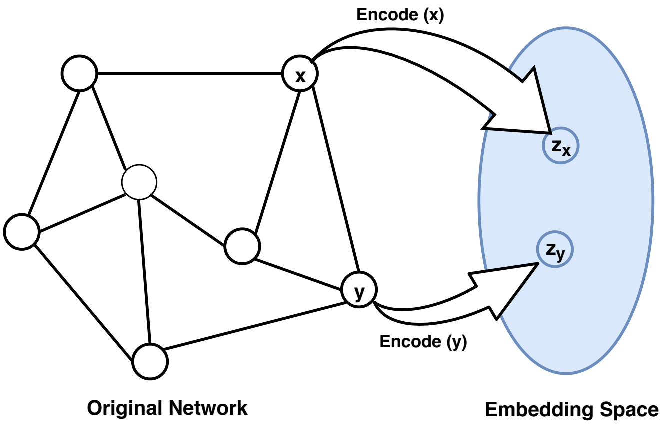

FIGURE 3.6: Embedding of nodes x and y to the embedding space

relationships equivalent to the relationships in the original network, i.e.,

$$S(x,y) \approx z_x^T z_y. \tag{3.53}$$

Here, $S(x,y)$ is the function that reconstructs pairwise similarity values from the generated embedding. The term $S(x,y)$ is the one that differ according to the function used in different factorization-based embedding approaches. For example, graph factorization [121] directly employ adjacency matrix A i.e. $(S(x,y) \triangleq A_{(x,y)})$ to capture first order proximity, GraRep [122] selects $(S(x,y) \triangleq A^2_{(x,y)})$ and HOPE [123] uses other similarity measures(e.g. Jaccard neighborhood overlap). Most embedding methods realize the reconstruction objective by minimizing the loss function, L

$$L = \sum_{(x,y)\in\{V\times V\}} l(z_x^T z_y, S(x,y)). \tag{3.54}$$

Once the equation 3.54 is converged (i.e. trained), one can use the trained encoder to generate nodes embedding, which can further be employed to infer missing link and other downstream machine learning tasks.

Recently, some network embedding techniques [4, 124–127] have been proposed and applied successfully in link prediction problem. The Laplacian eigenmaps [124], Logically linear embedding (LLE) [127], and Isomap [128, 129] are examples based on the simple notion of embedding. such embedding techniques are having quite complex in nature and face scalability issues. To tackle the scalability issue, graph embedding techniques have leveraged the sparsity of real-world networks. For example, DeepWalk [4] extracts local information of truncated random walk and embeds the nodes in representation space by considering the walk as a sentence in the language model [130, 131]. It preserves higher order proximity by maximizing the probability of co-occurrence of random walk of length $2k+1$ (previous and next k nodes centered at a given node). Node2vec [125] also uses a random walk to preserves higher order proximity but it is biased which is a trade-off between the breadth-first search (BFS) and depth-first search (DFS). The experimental results show that the Node2vec performs better than the Deepwalk. In next step, Trouillon et al. [132] introduced complex embedding in which simple matrix and tensor factorization have been used for link prediction that uses a vector with complex values. Such composition of complex embedding includes all possible binary relations especially symmetric and anti-symmetric relations. Recently, some more studies have been published in link prediction using embedding, for example, Cao et al. subgraph embedding [133], Li et al. deep dynamic network embedding [134], Kazemi et al. [126], etc. some seminal works in network embedding are listed in the Table 3.3.

TABLE 3.3: Deep learning models for embedding based link prediction

	Model	Proximity preserved	Embedding type	Scalability	Learning	Reference
With random walk	DeepWalk	Higher order	Shallow	Yes	Unsupervised	[4]
	Node2vec	Higher order	Shallow	Yes	Semi-supervised	[125]
	HARP	Higher order	Shallow	Yes	Supervised	[135]
	Walklets	Higher order	Shallow	Yes	Unsupervised	[136]
Without random walk	LINE	First and second order	Shallow	Yes	Supervised	[137]
	SDNE	First and second order	Deep	No	Semi-supervised	[138]
	DNGR	Higher order	Deep	Yes	Unsupervised	[139]
	GCN	Higher order	Deep	Yes	Semi-supervised	[140]
	VGAE	Higher order	Deep	No	Unsupervised	[141]
	SEAL	First and second order	Deep	Yes	Supervised	[142]
	ARGA	Higher order	Deep	No	Unsupervised	[143]

3.3.2 Factorization-based frameworks for link prediction

From last decade, matrix factorization has been used in lots of papers based on link prediction [144–151] and recommendation systems [48]. Typically, the latent features are extracted and using these features, each vertex is represented in latent space, and such representations are used in a supervised or unsupervised framework for link prediction. To further improve the prediction results, some additional node/link or other attribute information can be used. In most of the works, non-negative matrix factorization (NMF) has been used. Some authors also applied the singular value decomposition (SVD) technique [152].

Let the input data matrix is represented by $X = (x_1, x_2, ..., x_n)$ that contains n data vectors as columns. Now, factorization of this matrix can be expressed as

$$X \approx FG^T, \tag{3.55}$$

where $X \in \mathbb{R}^{p \times n}$, $F \in \mathbb{R}^{p \times k}$, and $G \in \mathbb{R}^{n \times k}$. Here, F contains the bases of the latent space and is called the basis matrix. G contains combination of coefficients of the bases for reconstructing the matrix X, and is called the coefficient matrix. k is the dimension of latent space $(k < n)$. Several well-known matrix factorization techniques are expressed based on some constraints on either of the three matrices, for example, [153],

SVD:

$$X_\pm \approx F_\pm G_\pm^T. \tag{3.56}$$

NMF:

$$X_+ \approx F_+ G_+^T. \tag{3.57}$$

Semi-NMF:

$$X_\pm \approx F_\pm G_+^T. \tag{3.58}$$

Convex-NMF:

$$X_\pm \approx X_\pm W_+ G_\pm^T. \tag{3.59}$$

In the above four equations, $Z_{\pm}$ represents the nature of the entries in the matrix Z, i.e. both positive and negative entries allowed in the matrix Z. In the last equation, $F = XW$ represents the convex combinations of the columns of F. Generally, such a factorization problem can be modeled as the following Frobenius norm optimization problem

$$\begin{aligned} \min_{f,g} \quad & \|X - FG^T\|_{fro}^2 \\ \text{subject to} \quad & F \geq 0, G \geq 0. \end{aligned} \tag{3.60}$$

Here, $\|Z\|_{fro}^2$ is the frobenius norm of Z and the constraints represent NMF factorization. However, any of the above four constraints can be used depending on the requirement of the problem underlying.

After solving the above optimization problem, the similarity between a non-existing pair (x,y) can be computed by the similarity of the x^{th} and y^{th} row vectors in the coefficient matrix G.

Acar et al. [144] expressed temporal link prediction as a matrix completion problem and solve it through the matrix and tensor factorization. They proposed a weighted method to collapsed the temporal data in a single matrix and factorize it using CANDECOMP/PARAFAC (CP) [154, 155] tensor decomposition method. Ma et al. [145] also applied matrix factorization to temporal networks where features of each network are extracted using graph communicability and then collapsed into a single feature matrix using weighted collapsing tensor (WCT) [146]. They showed the equivalence between eigen decomposition of Katz matrix and NMF of the communicability matrix that serves as the foundation of their framework. Further, a notable work by Menon et al. [147] is proposed for structural link prediction. Here, the problem is modeled as matrix completion problem [156], and matrix factorization are used to solve it. They introduced a supervised matrix decomposition framework that learns latent (unobserved) structural features of the graph and incorporates it with additional node/link explicit feature information to make a better prediction. Additionally, they allowed the factorization model to solve class imbalance problem [24] by optimizing ranking loss. Chen et al. [148] proposed somehow similar to work [147], where the authors extracted topological matrix and attribute matrix and factorized these matrices using NMF. The final score matrix is obtained by integrating these two matrices in the latent space. Recently some more works [149–151] have been published in this area.

3.4 Other approaches

3.4.1 Learning-based frameworks for link prediction

Earlier described approaches (e.g., similarity and probabilistic methods) deal with the computing a score of each non-observed link either by a similarity or a probabilistic function. However, the link prediction problem can also be modeled as a learning-based model to exploit graph topological features and attribute information. The problem is cast as a supervised classification model where a point (i.e., training data) corresponds to a vertex-pair in the network, and the label of the point represents the presence or absence of an edge (link) between the pair. In other words, consider a vertex-pair (x,y) in the graph $G(V,E)$ and the label of the corresponding data point in the classification model is $l_{(x,y)}$. Then,

$$l_{(x,y)} = \begin{cases} +1 & \textit{if } (x,y) \in E, \\ -1 & \textit{if } (x,y) \notin E. \end{cases} \tag{3.61}$$

This is typically a binary classification task where several classifiers (e.g., decision tree, naive Bayes, support vector machine, etc.) can be employed to predict the label of unknown data points (corresponding to missing links in the network).

One of the major challenges of this model (i.e., machine learning) is the selection of appropriate feature set [27]. Majority of the existing research works [17, 63, 107] extract feature sets from the network topology (i.e., topological information of the network).

These features are generic and domain-independent that are applicable to any network. Such features are typical, neighborhood, and path-based features. Some other works [63, 157] concentrate on extracting node and edge features that play a crucial role to improve the performance of link prediction. Hasan et al. [63] employed vertex attribute viz., the degree of overlap among research keywords incorporated with other features in the coauthorship network, and showed that the author-pairs having higher values of these features are top rankers in the list. The cost of extraction of such features is cheap and easy, while the main disadvantage is the domain-specific nature of them.

3.4.2 Information theory-based link prediction

Several complex networks have utilized the concept of information theory to compute their complexity on different scales [158, 159]. They defined several correlation measures and modeled some networks (e.g., star, tree, lattice, ER graph, etc.). They also showed that the real networks spanned noise entropy space. Bauer et al. [160] used the maximum entropy principle to assign a statistical weight to any graph and introduced random graph construction with arbitrary degree distribution.

Tan et al. [161] posed the link prediction problem in the framework of information theory. They mainly focus on local assortativity to capture local structural properties of the network and showed that mutual information (MI) method performs well on both low and highly correlated networks. Motivated by [161], Zhu, B. and Xia [162] added more local features (i.e., links information of neighbors of the seed nodes as well as their common neighbors) in their framework and called it as neighbor set information (NSI) index. Thus, they showed that the different features could be combined in an information-theoretic model to improve the link prediction accuracy.

Xu et al. [163] considered path entropy as a similarity metric for the link prediction problem. The authors assumed that there is no correlation among the degrees of the nodes in the network. Consider the following notations based on the paper [163]: L_{xy}^{0} shows no link exists between two vertices x and y, and the corresponding existence is represented by L^1_{xy}. Probability of existence of a link between the above two vertices is given as

$$P(L^1_{xy}) = 1 - P(L^0_{xy}) = 1 - \frac{C^{k_y}_{M-k_x}}{C^{k_y}_{M}}, \tag{3.62}$$

where $C^{k_y}_{M}$ represents the number of candidate link sets for the vertex y with all links incident with y and $C^{k_y}_{M-k_x}$ denotes the number of candidate link sets for the vertex y with all links incident with y but none of them is incident with x.

$$I(L^1_{xy}) = -\log P(L^1_{xy}) = -\log(1 - \frac{C^{k_y}_{M-k_x}}{C^{k_y}_{M}}) \tag{3.63}$$

They show that the likelihood of occurrence of a path having no loops equates to multiplication of the occurrence probabilities of the links involved in that path. i.e., given a simple path $D = v_0, v_1, v_2, v_\gamma$ of length γ, the co occurrence probability of path D is evaluated to

$$P(D) \approx \prod_{i=0}^{\gamma-1} P(L^1_{v_i v_{i+1}}) \tag{3.64}$$

and, the sum of links entropies involved in a path equals to the entropy of the path.

$$I(D) \approx \sum_{i=0}^{\gamma-1} I(L^1_{v_i, v_i+1}). \tag{3.65}$$

Further, they calculated similarity based on entropy of the path which is the negative of conditional entropy

$$S^{PE}_{xy} = -I(L^1_{xy} | \cup_{i=2}^{maxlen} D^i_{xy}), \tag{3.66}$$

where D^i_{xy} represents the set consisting of all simple paths of length i between the two vertices and maxlen is the maximum length of simple path of the network. Outcome results on several networks demonstrate that the similarity index based on path entropy performs better than other indices in terms of prediction accuracy and precision. Xu et al. [164] extend the previous work [163] to the weighted network by considering the weight of the paths. Recently, some more efforts have been applied in this direction based on different features of the networks like influential nodes [165], combining node attributes with structural similarity [166], local likelihood [167], and maximal entropy random walk [168].

3.4.3 Clustering-based link prediction

This paragraph gives an overview of the clustering-based link prediction. Huang [5] presented a paper on graph topology-based link prediction where a generalized clustering coefficient is used as a predictive parameter. The author introduces a cycle formation model that shows the relationship between link occurrence probability and its ability to form different length cycles. This model suggests that the occurrence probability of a particular link depends on the number of different lengths cycles formed by adding this link. The model is based on the assumption of the stationary property of the degree of clustering of the network [169]. This model captures longer cycles by extending the higher-order clustering coefficients [170] and defines the generalized clustering coefficient $C(k)$ as

$$C(k) = \frac{\textit{number of k-length cycles}}{\textit{number of k-length paths}}, \tag{3.67}$$

where k is the degree of the cycle formation model.

The author treats the link occurrence probability as governed by t link generation mechanisms $g(1)$, $g(2)$,...,$g(k)$ of cycle formation model, each described by a single parameter c_1, c_2,..., c_k. The above mentioned link generation mechanism can be understood with the help of the Figure 3.7. Consider a cycle formation model $(CF(k))$ of degree $(k = 3)$. The Seed link (x,y), here, can be generated by the following three mechanisms; the random link occurrence $g(1)$, length-2 cycle generation $g(2)$ i.e. $(x-a-y$ and $x-c-y)$, and length-4 cycle generation $g(3)$ i.e. $(x-b-d-y)$. The main issue is to combine several generation mechanisms to compute total link

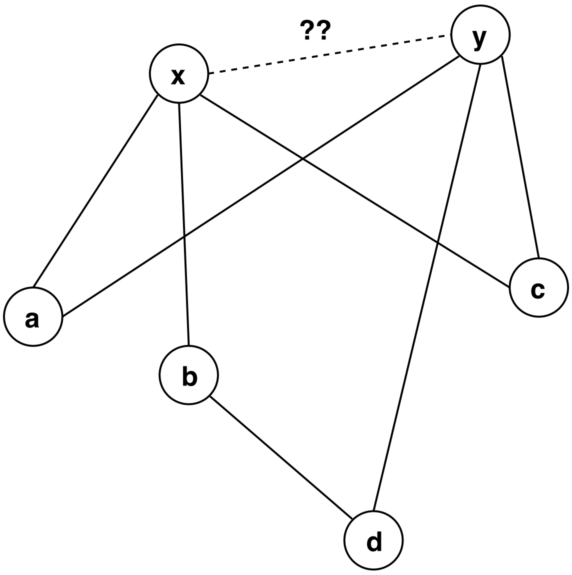

FIGURE 3.7: An example illustrating the cycle formation link probability model [5], where the the probability of the missing link $(x-y)$ is generated by the following three mechanisms; random link occurrence $g(1)$, length-2 cycle generation $g(2)$ i.e. $(x-a-y, x-c-y)$, and length-4 cycle generation $g(3)$ i.e. $(x-b-d-y)$.

occurrence probability. The author [5] posits a method to combine both path and cycle (of different lengths) generation mechanism in the framework. The expected general clustering coefficient of degree k for this model can be estimated as [5]

$$\begin{aligned} E[C(k)] &= f(c_1, c_2, ..., c_k) \\ &= \sum_i |G_i| p(G_i) p((e_{l,k+l}) \in E | G_i), \end{aligned} \tag{3.68}$$

where $|G_i|$ is the number of subgraphs possible corresponding to the graph pattern G_i, listed in Table 1 of the paper [5], $p(G_i)$ is the probability of occurrence of one of such graphs G_i, and $p((e_{l,k+l})$ is the probability of edge $e_{l,l+1}$ to occur given the pattern G_i. Finally, given the coefficients, the probability of existence of link is

$$p_{x,y}(c_1, ..., c_k) = \frac{c_1 \prod_{i=2}^{k} c_i^{|path^i_{x,y}|}}{c_1 \prod_{i=2}^{k} c_i^{|path^i_{x,y}|} + (1-c_1) \prod_{i=2}^{k} (1-c_i)^{|path^i_{x,y}|}}. \tag{3.69}$$

Liu et al. [171] proposed degree related clustering coefficient to quantify the clustering ability of nodes. They applied the same to paths of shorter lengths and introduced a new index Degree related Clustering ability Path (DCP). They performed the degree of robustness (DR) test for their index and showed that missing links have a small effect on the index. Recently Wu et al. [35] extracted triangle structure information in the form of node clustering coefficient of common neighbors. Their experiments on several real datasets show comparable results to the CAR index in [68]. The same concept of the clustering coefficient also introduced in the work presented by Wu et al. [71]. Authors introduce both node and link clustering information in their work. Their experiments on different network datasets showed better performance results against existing methods, especially on middle and large network datasets. Kumar et al. [172] explored the concept of node clustering coefficient to the next level (level-2) that captures more clustering information of a network. Meanwhile, Benson et al. [173] studied simplicial closure events to capture higher-order structures in several temporal networks. The simplicial closure events are the process of closure of timestamped simplices (simplicial complexes[3] are set of nodes with different sizes) available in a dataset. These structures are common in several real-time complex systems, for example, communication in a group, collaboration of authors for a paper, etc. To assess these higher-order structures, the authors study the simplicial closure events on triples of nodes (for simplicity) and suggest that the open triangles or triples of nodes with strong ties are more likely to close in the future.

3.5 Experimental Setup and Results Analysis

There can be two mainly scenarios of link prediction. In the first scenario where static graph is taken into account, a percentage of links are randomly removed fron the network and during evaluation, probabilties of these links are predicted along with other non-existing links. This scenario aims to take the temporal evolution of the graph into account and only links formed after some point in time, t, are removed. The state of the

[3] https://en.wikipedia.org/wiki/Simplicial_complex

graph before t is given to the link predictor and its aim is to predict links formed at a later time. The first setting is applicable when the current knowledge represented by the graph is incomplete and link prediction aims to complete it as well as when the temporal data for the graph is unknown or irrelevant. In the secoend scenario, link prediction finds future links that may appear in near future. This scenario is more challenging because of the reason: during the time, new nodes can be introduced in the network that may contain little or no information as these nodes may not be connected to other nodes in the network initially.

We have used the first scenario during the evaluation of link prediction algorithm. Random-slicing has been used with 10-fold cross-validation. In this setting, we randomy split a given dataset in 10 disjoint subsets with consideration of one of them as test set and remaining as training sets in each iteration.

3.5.1 Datasets

This chapter used 8 network datasets from various fields to study the performance of similarity-based algorithms. Karate[4] [174]: A friendship network of 34 members of a Karate club at a US university. Dolphin[3] [175]: A social network of dolphins living in Doubtful Sound in New Zealand. Macaque[5] [176]: is a biological network of cerebral cortex of Rhesus macaque. Football[3] [177]: American football games network played between Division IA colleges during regular season Fall 2000. Jazz[6] [178]: A collaboration network of 115 jazz musician where a link between two musicians denotes music played by both in a band. C. Elegans[3] [11]: A neural network of C. Elegans compiled by D. Watts and S. Strogatz in which each node refers a neuron and, a link joins two neurons if they are connected by either a synapse or a gap junction. USAir97[7] is an airline network of US where a node represents an airport, and a link shows the

[4] http://www-personal.umich.edu/ mejn/netdata/

[5] https://neurodata.io/project/connectomes/

[6] http://deim.urv.cat/ alexandre.arenas/data/welcome.htm

[7] http://vlado.fmf.uni-lj.si/pub/networks/data/

connectivity between two airports. Netscience[3] [179] is a Coauthorship network of researchers in the network theory domain where a node is denoted by a researcher, and an edge denotes coauthorship of at least one paper between two researchers.

TABLE 3.4: Topological information of real-world network datasets

Datasets	$\|V\|$	$\|E\|$	$\langle D\rangle$	$\langle K\rangle$	$\langle C\rangle$
Karate	34	78	2.337	4.588	0.570
Dolphin	62	159	3.302	5.129	0.258
Macaque	91	1401	1.658	30.791	0.742
Football	115	613	2.486	10.661	0.403
Jazz	198	2742	2.235	27.697	0.620
C. Elegans	297	2148	2.447	14.456	0.308
USAir97	332	2126	2.738	12.807	0.749
Netscience	1589	2742	5.823	3.451	0.878

Table 3.4 shows some basic topological properties of the considered network datasets. $|V|$ and $|E|$ are the total numbers of nodes and links of the networks, respectively. $\langle D\rangle$ represents the average shortest distance, $\langle K\rangle$, the average degree, and $\langle C\rangle$, the average clustering coefficient of the network.

3.5.2 Accuracy

Four accuracy measures, namely Recall@k [180], area under the Precision-Recall curve [181], area under the ROC curve [52, 182], and average precision [180], have been used to evaluate each similarity-based algorithm and some other representative methods. We report these results in the Tables 3.5, 3.6, 3.7, and Table 3.8 for similarity-based approaches and Tables 3.9, 3.10, 3.11, and 3.12 for other representative methods. In the tables of other representative methods, the first method (i.e., HSM) is the maximum likelihood-based method followed by the next three embedding-based methods followed by the three clustering methods. The last method of the table belongs to other category. The best results are highlighted in the table on each dataset. These results are generated with the help of code implemented by Gregorio Alanis-Lobato[8].

[8] https://github.com/galanisl/LinkPrediction

Recall@k In the context of recommendation systems we are most likely interested in recommending top-N items to the user. So it makes more sense to compute recall metric in the first N items instead of all the items. Thus the notion of recall@k where k is a user definable integer that is set by the user to match the top-N recommendations objective.

The recall results for each similarity-based method on all the datasets have been shown in Table 3.5. This measure represents the ability to find all positive/relevant samples by a classifier. We observe that the SPM outperforms against the existing methods on four datasets (Macaque, C. Elegans, Jazz, and USAir97). CAR method best performs on Karate and HDI on dolphins. The global version of LHN (i.e., LHNG) works best on football dataset, and RA is the best performing approach on Netscience. Local similarity methods extract relevant documents more precisely on 3 datasets, and the global methods retrieve more accurate on 5 datasets. Quasi-local approaches and CAR-based indices lie in top-5 ranked algorithms. The quasi-local methods have average good performance compared to the global approaches.

Table 3.9 shows the recall results for other representative methods where SPM outperforms on C. Elegans, Jazz, USAir97, and Netscience. HSM is a good indicator for Dolphin, Node2vec for Macaque, and CCLP2 for Karate and Football networks. On Karate, both the CCLP2 and The SPM show equally good performance.

Area under the precision-recall curve (AUPR) Area under the precision-recall curve (AUPR) is proved to be more informative for imbalanced datasets. The real-world networks are highly imbalanced as the number of positive samples is very less than the negative samples. Table 3.6 shows the AUPR results on eight datasets. Here, We observe that the aupr results resemble the recall@k, i.e., SPM best performs on five datasets as that of recall results and CAR, HDI, LHNG, and RA outperform on karate, dolphin, football and netscience datasets respectively.

AUPR results corresponding to other representative methods are tabulated in the Table 3.10. Here, The Node2vec shows the best performance on all datasets except the

TABLE 3.5: Recall Results

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
CN	0.11363	0.30152	0.08918	0.23800	0.50078	0.40944	0.12500	0.52000
JC	0	0.01908	0.02540	0.31800	0.52007	0.07722	0.08750	0.60322
AA	0.10000	0.28320	0.09945	0.22600	0.52125	0.40444	0.11250	0.67419
RA	0.02500	0.27328	0.09513	0.23400	0.52795	0.45833	0.13125	**0.70709**
PA	0.05000	0.33053	0.05459	0	0.10984	0.32611	0.03125	0.00129
SALTON	0.17500	0.27709	0.09891	0.23600	0.51181	0.39000	0.13750	0.52129
SORENSON	0.15000	0.27633	0.08648	0.25200	0.50039	0.37333	0.13125	0.53870
CAR	**0.20000**	0.27251	0.09243	0.26200	0.51850	0.38333	0.12500	0.54129
CAA	0.15000	0.28015	0.10594	0.33600	0.52362	0.38611	0.13750	0.58000
CRA	0.07500	0.27709	0.11459	0.31800	0.56732	0.43888	0.13125	0.61806
CPA	0.11111	0.29313	0.10108	0.20400	0.48858	0.38555	0.08750	0.33225
HPI	0.10000	0.28167	0.07783	0.23800	0.51259	0.40444	0.10625	0.50451
HDI	0.15000	0.28473	0.09405	0.24600	0.48897	0.38277	**0.18750**	0.52129
NLC	0.05000	0.30763	0.07351	0.08800	0.44803	0.39444	0.05000	0.00516
LNBCN	0.10000	0.00305	0.08972	0.26200	0.38897	0.40555	0.11250	0.58322
LHNL	0.12500	0.27251	0.09297	0.24400	0.49409	0.41222	0.09375	0.52451
CCLP	0.05000	0.28015	0.09675	0.29000	0.52244	0.40555	0.10625	0.60000
KATZ	0.05000	0.34122	0.08162	0.20600	0.44212	0.39722	0.10625	0.43741
RWR	0.10000	0.38855	0.10216	0.24600	0.33937	0.08222	0.06000	0.30925
Shortest Path	0	0.09160	0.02702	0.03200	0.02007	0.02111	0.02000	0.13868
LHNG	0	0	0	**0.36200**	0.10669	0.00388	0.01250	0.05185
ACT	0.02500	0.30076	0.04972	0.03600	0.15748	0.33444	0.02000	0.20740
NACT	0	0	0.00540	0.32800	0.33740	0.00888	0	0.34024
Cos^+	0.02500	0.20610	0.04540	0.30400	0.13464	0.01888	0.02000	0.07037
MF	0.05000	0.19923	0.04324	0.30200	0.15590	0.04111	0.10000	0.41642
SPM	0.10000	**0.51297**	0.**16216**	0.28000	**0.65000**	**0.47111**	0.15000	0.63161
L3	0.05000	0.38549	0.11189	0.20200	0.34409	0.37777	0.07500	0.34645
LP	0.15000	0.38931	0.10594	0.23000	0.36692	0.40000	0.13125	0.37677

Macaque, where SPM performs well. We also observe that the Laplacian eigenmaps (Leig) and Isomap are the worst performers on all datasets.

Area under the receiver operating characteristics curve (AUROC) The AUROC (or AUC) results have been reported in the Table 3.7. Here, we observe that the global approaches perform best on Macaque, C. Elegans, Football, jazz, and dolphin, while the RWR is the best-ranking algorithm on C. Elegans and Dolphin, the SPM is the best ranker on the Macaque and Jazz networks and Cos^+ is best on Football. The table shows the local approaches (i.e., RA and LNBCN) best result on usair97 and Netscience. The bast performance of the RA index on usair97 is that this network is highly heterogeneous with a higher clustering coefficient and absence of a strongly assortative linking pattern.

TABLE 3.6: AUPR Results

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
CN	0.07030	0.28027	0.04234	0.17455	0.51238	0.39009	0.11986	0.58330
JC	0.01397	0.03590	0.01626	0.23470	0.51956	0.04744	0.05666	0.48951
AA	0.05710	0.27442	0.05181	0.16056	0.54088	0.39757	0.08579	0.74908
RA	0.04075	0.26368	0.04786	0.16674	0.56630	0.43537	0.10066	**0.76599**
PA	0.02765	0.34323	0.02085	0.00506	0.06746	0.28537	0.02237	0.00276
SALTON	0.12384	0.26720	0.04554	0.15954	0.52720	0.36980	0.10740	0.58826
SORENSON	0.06751	0.25750	0.04205	0.17329	0.51526	0.36370	0.10011	0.60575
CAR	**0.21733**	0.25305	0.04368	0.18561	0.53572	0.36416	0.10058	0.59813
CAA	0.16841	0.27093	0.04799	0.25907	0.55630	0.35679	0.06648	0.63863
CRA	0.10457	0.26834	0.05257	0.22227	0.60831	0.42311	0.08433	0.66808
CPA	0.09365	0.28103	0.04094	0.13220	0.50689	0.36225	0.05561	0.29581
HPI	0.03955	0.26042	0.04161	0.17890	0.52255	0.38245	0.09205	0.57262
HDI	0.10237	0.27818	0.04596	0.16474	0.50829	0.37266	**0.17651**	0.48951
NLC	0.07886	0.29955	0.03999	0.05210	0.41044	0.35726	0.04364	0.00123
LNBCN	0.07839	0.02712	0.03942	0.17575	0.41752	0.39835	0.05266	0.65339
LHNL	0.07935	0.24733	0.04664	0.17493	0.51164	0.39038	0.09565	0.59308
CCLP	0.05137	0.26904	0.04862	0.21183	0.55269	0.39442	0.07150	0.68109
KATZ	0.07354	0.33081	0.04047	0.16959	0.43602	0.38977	0.08983	0.51171
RWR	0.07874	0.38532	0.06197	0.21580	0.25769	0.09175	0.04901	0.20418
Shortest Path	0.01768	0.06612	0.01401	0.02278	0.02293	0.01120	0.02404	0.09914
LHNG	0.01393	0.02578	0.00781	**0.30867**	0.08835	0.00593	0.02896	0.04370
ACT	0.02262	0.31290	0.01823	0.01571	0.10025	0.29573	0.03442	0.17097
NACT	0.01379	0.02903	0.00725	0.25823	0.22698	0.00594	0.01488	0.19203
Cos^+	0.03392	0.19253	0.02463	0.23921	0.11242	0.02406	0.02476	0.02591
MF	0.06056	0.18374	0.02706	0.22821	0.14706	0.04203	0.04985	0.37889
SPM	0.08345	**0.54004**	**0.08662**	0.20540	**0.68789**	**0.44342**	0.08450	0.67826
L3	0.07943	0.38737	0.04389	0.13276	0.29754	0.35378	0.05865	0.32287
LP	0.06738	0.38600	0.04183	0.17927	0.31623	0.37741	0.07059	0.38762

One more thing to note that the PA index works well on networks that follow rich-club phenomenon but, here we observe that the auroc results (please see Table 3.7) on netscience dataset (having a member of rich-club phenomenon), the PA and its CAR version (i.e., CPA) are having lowest values compared to all other methods. The reason is that this network is disconnected (consists of many connected components), and hence many nodes are isolated and lower degree values. The Quasi-local method viz., the path of length 3 (i.e., $L3$), is the best performer on the karate network. On these datasets, only quasi-local and global approaches lie in top-5. The quasi-local methods lie among top-5 on almost all datasets considered here.

The auroc results of other representative methods are shown in the Table 3.11, where SPM is the best performer on Macaque, C. Elegans, and Jazz networks. HSM performs best on Football and Dolphin networks, CCLP is the best method on USAir97 and Netscience networks. On Karate, Isomap is the best performing similarity index.

TABLE 3.7: AUROC Results

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
CN	0.66139	0.78749	0.87663	0.86928	0.95854	0.96328	0.81040	0.99832
JC	0.60817	0.40191	0.81608	0.85834	0.96612	0.93311	0.77303	0.99945
AA	0.65683	0.78724	0.88189	0.84932	0.96488	0.97402	0.74429	0.99932
RA	0.72318	0.79259	0.88736	0.85685	0.97466	**0.97728**	0.78629	0.99953
PA	0.67080	0.91731	0.76145	0.25727	0.76620	0.91754	0.66843	0.74877
SALTON	0.71231	0.78275	0.86753	0.85063	0.96092	0.96565	0.76276	0.99933
SORENSON	0.66563	0.77183	0.86141	0.85060	0.95879	0.96409	0.75858	0.99919
CAR	0.50134	0.78093	0.84277	0.84644	0.96118	0.95233	0.68292	0.95230
CAA	0.45998	0.79080	0.83838	0.84661	0.96122	0.95184	0.66276	0.94446
CRA	0.51796	0.78359	0.84652	0.83926	0.96944	0.96218	0.66596	0.94907
CPA	0.62594	0.81236	0.77510	0.67895	0.94639	0.91978	0.57540	0.76972
HPI	0.65191	0.78251	0.86798	0.88764	0.96274	0.96466	0.75740	0.99957
HDI	0.74994	0.78996	0.87213	0.86098	0.96041	0.96735	0.80747	0.99887
NLC	0.70911	0.84349	0.86321	0.80929	0.95393	0.90702	0.73127	0.59213
LNBCN	0.41790	0.21923	0.75164	0.78726	0.87271	0.96541	0.60227	**0.99964**
LHNL	0.70088	0.77854	0.86965	0.85957	0.95841	0.96452	0.75054	0.99937
CCLP	0.67934	0.79209	0.88046	0.86181	0.96510	0.96933	0.80289	0.99849
KATZ	0.73788	0.86200	0.86363	0.86480	0.94763	0.96276	0.80212	0.99934
RWR	0.84177	0.92849	**0.90660**	0.90110	0.95999	0.97113	**0.88020**	0.99347
Shortest Path	0.61283	0.61649	0.79575	0.75712	0.68467	0.83586	0.85132	0.95818
LHNG	0.64006	0.13196	0.72447	0.89802	0.90150	0.73105	0.77995	0.98541
ACT	0.51413	0.86292	0.74735	0.56118	0.80132	0.92371	0.81672	0.94354
NACT	0.67774	0.23142	0.65767	0.90085	0.91021	0.69823	0.78073	0.94845
Cos^+	0.80465	0.70811	0.86812	**0.90329**	0.91351	0.91748	0.82919	0.95781
MF	0.75465	0.71741	0.87700	0.89923	0.92916	0.95041	0.84830	0.95504
SPM	0.74565	**0.95551**	0.90499	0.84371	**0.97807**	0.95203	0.75200	0.99148
L3	**0.84751**	0.91431	0.84857	0.84796	0.91036	0.93500	0.77610	0.97264
LP	0.76661	0.91639	0.84942	0.87907	0.91572	0.94884	0.78649	0.99915

Average precision Table 3.8 shows the average precision results of similarity-based methods on eight datasets. The global approaches here, also are the best performer on all datasets except Karate and usair97, and dolphin where the quasi-local index ($L3$), and local indices RA and HDI respectively are the best. Here, SPM performs overall best on Macaque, C. Elegans, and jazz networks. The Resource allocation and HDI methods are top rankers on usair97 and dolphin networks, respectively.

Table 3.12 represents the average precision results of the other representative methods. From the table, it is observed that the Node2vec shows the highest average precision values against all networks except Macaque and Dolphin, where SPM and CCLP respectively show the best results.

TABLE 3.8: Average Precision Results

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
CN	0.01821	0.11240	0.01319	0.03005	0.06329	0.01728	0.02566	0.00113
JC	0.01164	0.03315	0.00988	0.03158	0.06432	0.01179	0.02240	0.00112
AA	0.01585	0.11140	0.01381	0.02882	0.06464	0.01764	0.02238	0.00117
RA	0.01775	0.11130	0.01380	0.02942	0.06649	**0.01850**	0.02538	0.00118
PA	0.01457	0.13441	0.01004	0.00373	0.03275	0.01556	0.01523	0.00032
SALTON	0.01938	0.11038	0.01321	0.02895	0.06384	0.01709	0.02419	0.00112
SORENSON	0.01658	0.10809	0.01288	0.02963	0.06347	0.01705	0.02333	0.00112
CAR	0.01882	0.10872	0.01277	0.02994	0.06429	0.01688	0.02151	0.00104
CAA	0.01644	0.11117	0.01291	0.03153	0.06465	0.01692	0.01889	0.00104
CRA	0.01556	0.11031	0.01335	0.03064	0.06685	0.01782	0.01967	0.00104
CPA	0.01932	0.11484	0.01164	0.02437	0.06280	0.01653	0.01514	0.00064
HPI	0.01461	0.10972	0.01294	0.03078	0.06397	0.01714	0.02288	0.00112
HDI	0.02094	0.11184	0.01334	0.02948	0.06339	0.01722	**0.02800**	0.00112
NLC	0.01725	0.12147	0.01303	0.02145	0.06099	0.01615	0.02017	0.00020
LNBCN	0.01141	0.01766	0.01185	0.02665	0.05660	0.01760	0.01864	0.00114
LHNL	0.01830	0.10794	0.01317	0.02979	0.06336	0.01733	0.02323	0.00112
CCLP	0.01529	0.11138	0.01371	0.03056	0.06488	0.01754	0.02424	0.00115
KATZ	0.01915	0.12693	0.01284	0.02957	0.06022	0.01723	0.02383	0.00108
RWR	0.02122	0.13976	0.01436	**0.03425**	0.05601	0.01427	0.01647	**0.00511**
Shortest Path	0.01252	0.06152	0.00914	0.01551	0.02105	0.00731	0.01347	0.00142
LHNG	0.01212	0.02209	0.00687	0.03401	0.04081	0.00520	0.01889	0.00391
ACT	0.01155	0.12567	0.00950	0.01200	0.03723	0.01554	0.01414	0.00354
NACT	0.01244	0.02610	0.00650	0.03337	0.05158	0.00519	0.01091	0.00359
Cos^{+}	0.01698	0.09097	0.01154	0.03260	0.04411	0.00970	0.01290	0.00324
MF	0.01748	0.09138	0.01193	0.03243	0.04785	0.01174	0.01561	0.00166
SPM	0.01960	**0.15628**	**0.01565**	0.03046	**0.06972**	0.01798	0.02276	0.00115
L3	**0.02324**	0.13855	0.01256	0.02746	0.05304	0.01644	0.02174	0.00095
LP	0.01951	0.13865	0.01258	0.03025	0.05424	0.01689	0.02297	0.00103

Concluding remarks The above results of similarity-based methods on several datasets show that the performance of each technique strongly depends on the structural properties of the network. This highlights the importance of analyzing the properties of the network before choosing a particular link prediction technique. As we observe in our results, the quality of the results is related to the average clustering coefficient of the nodes with degree above one. This is reasonable since most link prediction techniques are variations of counting shared neighbors, and the count of shared neighbors increases as the clustering coefficient does. Another variable that seems to play an important role is the average degree. This makes sense since as we know the more neighbors there are of a node, the more information we have to predict new links for it. However, revealing which specific properties play such an important role in link prediction is still an unsolved problem that requires further work.

Parameters settings We conducted 10-fold cross-validation to evaluate each method on four different evaluation metrics described in the earlier subsection. The disadvantage with the global approaches is parameters tuning that needs to be done carefully to obtain good results. The dumping parameter β of the Katz index is set to 0.01, the return probability $(1-c) = 0.3$ in the random walk with restart method. The ϕ of the global version of Leicht-Holme-Newman, i.e., LHNG, is set to 0.5 that equally balances both self and neighborhood similarity terms. The free parameter $\varepsilon = 0.5$ and path up to the length 3 is considered in the local path index.

TABLE 3.9: Recall results for other representative methods

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
HSM	0.07500	0.34885	0.07405	0.24400	0.29606	0.22666	**0.17000**	0.17407
Leig	0.02500	0.07022	0.01405	0.06000	0.12007	0.01777	0.02000	0.08333
Isomap	0	0.01221	0.01081	0.04200	0.10433	0.01444	0.02000	0.19814
Node2vec	0.03898	**0.67234**	0.02167	0.09887	0.09738	0.02741	0.01947	0.08102
CCLP	0.05000	0.28015	0.09675	0.29000	0.52244	0.40555	0.10625	0.60000
CCLP2	**0.10000**	0.40305	0.09675	**0.32600**	0.41850	0.38555	0.12500	0.41419
NLC	0.05000	0.30763	0.07351	0.08800	0.44803	0.39444	0.05000	0.00516
SPM	**0.10000**	0.51297	**0.16216**	0.28000	**0.65000**	**0.47111**	0.15000	**0.63161**

TABLE 3.10: AUPR results for other representative methods

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
HSM	0.06145	0.33552	0.03442	0.18720	0.23787	0.15064	0.12695	0.14478
Leig	0.03166	0.05281	0.00944	0.04602	0.07086	0.01360	0.01759	0.04326
Isomap	0.03072	0.03123	0.01084	0.03013	0.06972	0.01223	0.01948	0.11020
Node2vec	**0.90000**	0.08058	**0.72930**	**0.79032**	**0.91563**	**0.84788**	**0.65000**	**0.85818**
CCLP	0.05137	0.26904	0.04862	0.21183	0.55269	0.39442	0.07150	0.68109
CCLP2	0.11750	0.40541	0.04602	0.27284	0.44192	0.36988	0.08118	0.41091
NLC	0.07886	0.29955	0.03999	0.05210	0.41044	0.35726	0.04364	0.00123
SPM	0.08345	**0.54004**	0.08662	0.20540	0.68789	0.44342	0.08450	0.67826

TABLE 3.11: AUROC results for other representative methods

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
HSM	0.78390	0.91606	0.84022	**0.88921**	0.87704	0.92754	**0.85570**	0.97286
Leig	0.72189	0.50369	0.70811	0.81513	0.81454	0.81281	0.77543	0.91977
Isomap	**0.80300**	0.32430	0.74469	0.79262	0.85134	0.81146	0.81074	0.96592
Node2vec	0.76850	0.63184	0.80230	0.85278	0.87941	0.85538	0.71474	0.89241
CCLP	0.67934	0.79209	0.88046	0.86181	0.96510	**0.96933**	0.80289	**0.99849**
CCLP2	0.74751	0.91921	0.84180	0.87319	0.93687	0.94741	0.77198	0.97351
NLC	0.70911	0.84349	0.86321	0.80929	0.95393	0.90702	0.73127	0.59213
SPM	0.74565	**0.95551**	**0.90499**	0.84371	**0.97807**	0.95203	0.75200	0.99148

TABLE 3.12: Average precision results for other representative methods

Methods	Karate	Macaque	C. Elegans	Football	Jazz	USAir97	Dolphin	Netscience
HSM	0.01995	0.13304	0.01201	0.03063	0.04953	0.01455	0.01862	0.00427
Leig	0.01585	0.05008	0.00751	0.02061	0.03569	0.00776	0.01124	0.00325
Isomap	0.01764	0.02514	0.00816	0.01789	0.03630	0.00747	0.01200	0.00382
Node2vec	**0.04780**	0.08355	**0.02206**	**0.10330**	**0.09875**	**0.02816**	0.02305	**0.08175**
CCLP	0.01529	0.11138	0.01371	0.03056	0.06488	0.01754	**0.02424**	0.00115
CCLP2	0.02145	0.14074	0.01259	0.03241	0.05924	0.01669	0.02390	0.00100
NLC	0.01725	0.12147	0.01303	0.02145	0.06099	0.01615	0.02017	0.00020
SPM	0.01960	**0.15628**	0.01565	0.03046	0.06972	0.01798	0.02276	0.00115

3.5.3 Efficiency

We have performed our experiment on a 64-bit core i7 Intel system having 8 GB internal memory and 3.60 GHz speed without a dedicated graphics card. To reduce the computational time, some optimization strategies can be applied (if possible) for example, the union and intersection of two sets of sizes m and n can be computed in $O(m+n)$ using hash tables. The computational complexity of the addition and the subtraction of two matrices are $O(n^2)$, however, these operations can be performed in $O(nt)$ in sparse networks where, $t << n$. The matrix multiplication of two dense matrices of sizes $m \times n$ and $m \times p$ are done in $O(mnp)$, while it is $O(mtp)$, where $t << n$. The matrix inversion typically takes $O(n^3)$ time for a square matrix of size $n \times n$ however, some improvements are available that reduce the time to $O(n^{2.81})$ or even less. The time complexity of the similarity-based methods have been tabulated in the Table 3.13, in which most complexities are explained in [45]. In the table, computational complexity of each method are shown using big O notation where n, m, and K are the number of nodes, number of links, and average degree of the networks.

TABLE 3.13: The computational Complexity of similarity-based methods and the corresponding references

Method	Time Complexity	Reference
Local Similarity Index		
CN	$O(nK^3)$	[17]
JC	$O(nK^3)$	[62]
AA	$O(nK^3)$	[58]
RA	$O(nK^3)$	[59]
PA	$O(nK^2)$	[57]
Salton	$O(nK^3)$	[65]
Sorenson	$O(nK^3)$	[66]
CAR	$O(nK^4)$	[68]
CAA	$O(nK^4)$	[68]
CRA	$O(nK^4)$	[68]
CPA	$O(nK^3)$	[68]
HPI	$O(nK^3)$	[15]
HDI	$O(nK^3)$	[15]
LNBCN	$O(n.O(f(z)+nK^3))$	[69]
LHNL	$O(nK^3)$	[70]
CCLP	$O(n^2K^2)$	[35]
NLC	$O(nK^3)$	[71]
Global Similarity Index		
Katz	$O(nK^3)$	[55]
RWR	$O(cn^2K)$	[72]
Shortest Path	$O(nmlogn)$	[17]
LHNG	$O(cn^2K)$	[70]
ACT	$O(n^3)$	[81]
NACT	$O(n^3)$	[81]
L+	$O(n^3)$	[77]
MF	$O(n^3)$	[83]
SPM	$O(n^3)$	[183]
Quasi-local similarity Index		
LP	$O(ln^2K)$	[53]
L3	$O(n^3)$	[54]

3.6 Variations of link prediction problem

As earlier mentioned that the techniques listed in this work mainly focus on a simple abstract graph (i.e., a graph with no vertex or edge attribute). The networks considered in this work simple undirected and unweighted. However, some modification needs to be done to apply on weighted and directed networks. In such networks, links are assigned with weights that represent the strengths of these links. Two types of link direction can be possible of a node (i.e., incoming and outgoing). So, a node x can have two types of neighbors (degrees) viz., in-neighbors $\Gamma_i(x)$ and out-neighbors $\Gamma_o(x)$. Based on these modifications, earlier similarity approaches can be redefined as given below.

3.6.1 Link prediction in weighted and directed networks

In a directed network, the common neighbor method based on in-neighbors and out-neighbors are expressed as

$$S_i(x,y) = |\Gamma_i(x) \cap \Gamma_i(y)|, \tag{3.70}$$

and

$$S_o(x,y) = |\Gamma_o(x) \cap \Gamma_o(y)|. \tag{3.71}$$

In weighted directed network, the expression are

$$S_i^{weight}(x,y) = \sum_{z \in (\Gamma_i(x) \cap \Gamma_i(y))} \frac{w(z,x) + w(z,y)}{2}, \tag{3.72}$$

and

$$S_o^{weight}(x,y) = \sum_{z \in (\Gamma_o(x) \cap \Gamma_o(y))} \frac{w(x,z) + w(y,z)}{2}. \tag{3.73}$$

In a similar way, other approaches can be modified for directed and weighted networks. The point to be noted here is that first, define several topological features (e.g., degree, path, clustering coefficient, etc.) in weighted directed networks and apply these features to implement several link prediction algorithms.

Mostly works on link prediction focus mainly on simple undirected networks due to simplicity. The cost for this simplicity is that it fails to extract rich information available in most real-world networks, which are not undirected in general. Some notable works on directed weighted networks are nicely presented in [184–194]. Lichtenwalter et al. [184] work, published in SIGKDD, extracts 12 topological features on a large directed weighed network of over 5 million nodes and performs ensembles of classification algorithms (C4.5, J48, Naive Bayes). The training over such a big network (millions of edges) is problematic; to mitigate this issue, the authors defined edge features of vertices of 2 and 4 hops only. They perform a quasi-local training to obtain the final model, and their results are outperforming compared to the state-of-the-art. Further, Bütün et al. [189] proposed a new topological similarity metric published in ASONAM that takes into account temporal and weighted information in directed networks which are useful for the improvement of the accuracy. They extract all possible triad pattern features and incorporate them with the weighted version of baseline topological similarity metrics (CN, JC, AA, RA, and PA). They employed a supervised learning framework using these metrics as features and predict missing links. Recently, Bütün et al. [191] introduced a supervised learning model for predicting the citation count of scientists (PCCS). They formulate the problem of PCCS as a link prediction problem and predict links with their weights in weighted temporal and directed (citation) networks. Their model incorporated both local and global topological features and claims the excellence of their proposed work.

The link prediction problem is being explored in several other types of networks, including temporal networks, signed social networks, heterogeneous networks, bipartite networks, etc. Some of these variations are studied in this section.

3.6.2 Link prediction in temporal networks

Today‘s scenario shows that the relationships among users in social networks are continuously changing; for example, each time in the Facebook network, some users join, and some others quit. It results in the networks to be highly complex. Here, time is an important parameter to consider for the evolution of networks. In temporal link prediction, time is considered as the third dimension and represented by a third-order tensor A.

$$A(i,j,T) = \begin{cases} 1 & \text{if node i is connected with node j at time T,} \\ 0 & \text{Otherwise} \end{cases} \tag{3.74}$$

Thus, for a given sequence of snapshots of a network at different time interval T_1, T_2, ... T_t, the link prediction finds links that evolves at the next time slot T_{t+1}.

Several efforts have been employed by the researchers in this direction in the last decade. Purnamrita et al. [195] introduced a nonparametric method for temporal network link prediction where the time dimension is partitioned into subsequences of snapshots of the graph. This approach predicts links based on topological features and local neighbors. Dunlavy et al. [196] employ matrix and tensor techniques in a framework where matrix part collapses sequence of snapshots of networks into a single matrix and computes link scores using truncated svd and extended Katz methods. The tensor part computes the scores using heuristics and temporal forecasting. The tensor part captures the temporal patterns effectively in the network, but it costs heavily also. Moreover, Gao et al. [197] proposed a model based on latent matrix factorization that employs content values with the structural information to capture the temporal patterns of links in the networks. Table 3.14 shows some more works in this direction.

TABLE 3.14: Link Prediction in Temporal Networks

Models	Network Types	Characteristics	References
Learning-based models	Coauthorship networks	High computational cost	Vu et al.[198], Pujari et al.[199], Zeng et al.[200], He et al.[201], Bao et al.[202], Madadhain et al.[203], Bringmann et al. [204]
Heuristics-based models	Twitter, Collaboration and Coauothorship networks	Fast convergence and high precision	Catherine et al.[205], Sherkat et al. [206]
Probabilistic model	Nodes-attributed graphs	Characterize the stochastic and dynamic relations. Need prior link distribution so impractical for real networks	Hu et al.[207], Barbieri et al.[208], Gao et al.[197], Ji Liu et al.[209], Hanneke et al. [210]

3.6.3 Link prediction in bipartite networks

Till now, we have reviewed link prediction methods in unipartite networks in which links may present between any pair of vertices. Now, we review the link prediction problem in a specific graph where only two sets of vertices are present, and a link can be possible between a pair of vertices in which one vertex belongs to one set of vertices and the other vertex to another set of vertices. Such types of networks are called bipartite networks. Lots of social networks logically can be considered as bipartite such as Term-Document network [211], Scientists-Papers cooperation network [212], RNA-PI network [213], IMDb network, and many more.

Kunegis et al. [214] study the link prediction problem in bipartite networks and observed that most common neighbor-based approaches (e.g., Common Neighbors, Adamic/Adar, Resource Allocation, etc.) are not applicable to these networks. The reason is that adjacent nodes belong to different clusters and are connected with the path of odd lengths only. Also, common neighbor-based approaches are based on the path of length two. The authors give hyperbolic Sine and Von Neumann kernels of odd order to compute the similarity between vertices. Only the PA method is applicable to these networks in its natural form because it considers the degree of the neighbors. Some researchers [215–217] have implemented common neighbor-based methods (e.g., CN, AA, RA, PA, LCP-CN, etc) in bipartite networks. Xia et al. [215] studied the link prediction problem by exploiting structural holes in bipartite networks. They proposed two implementations of structural holes viz., absent links (consisting of c-type and s-type links), and minimum description length [218, 219].

Recently several methodologies of link prediction in bipartite networks have been addressed. Baltakiene et al. [220] implemented maximum entropy principle, an extension of the recent one [167]. They used probability of Bipartite Configuration Model [221] as the score function. Allali et al. [222] presented the term ”internal link” based on which they proposed a new link prediction algorithm.

3.6.4 Link prediction in heterogeneous networks

Most of the contemporary approaches of link prediction focus on homogeneous networks where the object and the link are of single (same) types such as author collaboration networks. These networks comprise less information, like which two authors have collaborated with a paper that causes less accuracy for the prediction task. In heterogeneous networks, the underlying assumption of a single type of object and links does not hold good. Such networks contain different types of objects as well as links that carry more information compared to homogeneous networks and hence more fruitful to link prediction, also called multi-relational link prediction (MRLP). Examples of such networks are DBLP bibliography [9] and Flickr networks [10]. In the bibliography database, authors, papers, venue, terms are different types of objects/nodes, and relationships are paper-author, author-author, paper-term, paper-venue, and so on.

Sun et al. [223, 224] coined the concept of heterogeneous information network (HIN) and subsequently meta path concept [225], since then it becomes popular among

[9] https://dblp.uni-trier.de

[10] http://www. flickr.com/

researchers. The key idea to multi-relational link prediction (MRLP) is to employ an appropriate weighting scheme to combine different link types. The authors predict the relationship building time between two objects by encoding the target relation and topological features to meta paths in a supervised framework. Moreover, Yang et al. [226] proposed a new topological feature, namely multi-relational influence propagation to capture the correlation between different types of links and further incorporate temporal features to improve link prediction accuracy. Davis et al. [227] proposed a novel probabilistic framework, a weighted extension of Adamic/Adar measure. Their approach is based on the idea that the non-existing node pair forms a partial triad with their common neighbor, and their probabilistic weight is based on such triad census. Then the prediction score is computed for each link type by adding such weights. Meanwhile, Sun et al. [228] a new supervised framework for HIN where meta path-based topological features (i.e., path count, random walk) are extracted, and then logistic regression is applied to build the relationship prediction model that learns the weight associated with these features. Table 3.15 lists some more works on link prediction in heterogeneous networks.

3.7 Link prediction applications

3.7.1 Network reconstruction

Guimerà et al. [2] proposed a framework that applies link prediction for network reconstruction. They reconstruct of the true network is done from the observed network based on missing links (removed one) and the spurious links (added links). Although it is not obvious because no one knows about the amount of missing and spurious links in the networks. For this, the authors describe the reliability of networks based on the reliability of both missing and spurious links by formulating the link prediction problem

TABLE 3.15: Link Prediction in Heterogeneous Networks

Models	Network Types	Characteristics	Reference
Supervised models	Youtube, Gene, Climate	Proposed both unsupervised and supervised approach to link prediction	Davis et al. [227]
	DBLP	Extracts meta path-based topological features and applies logistic regression as prediction model	Y. Sun et al. [229]
	Epinions, Slashdot, Wikivote, Twitter	Define social pattern-based features (social balance and microscopic mechanism), input to the inference model namely (transfer) factor graph models	Y. Dong et al. [230]
Collective LP models	MovieLens, Book-Crossing, Douban	Non-parametric Bayesian model that considers the similarity between tasks when leveraging all the link data together	B. Cao et al. [231]
	Flickr, DBLP	Distance feature extraction usibg both network and node features and for learning Multi-Task Structure Preserving Metric Learning (MTSPML) is used	S. Negi et al. [232]

as a stochastic block model [113]. The reliability of the network A is [2]

$$R(A) = \prod_{A_{xy}=1, x<y} R_{xy} = \prod_{A_{xy}=1, x<y} L(A_{xy} = 1/A^o), \tag{3.75}$$

where R_{xy} is the reliability of the link (x,y) that is defined by the likelihood that the link (x,y) truly exists given the observed network A^o. This equation can be solved by finding out the network A that maximizes the reliability given by 3.75.

The computational cost of the equation is high, so the authors [2] give a greedy algorithm to compute it. The algorithm starts with computing the link reliability of all pairs of vertices. At each step, the algorithm removes the least reliable link and adds the most reliable link (non-existing in the current network). This change in the network is accepted when the reliability of the network increases and rejected otherwise. In case of rejection, the next step selects the least reliable existing link and the highest reliable non-existing links for swapping. The algorithm stops when there are no five consecutive changes (swaps) in the network. The reliability of the network improves from the initial observed network, which is the reconstructed ones. Now, the authors compare the six (6) global properties of both the observed and the reconstructed networks and show that the reconstruction improves the estimates.

3.7.2 Recommender system

The recommender systems [21, 22, 48, 49] (also called information filtering systems) have been widely applied in social media (like Facebook, Twitter) and online shopping websites (e.g., Flipkart, Amazon, etc.). Such systems recommend new friends, followees, and followers on social networking platforms and new products on online shopping portals based on users' previous browsing history (such as interests, preferences, ratings, etc.). Even though collaborative filtering (CF) is a successful recommendation paradigm that applies transaction (Transaction/purchase is essentially an implicit and coarse rating on preferring an item [233]), information to enrich user and item features for recommendation. Although they have been applied in many recommender systems, they are greatly limited by data sparsity problem [234]. The recommender system in bipartite networks can be mapped to link prediction problem as follows [235]. Consider U^* and O be the sets of users (first set of vertices) and objects/items (second set of vertices). Construct the user-item interaction graph $G = (V,E)$ from the available transactions T (purchasing patterns), where $V = U^* \cup O$ and $E = \{(u,o) : u \in U^*, o \in O, u \to o \in T\}$.

Huang et al. [21] and Li et al. [236] proposed approaches, where the recommender system (user-item recommendation) is represented as a bipartite graph, and employed basic link prediction approaches for the items recommendation. Sadilek et al. [237] proposed FLAP

(Friendship + Location Analysis and Prediction) system in which both friendship and location prediction tasks are implemented. They employed users tweets, their locations, and their neighboring information as model features and inferred social ties and location using MRF. More related works can be found in [235, 238, 239].

3.7.3 Network completion problem

In general, the network representation of the real-world problem is incomplete or partially observed or incremental with both missing links and nodes such as wall posts on Facebook, tweets in the Tweeter, etc. The problem arises due to several reasons like security, data aggregation overhead, manual errors, etc. Predicting such nodes and links is the network completion problem in which some notable works like [240] in SIGKDD, [241] in ASONAM, and [156] in EPL. Filling missing entries of the adjacency matrix of a network is link prediction, which can be considered as a subset of network completion problem. Kim et al. [242] cast network completion problem to the Expectation-Maximization (EM) framework and proposed KronEM, an EM model based on Kronecker graphs. They, first, represent the network as Kronecker graph and estimate the model parameters as well as missing links using KronEM algorithm. The estimated network is then considered as the complete network and re-estimate the model, and this process is repeated until the convergence. Further, Pech et al. [156] employed the robust principle component analysis (Robust PCA) [243] [to recover both low rank and sparse components of a data adjacency matrix] in link prediction framework and introduce a novel global prediction method using both the components. They reconstruct the original network using the robust PCA where these components are extracted by minimizing the weighted combination of the nuclear norm and of the l_1 norm [243].

$$\min_{X^*,\mathscr{E}} \|X^*\|_* + \lambda \|\mathscr{E}\|_1 , \tag{3.76}$$

where $\|.\|_*$ is nuclear norm (i.e., sum of singular values) of the matrix and $\|.\|_1$ is the l_1 norm, $\mathscr{E}$ is error or noise matrix (sparse matrix containing spurious links as positive entries and missing links as negative entries) and λ is a positive weighing parameter that balances the contribution of both the components (low rank and sparse components). X^* $[= A^O - \mathscr{E}$, here, A^O is the observed network] is the set of patterns (links that are newly predicted and some links that are eliminated). From which only the newly appeared links are extracted and added to the observed network A^O to recover the original matrix (also known as reconstructed matrix). Once the reconstructed matrix is obtained, link prediction can be performed accordingly.

3.7.4 Spam mail detection

Spreading and receiving irrelevant emails is common in today's world that consumes network bandwidth, memory, etc. Many email service companies are trying to implement several filter mechanism to stop such emails known as spam mails. To implement spam filter mechanism, spam detection is a necessary task. In this context, Huang and Zeng [244] proposed a model to detect spam emails using link prediction. They construct an email graph (directed and weighted) based on the email data, consisting of a sender, recipient, and timestamp of the communication as attributes. Many email communication links between the sender and the receiver are mapped to a weight of the link between them. Then, an anomaly score is computed for each distinct sender-recipient pair using the Adamic/Adar link prediction approach by making it adaptive based on the spreading activation algorithm. Some more work related to this can be found in [245, 246].

3.7.5 Privacy control in social networks

Lots of users share personal posts, audios, videos, and other sensitive information to social networking websites. Trust is an important parameter to evaluate users' relationships on such media, i.e., the strength of a relationship between two users can be determined based on the trust in the form of link weights. Thus, it is important for companies to maintain the privacy of users from anomalous ones. Oufi et al. [247] proposed a framework implementing a capacity-based algorithm that employs Advogato trust metric [248, 249] to compute the level of trust between users. This means that the framework identifies all possible trustworthy users of a seed user, which results in the privacy of that user in the network from anomalous users.

3.7.6 Identifying missing references in a publication

A research article may contain some irrelevant references and miss some relevant ones. Identifying such missing references in a research article is an important task to avoid plagiarism. It becomes more critical for the point of a novice researcher due to a lack of literature survey carried out by him. Kc et al. [250] proposed a machine learning approach to link prediction tackle this problem. They provide a framework for the generation of links between referenced and otherwise interlinked documents. The nodes of the graph represent documents, and the links between them show references available between them. They find new links/references of documents based on this graph using Probability Measure Graph Self-Organizing Map (PM-GraphSOM).

3.7.7 Routing in networks

In complex network theory, link prediction in social networks resembles link quality prediction in wireless sensor network [251]. The routing problem in a network finds the shortest path (optimal) between the sender and the receiver. The strength of signal frequently varies in mobile and ad hoc networks that results in frequent breaks in routes and degrades the performance result. Weiss et al. [252] and Yadav et al. [253] proposed some models to estimate the signal strength-based link availability prediction for optimal routing. Such link information is beneficial to estimate the link breakage time and hence, to repair the existing route or to discover a new route for the packets. This reduces end-to-end routing delay and packet drops, thereby improving the performance. Once broken links are identified earlier (using link breakage time), routing management protocol needs either to repair the broken link or to find an alternate route. Several works state that link prediction may play a crucial role in this scenario that results in low latency in packet delivery to the receiver and hence improves reliability. Hu and Hou [254] presented link prediction-based traffic prediction for the best routing of packets in a wireless network. Some more works in this area can be found in [255, 256]. Recently, Zhao et al. [251] proposed a neighborhood-based NMF model to estimate the link quality in the wireless sensor network. They extend the link prediction model to the wireless sensor network, where they predict the quality of a link based on NMF associated with structural (neighborhood) information.

3.7.8 Incorporating user's influence in link prediction

Lots of works based on individual influence have been proposed in social network analysis, such as link prediction [257, 258], information diffusion [259–261], influence maximization [44, 262–266], community detection [267, 268], etc. Particularly, the role of individual influence in link prediction provides a new perspective/insight into the problem. Influence maximization (IM) [44] is one of the fundamental problems in social network analysis where the goal is to find a set of users (seed set) that can be further utilized to maximize the expected influence spread (defined as the expected number of influenced users) among others. The influence (social influence here) is propagated through certain channels (i.e., intermediate nodes), that are captured by diffusion models [259]. IM and diffusion models are a cooperative and correlated task as for IM, several Diffusion models are used in the computing framework. Zhang et al. [269] proposed a new framework of link diffusion to predict more links in the microblogging networks. They find the triadic structure to be the crucial factor that affects the link diffusion process and hence, link prediction. Earlier, Cervantes et al. [270] proposed a supervised learning model to find an influential collaborative researcher in the collaboration network. They employ the model to the whole network and compare its result with those sub-networks generated each time when a distinct vertex is removed from the training set. Finally, results are ranked and examine the collaborative influential of each researcher based on the presence or absence of it in the network. Finding influential users (i.e., the seed set) is useful in many applications like viral marketing, where an influential user can be used to advertise the product to maximize the profit. Other application areas may be disease prevention using vaccinate to the most influential patient.