Link Prediction Based on Minimum Degree-Biased Random Walk with Restart: Postprint

Authors: Li Qiaoli, Han Hua

Date: 2022-05-10T11:22:58Z

Abstract

Link prediction is an important problem in the field of data mining. Similarity methods based on random walks generally assume that the probability of a walking particle transferring to neighboring nodes is equal, ignoring the influence of node degree values on the transfer probability. To address this issue, a link prediction method based on lowest-degree biased restart random walk is proposed. First, a lowest-degree bias function is introduced to redefine the transfer probability of walking particles, then the lowest-degree biased random walk strategy is applied to restart random walk to investigate the influence of the lowest-degree bias strategy on node similarity during the walking process. Link prediction is conducted on nine real-world network datasets, and the results demonstrate that the proposed method achieves favorable prediction accuracy and uncovers more network topological structure information, proving that the algorithm possesses certain advantages in evaluating node similarity.

Full Text

Preamble

Link Prediction Algorithm Based on Lowest-Degree Preference Random Walk with Restart

Li Qiaoli, Han Hua†
(School of Science, Wuhan University of Technology, Wuhan 430070, China)

Abstract: Link prediction is an important problem in data mining. Similarity methods based on random walk typically assume that the probability of a particle transferring to adjacent nodes is equal, ignoring the influence of node

degree on transition probability. To address this issue, this paper proposes a link prediction algorithm based on lowest-degree preference random walk with restart. First, a lowest-degree preference function is introduced to redefine the transition probability of walkers. Then, the lowest-degree preference random walk strategy is applied to random walk with restart to investigate the effect of the lowest-degree preference strategy on node similarity during the walk. Experiments on nine real-world network datasets demonstrate that the proposed method achieves good prediction accuracy and uncovers more network topological structure information, proving that the algorithm has certain advantages in evaluating node similarity.

---

0 Introduction

In recent years, research in network science has flourished, with an increasing number of complex systems becoming subjects of study in complex network theory. Individuals and their interactions in complex systems can be abstracted as complex networks. Common examples include biological networks, social networks, and communication networks. As an important research tool for complex networks, link prediction aims to discover unknown connection relationships in networks using known information. Link prediction research holds significant value across numerous domains. Theoretically, it can help us better understand network evolution mechanisms and dynamical behaviors. Practically, typical applications include user expansion in social networks, fraud source identification in telecommunication networks, and precision marketing in e-commerce platforms.

Many classical link prediction algorithms have been proposed, with similarity-based methods being the most widely applied. Structure-based similarity methods can be broadly categorized into: (a) local information-based methods, (b) path-based similarity methods, and (c) random walk-based methods. Local information-based methods primarily utilize node-local information (such as node degree and number of common neighbors) for link prediction. These methods have low computational complexity but often sacrifice accuracy. Path-based methods tend to leverage path information between nodes (such as the number of paths and information about intermediate nodes) to calculate node similarity. These methods have relatively high computational complexity when involving multi-order and global path information. Random walk-based methods are defined based on particle random walk processes, where a particle starts from an initial node and randomly walks to its neighbors with certain probabilities until the probability distribution reaches a stationary state. These metrics focus only on local neighbor information and achieve a good trade-off between computational complexity and prediction performance, making them widely applied in

recommendation systems, information propagation, and community detection.

This advantage has made random walk a primary approach for solving link prediction problems, yielding many achievements. A typical example is the PageRank algorithm, where random walk methods play a key role. Additionally, Li et al. proposed a maximum entropy random walk algorithm for link prediction, arguing that in real networks, nodes tend to connect not only to low-degree nodes but also to central nodes. However, this algorithm involves calculating node centrality, resulting in relatively high complexity. Another study used the DeepWalk network representation learning algorithm to obtain node vector representations and characterized structural similarity through Euclidean distance, proposing a link prediction algorithm combining network representation learning and random walk. This algorithm considers both network structure and node attribute information but struggles with large-scale networks. Jin et al. proposed a supervised and extended restart random walk method where each node has a corresponding restart probability. Experimental results showed good performance for ranking and link prediction tasks, but the non-universal setting of node restart probabilities limits the algorithm's applicability.

Most existing random walk-based methods define particle transition probabilities using uniform distributions, ignoring the influence of subtle local structures on transition probabilities. In fact, degree-degree correlations in networks show that connections between nodes are not randomly generated, and particles are affected by node degrees during walks. Recent research found that random walkers frequently visit high-degree nodes, leading to lower search efficiency. Inspired by the PageRank algorithm, they proposed a lowest-degree preference random walk search strategy (LPRW), demonstrating significant reduction in search time compared to unbiased random walks. Another study argued that particles exhibit degree bias during walks and proposed the BRWR method, also showing that greater bias toward high-degree nodes reduces prediction accuracy.

Inspired by these methods and the PageRank algorithm, this paper proposes a lowest-degree preference random walk with restart link prediction algorithm. This algorithm employs a mixed walk strategy combining pure random walk with preference for visiting lowest-degree neighbors, applied to link prediction. The method first redefines walker transition probabilities by introducing a lowest-degree preference function, then applies this strategy to random walk with restart to investigate the effect of lowest-degree preference on particle transitions, and finally validates the method's effectiveness through multiple real-world network datasets.

1.1 Problem Description

Given an unweighted, undirected network represented as a binary pair $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. For all nodes in the network, the set of all possible node pairs that could form

connections is denoted by $U = V \times V \setminus E$. The network can be represented by an adjacency matrix $A = (a_{uv})_{N \times N}$, where $a_{uv} = 1$ if nodes u and v are connected, otherwise $a_{uv} = 0$. A prediction algorithm assigns a similarity score value S_{xy} to each pair of unconnected nodes $(x, y) \in U$ in the network. All S_{xy} values are sorted in descending order, with edges ranked higher having greater likelihood of existence.

In practical prediction, similarity score thresholds are typically set based on different evaluation requirements, with edges above the threshold selected as recommendations. Alternatively, the top L predicted edges are selected based on similarity ranking. Predicted edges can be further applied to e-commerce recommendation systems or serve as guidance in biological experiments.

1.2 Link Prediction Methods

For any two nodes u and v in a network, let $\Gamma(u)$ and $\Gamma(v)$ represent their neighbor sets, and $\Gamma(u) \cap \Gamma(v)$ represent their common neighbor set. Let k_u denote the degree of node u. Below are several commonly used similarity indices:

a) **Common Neighbors (CN).** This measures the similarity between nodes u and v by the number of their common neighbors, expressed as:

$$S_{uv}^{\text{CN}} = |\Gamma(u) \cap \Gamma(v)|$$

where $\Gamma(u)$ is the neighbor set of node u, and $|\cdot|$ represents set cardinality.

b) **PA Index.** Based on preferential attachment characteristics, this index assumes nodes tend to connect to high-degree nodes:

$$S_{uv}^{\text{PA}} = k_u \cdot k_v$$

c) **RA Index.** This is a similarity measure based on shared features, where low-degree common neighbors contribute more than high-degree ones. It weights similarity using the inverse of common neighbor degrees:

$$S_{uv}^{\text{RA}} = \sum_{\omega \in \Gamma(u) \cap \Gamma(v)} \frac{1}{k_\omega}$$

d) **HDI Index.** Called the High-Degree Node Disadvantage Index:

$$S_{uv}^{\text{HDI}} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\max\{k_u, k_v\}}$$

e) **Katz Index.** This is essentially a shortest path method that considers all paths between two nodes, applying hierarchical penalties based on path

length:

$$S_{uv}^{\text{Katz}} = \beta A_{uv} + \beta^2 (A^2)_{uv} + \beta^3 (A^3)_{uv} + \cdots = \sum_{l=1}^{\infty} \beta^l (A^l)_{uv}$$

where β is a path weight adjustment parameter, and $(A^l)_{uv}$ represents the number of paths of length l between nodes u and v.

f) **SimRank Index (SimR).** This assumes that two nodes are similar if they are connected to similar nodes, describing the average time for two particles starting from nodes u and v to meet:

$$S_{uv}^{\text{SimR}} = C \cdot \frac{\sum_{\omega \in \Gamma(u)} \sum_{\omega' \in \Gamma(v)} S_{\omega\omega'}^{\text{SimR}}}{k_u \cdot k_v}$$

where $C \in [0, 1]$ is a decay parameter for similarity propagation.

g) **Average Commute Time (ACT).** This similarity index is defined based on random walk, representing the average number of steps for a particle to travel from node u to node v:

$$S_{uv}^{\text{ACT}} = \frac{1}{l_{uu}^+ + l_{vv}^+ - 2l_{uv}^+}$$

where l_{uv}^+ represents the element in the u-th row and v-th column of the network's Laplacian matrix pseudoinverse.

h) **Random Walk with Restart (RWR).** This index extends the PageRank algorithm. A particle performing random walk may return to its starting position with a certain probability at each step. Let the return probability be α, and the network's Markov transition matrix be $P = (p_{uv})_{N \times N}$, where p_{uv} and a_{uv} represent elements in matrices P and adjacency matrix A, respectively. If a particle starts at node u at time $t = 0$, the probability distribution vector of its location at time t is:

$$\pi_u(t + 1) = (1 - \alpha) \cdot \pi_u(t) \cdot P + \alpha \cdot e_u$$

where e_u represents the initial state. The stable solution is $\pi_u = (1 - \alpha) \cdot \pi_u \cdot P + \alpha \cdot e_u$, where π_u is the stationary solution vector and $\pi_u(v)$ is its v-th element. The RWR similarity is then defined as:

$$S_{uv}^{\text{RWR}} = \pi_u(v) + \pi_v(u)$$

---

2 Similarity Method Based on Lowest-Degree Preference Random Walk with Restart

Random walk plays a crucial role in complex network research and has achieved significant results in various domains, including community detection, link prediction, and important node mining. It is generally divided into pure random walk and biased random walk.

Pure random walk refers to a walker starting from any node or source node u and jumping to a neighbor with equal probability. In contrast, biased random walk forcibly seeks the nearest target node in an unknown network. In a biased random walk, the transition probability from the current node to potential new nodes is unequal, and the walker tends to visit or ignore nodes with high topological attribute values, including strength, clustering coefficient, or degree. Therefore, this paper assumes that during random walk, particles adopt a mixed strategy combining pure random walk with preference for visiting lowest-degree neighbors. Based on this mixed strategy, we derive the particle's transition probability matrix. On this foundation, particles walk using random walk with restart to calculate similarity scores for unconnected node pairs in the network, finding the optimal lowest-degree bias adjustment parameter for each network to improve prediction accuracy.

2.1 Lowest-Degree Preference Random Walk with Restart

Definition 1: Lowest-Degree Preference Transition Probability. Consider a particle jumping between adjacent nodes in a network. According to Markov processes, the particle's next state depends only on its current state. In the lowest-degree preference random walk process, at each time step, the walker adopts a mixed strategy of pure random walk and preference for visiting lowest-degree neighbor nodes, using a variable parameter β to adjust the fusion ratio. The transition probability for a walker currently at node u to jump to node v is defined as:

$$w_{uv}^{(1)} = \frac{a_{uv}}{k_u}$$

$$w_{uv}^{(2)} = \begin{cases} \frac{1}{\text{card}(U_v)} & v \in U_v \\ 0 & v \notin U_v \end{cases}$$

$$w_{uv} = (1 - \beta) \cdot w_{uv}^{(1)} + \beta \cdot w_{uv}^{(2)}$$

where $w_{uv}^{(1)}$ represents the transition probability under pure random walk strategy, $w_{uv}^{(2)}$ represents the probability under lowest-degree walk strategy, U_v denotes the set of node v's lowest-degree neighbors, and $\text{card}(U_v)$ is the number of lowest-degree neighbors. Notably, when $\beta = 0$, the lowest-degree preference random walk degenerates to standard random walk, where the stationary probability of the walker staying at node u is proportional to node u's degree, making high-degree nodes more likely to be visited. The lowest-degree preference random walk avoids this by simultaneously adopting the lowest-degree search strategy when $\beta > 0$. Figure 1 illustrates the transition probabilities for lowest-degree preference random walk when $\beta = 1/3$.

Definition 2: Lowest-Degree Preference Random Walk with Restart (LPRWR). This algorithm applies the lowest-degree preference transition probability from Definition 1 to random walk with restart. Let $\pi_{uv}(t)$ be the probability that a particle starting from node u at time 0 stays at node v at time t.

This probability evolution is governed by:

$$\pi_{uv}(t+1) = (1-\alpha)\sum_{l=1}^{N}\pi_{ul}(t)\cdot w_{lv} + \alpha\cdot\pi_{uv}(0)$$

where α is the restart probability and $\pi_{uv}(0)$ represents the v-th element of the initial state vector. Using matrix notation, the one-step transition probability can be expressed as:

$$\pi_u(t+1) = (1-\alpha)\cdot\pi_u(t)\cdot W + \alpha\cdot\pi_u(0)$$

where W is the lowest-degree preference transition matrix. According to the Chapman-Kolmogorov equation, the m-step transition probability is:

$$\pi_u(t+m) = (1-\alpha)^m\cdot\pi_u(t)\cdot W^m + \alpha\sum_{n=0}^{m-1}(1-\alpha)^n\cdot\pi_u(0)\cdot W^n$$

As $t\to\infty$, by the stationary state property of Markov chains, the probability distribution converges to a limiting distribution satisfying $\Pi = (1-\alpha)\cdot\Pi\cdot W + \alpha\cdot\Pi(0)$. This can be rewritten as:

$$\Pi = \alpha\cdot\pi_u(0)\cdot[I-(1-\alpha)W]^{-1}$$

where I is the identity matrix. The element π_{uv} represents the probability that a particle starting from node u ultimately reaches node v. Therefore, the LPRWR similarity is defined as:

$$S_{uv}^{\text{LPRWR}} = \pi_{uv} + \pi_{vu}$$

Algorithm 1: LPRWR Algorithm

Input: Network adjacency matrix A, lowest-degree bias adjustment parameter β, restart probability α.

Output: Network node similarity score matrix S.

- a) Initialize lowest-degree preference transition matrix W, node similarity score matrix $S \leftarrow O_{N\times N}$.

- b) For $i = 1$ to N, $j = 1$ to N:

- c) Calculate lowest-degree preference transition probability between nodes using the formula;

- d) Update lowest-degree preference transition matrix W;

- e) End for

- f) Calculate similarity scores between node u and other nodes in the network /* Compute node u' s similarity scores with other nodes */

---

g) End for

h) Return S

2.2 Algorithm Convergence

The convergence of the particle random walk process in the LPRWR algorithm is a necessary condition for its applicability. Below is a rigorous proof of convergence.

Theorem 1: The LPRWR algorithm is convergent.

Proof: a) Since elements w_{uv} in the lowest-degree preference transition matrix W satisfy $w_{uv} \geq 0$ and $\sum_{v \in V} w_{uv} = 1$, matrix W is stochastic. From the properties of stochastic matrices, W is irreducible.

b) The random walk process is a Markov chain. For any state, after the random walk passes through it, the number of steps required to revisit it is uncertain due to the restart probability, making the entire walk process aperiodic.

Therefore, the random walk process adopted by the LPRWR algorithm is ergodic, proving that the LPRWR algorithm is convergent.

2.3 Complexity Analysis

Theorem 2: The time complexity of the LPRWR algorithm is $O(N^3)$.

Proof: Since the probability distribution of the LPRWR algorithm converges to a stationary distribution, the key to calculating the stationary solution $\Pi = \alpha \cdot \pi_u(0) \cdot [I - (1-\alpha)W]^{-1}$ is computing the inverse of matrix $[I - (1-\alpha)W]$. The complexity of matrix inversion is $O(N^3)$, hence the LPRWR algorithm's time complexity is $O(N^3)$.

---

3 Experimental Setup

In experiments, network edges E are divided into training set E^T and test set E^P, with $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. The ratio is typically set as $|E^T| : |E^P| = 9 : 1$. The training set is considered known information for calculating scores of unconnected node pairs. An effective algorithm should assign higher scores to edges in the test set and lower scores to non-existent edges.

Ten-fold cross-validation is used to test the proposed algorithm's performance. For convenient data processing, all data is stored in CSV format in a MySQL database. The RapidMiner data mining tool is used to randomly select training and test sets according to the specified ratio. In experiments, each AUC and

Precision value is the average of no fewer than 100 independent experimental runs.

3.1 Evaluation Metrics

Mainstream evaluation metrics for link prediction algorithms include AUC (Area Under the Curve) and Precision. AUC focuses on overall discrimination of unknown objects, while Precision focuses on accurate prediction, concerning the hit ratio of top-ranked results.

AUC measures the probability that a randomly selected edge from the test set has a higher score than a randomly selected non-existent edge. During experiments, if a test set edge' s prediction score is greater than a non-existent edge's score, count 1 instance; if equal, count 0.5 instances. After n independent comparisons, AUC is calculated as:

$$\text{AUC} = \frac{n' + 0.5n''}{n}$$

where n' and n'' are the respective counts. Obviously, random prediction yields AUC ≈ 0.5. Additionally, the number of comparisons n must be considered. Lyu et al. proved that regardless of the test set proportion, taking $n = 672,400$ ensures the absolute calculation error of AUC does not exceed 1‰ with 90% confidence. Therefore, $n = 672,400$ is used in this paper' s experiments.

Precision focuses on the ratio of accurate predictions among the top L predicted edges:

$$\text{Precision} = \frac{L_p}{L}$$

where L_p represents the number of edges among the top L predicted edges that actually appear in the test set.

Nine real-world network datasets of different scales are selected for experiments, all from public network databases. These include Dolphins, Neural, Polbook, Metabolic, Netscience (NS), Football, Circuit, Facebook, and Hamster. Table 1 lists the relevant statistical characteristics of these datasets, where N and M are the numbers of nodes and edges, $\langle k \rangle$ is the average degree, $\langle d \rangle$ is the average shortest path length, r is the assortativity coefficient, H is degree heterogeneity, and C is the clustering coefficient.

Table 1. Topological parameters of nine real networks

Network	N	M	$\langle k \rangle$	$\langle d \rangle$	r	H	C
Dolphins							
Neural							
Polbook							
Metabolic							
Football							

Network	N	M	$\langle k \rangle$	$\langle d \rangle$	r	H	C
Circuit							
Facebook							
Hamster							

4 Experimental Results and Analysis

To evaluate the LPRWR method's performance, we first calculate inter-node similarity scores, then quantify prediction accuracy using AUC and Precision metrics. Following typical practices in random walk-based methods, the restart coefficient is set to $\alpha = 0.15$. Due to space limitations, only AUC results are presented below.

4.1 Impact of Parameters on AUC Results

In Equation (10), β primarily adjusts the proportion of lowest-degree preference walk, with $\beta \in [0,1)$. We investigated the effect of parameter β on prediction results, shown in Figure 2. Results indicate that compared with $\beta = 0$ (unbiased random walk), all $\beta \neq 0$ cases achieve improved prediction accuracy, with optimal precision attainable within certain parameter ranges. This demonstrates that lowest-degree preference walk is indispensable for similarity improvement. Observing each subfigure in Figure 2, AUC curves peak and then decline to varying degrees across different networks, with most networks (Dolphins, Neural, Polbook, etc.) showing relatively rapid decline. This suggests that smaller lowest-degree bias yields higher prediction accuracy. Specifically, $\beta = 0.05$ works best for Dolphins, Metabolic, and NS networks; $\beta = 0.1$ is better for Neural and Hamster networks; $\beta = 0.15$ is optimal for Polbook and Facebook networks; $\beta = 0.25$ is optimal for Football network; and for Circuit network, the optimal β is mainly distributed near 0.45. Therefore, while optimal β values differ across networks, better prediction effects can be achieved when the optimal parameter value is relatively small (e.g., between 0 and 0.2). Moreover, when AUC is optimal, β corresponds to particles biased toward low-degree nodes during walks, consistent with the RA index's philosophy that low-degree common neighbors contribute more than high-degree ones. In practical applications, smaller β values can be selected for prediction.

4.2 Feasibility Analysis

To further validate the feasibility of lowest-degree preference random walk and the effectiveness of the LPRWR algorithm, we compare its performance with eight mainstream indices (four local and four global). Table 2 shows the AUC values for each index. The LPRWR algorithm achieves the highest AUC values in eight networks, only slightly lower than RWR in the Facebook network.

Although other methods may score close to our method on certain networks, their performance varies significantly across other networks, indicating that the proposed method yields more stable predictions and has advantages across a wide range of networks, while baseline indices may only perform well on specific networks.

Among local indices (CN, PA, RA, HDI), RA penalizes high-degree nodes and performs relatively well. Among global indices (Katz, SimR, ACT, RWR), Katz considers all paths between nodes, while SimR, ACT, and RWR are based on random walk processes, with RWR showing relatively good overall performance. Using RWR as a baseline, the LPRWR algorithm improves prediction accuracy by an average of 2.14%, with a 4.48% AUC improvement on the Football network. From Theorem 2, LPRWR has the same time complexity as RWR ($O(N^3)$). With identical time complexity, LPRWR achieves better prediction accuracy, further demonstrating the effectiveness and feasibility of lowest-degree preference random walk with restart for link prediction.

Table 2. Comparison of AUC for different indices

Network	CN	PA	RA	HDI	Katz	SimR	ACT	RWR	LPRWR
Dolphins									
Neural									
Polbook									
Metabolic									
Football									
Circuit									
Facebook									
Hamster									

5 Conclusion

Accurately predicting node similarity in complex networks has practical significance for accelerating positive information propagation, preventing telecom fraud, and promoting e-commerce development. Current link prediction methods based on random walk processes mostly assume equal transition probabilities to different neighbors, ignoring detailed network structure information. This paper considers the influence of lowest-degree preference walk on particle transition probabilities, defines a lowest-degree preference function, proposes a mixed walk strategy, and applies it to random walk with restart to quantify node similarity. Extensive experiments on real networks and comparative analysis with various indices confirm the proposed method's effectiveness and feasibility, demonstrating its advantages in node similarity measurement.

The proposed algorithm is limited to unweighted, undirected, single-layer networks. Future research will focus on designing link prediction algorithms for

weighted, directed, multi-layer networks. Subsequent studies may explore more structural information affecting random walk processes and apply them to multi-layer networks to further improve link prediction accuracy.

References

[1] Tan Yangxin, Wu Junlin, Zhong Qing. Complex network [J]. Journal of Physics: Conference Series, 2020, 1601: 032011.

[2] Cannistra C V, Alanislobato G, Ravasi T. From link prediction in Brain connectomes and protein interactomes to the local community paradigm in complex networks [J]. Scientific Reports, 2013, 3 (4): 1-13.

[3] Fan Tongrang, Xiong Shixun, Zhao Wenbin, et al. Information spread link prediction through multi-layer of social network based on trusted central nodes [J]. Peer-to-Peer Networking and Applications, 2019, 12 (5): 1028–1040.

[4] Dzaferagic M, Kaminski N, Mcbride N, et al. A functional complexity framework for the analysis of telecommunication networks [J]. Journal of Complex Networks, 2018, 6 (6): 971–988.

[5] Zhou Tao, Lyu Linyuan, Zhang Yicheng. Predicting missing links via local information [J]. European Physical Journal B, 2009, 71 (4): 623-630.

[6] Gul H, Amin A, Adnan A, et al. A systematic analysis of link prediction in complex network [J]. IEEE Access, 2021, 9: 20531-20541.

[7] Lyu Linyuan, Zhou Tao. Link prediction in complex networks: a survey [J]. Physica A: Statistical Mechanics and its Applications, 2011, 390 (6): 1150-1170.

[8] Tan Suoyi, Qi Mingze, Wu Jun, et al. Link predictability of complex network from spectrum perspective [J]. Acta Physica Sinica, 2020, 69 (8): 188-197.

[9] Assouli N, Benahmed K, Gasbaoui B. How to predict crime informatics-inspired approach from link prediction [J]. Physica A: Statistical Mechanics and its Applications, 2021 (8): 125-143.

[10] Ai Jun, Liu Yayun, Su Zhan, et al. Link prediction in recommender systems based on multi-factor network modeling and community detection [J]. Europhysics Letters, 2019, 126 (3): 38003.

[11] Lyu Linyuan, Jin Cihuang, Zhou Tao. Similarity index based on local paths for link prediction of complex networks [J]. Physical Review E, 2009, 80 (4): 046122.

[12] Tong Hanghang, Faloutsos C, Pan Jiayu, et al. Fast random walk with restart and its applications [C]// Proc of the Sixth International Conference on Data Mining. Piscataway: IEEE Press, 2006: 613-622.

[13] Fu Xianghua, Wang Chao, Wang Zhiqiang. Scalable community discovery based on threshold random walk [J]. Journal of Computational Information Systems, 2012, 8 (21): 8953–8960.

[14] Zhao Haiyan, Zhang Jian, Cao Jian. Personalized App recommendation algorithm based on topic grouping and random walk [J]. Application Research of Computers, 2018, 35 (08): 2277-2280.

[15] Nassar H, Benson A R, Gleich D F. Neighborhood and PageRank methods for pairwise link prediction [J]. Social Network Analysis and Mining, 2020, 10 (1): 63.

[16] Li Ronghua, Yu Jeffreyxu, Liu Jianquan. Link prediction: the power of maximal entropy random walk [C]// Proc of the 20th ACM Conference on Information and Knowledge Management. United Kingdom: ACM Press, 2011: 24-28.

[17] Liu Si, Liu Hai, Chen Qimai, et al. Link prediction algorithm based on network representation learning and random walk [J]. Journal of Computer Applications, 2017, 37 (8): 2234-2239.

[18] Jin W, Jung J H, Kang U, et al. Supervised and extended restart in random walks for ranking and link prediction in networks [J]. PloS one, 2019, 14 (3): 1-23.

[19] Zhou Yinzuo, Wu Chencheng, Tan Lulu. Biased random walk with restart for link prediction with graph embedding method [J]. Physica A: Statistical Mechanics and its Applications, 2021 (6): 125783.

[20] Berahmand K, Nasiri E, Forouzandeh S, et al. A preference random walk algorithm for link prediction through mutual influence nodes in complex networks [J]. Journal of King Saud University-Computer and Information Sciences, 2021 (3).

[21] Elahe N, Kamal B, Li Y F. A new link prediction in multiplex networks using topologically biased random walks [J]. Chaos, Solitons & Fractals, 2021 (151).

[22] Vázquez A, Moreno Y. Resilience to damage of graphs with degree correlations [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2003, 67 (1): 015101.

[23] Wang Yan, Cao Xinxin, Weng Tongfeng, et al. Lowest-degree preference random walks on complex networks [J]. Physica A: Statistical Mechanics and its Applications, 2021, 577: 126075.

[24] Langville A N, Meyer C D. Google's pagerank and beyond: the science of search engine rankings [J]. The Mathematical Intelligencer, 2011, 30 (1): 68-69.

[25] Lyu Yanan, Han Hua, Jia Chengfeng, et al. Link prediction algorithm based on biased random walk with restart [J]. Complex Systems and Complexity Science, 2018, 15 (4): 17-24.

[26] Fronczak A, Fronczak P. Biased random walks in complex networks: the role of local navigation rules [J]. Physical Review E, 2009, 80 (1): 016107.

[27] Xu Quanzhi. Stochastic processes with its applications [M]. Beijing: Higher Education Press, 2013: 113-219.

[28] Kim T H, Lee K M, Lee S. U. Generative image segmentation using random walks with restart [J]. Lecture Notes in Computer Science, 2008, 5304 (1): 264-275.

[29] Zheng Wei, Wang Chaokun, Liu Zhang, et al. A multi-label classification algorithm based on random walk model [J]. Chinese Journal of Computers, 2010, 33 (8): 1418-1426.

[30] Hanley J A, Mcneil B J. The meaning and use of the area under a receiver operating characteristic (ROC) curve [J]. Radiology, 1982, 143 (1): 29–36.

[31] Lawera M. Predictive Inference: an introduction [J]. Technometrics, 1995, 37 (1): 121–121.

[32] Kunejis J. Konect: the Koblenz network collection [C]// International conference on World Wide Web companion. Brazil: ACM Press, 2013: 1343-1350.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv −Machine translation. Verify with original.