Journal of Computational Information Systems 11: 5 (2015) 1757–1764
Available at http://www.Jofcis.com

Link Prediction Based on Random Walks ⋆

Li LI*, Weisi FENG, Chenyang JING, Feng TAN, Ping HE, Jing WANG

Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China

Abstract

Link prediction in complex networks has been an attractive problem. Generally, when we obtained a snapshot of a network, we would like to infer which interactions are most likely to occur among the existing members in the future. This kind of problem has been extensively studied both from academia and industry. However, link prediction is challenging in practice. To address this issue. In this paper, we focused on the topology structure of the networks for link prediction. Two algorithms CN-LRW and CN-RWR are proposed based on local random walk and random walk with restart, respectively. We evaluate our approaches on three real data sets. Experiments show that CN-LRW and CN-RWR outperform LRW and RWR respectively in most cases. Incorporating node information with the network structure features seems promising in discovering the latent semantics of the network.

Keywords: Complex Network; Link Prediction; Random Walk

1 Introduction

Recently, the problem of link prediction in the complex has attracted much attention [1]. Link prediction has been studied for a long time in the computer field. It is an important research direction in the field of data mining. Typically, link prediction is a method to assess the possibility of new links generated between two nodes that haven not been linked in the network [2].

Commonly, two nodes are more likely to be connected if they are more similar. Link prediction based on node similarity has been paid attention by many researchers. The method mainly based on structural information of the network, and the computational cost of this algorithms are relatively small, so it is suitable for many real networks. L*ü* [3] concluded some algorithms, did some experiments on real networks and compared the results. Then found out the algorithms based on random walk had better performance. L*ü* and Liu [4] also proposed a method based on local random walk, which proved to have a better prediction and a lower computational complexity.

Random walk has been used in many application, and been applied on link prediction by many researchers for a long time. Don et al. [5] studied the design and analyzed of randomized on-line

*Project supported by the National Nature Science Foundation of China (No. 61170192).
*Corresponding author.
Email address: lily@swu.edu.cn (Li LI).

1553–9105 / Copyright © 2015 Binary Information Press
DOI: 10.12733/jcis13436
March 1, 2015

algorithms, and showed that this problem was closely related to the synthesis of random walks on graphs with positive real costs on their edges. Zhao et al. [6] proposed a TWU graph then they applied Random Walk on the graph to measure the relevance between terms and group them into collective viewpoints. Lars et al. [7] developed an algorithm based on Supervised Random Walks that naturally combines the information from the network structure with node and edge level attribute. Yang et al. [8] studied the trapping problem occurring on unweighted and weighted networks and found it could help improving the design of system with efficient trapping process and offered new insight into control of trapping in complex systems.

Although random walk does a good job in link prediction, the random-walk-based similarity is based only on the structure of the network. In common sense, the node's attributes are also very important to generate an edge. For example, if two nodes are in a same club, they will be friends more likely. How can we add some attributes to the classical algorithm is really a meaningful problem. Although this nodes' attributes can't be obtain in most time, we can use other attributes extracted from the network.

In our work, we consider the classical link prediction problem [9]. In the setting, we are given a snapshot of a social network at time t, and our aim is to predict the probability edges that will emerge in the network between t and a future time t'. More concretely, we are given some small complex networks at time t, such as co-author network. Then we would like to predict what new relationship will create between t and t' for each node in the network. So the problem can also be viewed as a link recommendation problem, where our goal is to suggest to each user a list of people that the user is most likely to create new relationship to. We analyze the characteristics of several networks and do some experiments on them with Local Random Walk and Random Walk with Restart. Then we compare the two algorithms and add Common Neighbour into the two algorithms so that we can come to a better result.

The remaining paper is structured as follows: In the next Section we describe the algorithms and Section 3 introduce our developed algorithm. Section 4 contains description of data analysis and the experimental evaluation. We finally conclude this paper in Section 5.

2 Background

There are many similarity algorithm based on random walk used for link prediction. In the following, we will describe two typical method which will be compared in our experiments: Local Random Walk (LRW) and Random Walk with Restart (RWR). Table 1 lists all the notations of symbols which will be referred to in herein.

2.1 Local random walk

First we consider an undirected simple network G(V, E), where V is the set of nodes and E is the set of edges. Multiple edges and self-connections are not allowed. For each pair of nodes, $x, y \in V$, There is a score, S_{xy}, which shows the the score of similarity between two nodes. The higher the score, the greater the probability of new edges create.

Random walk is a Markov chain that describes the sequence of nodes visited by a random walker [10, 11]. The key fraction of the algorithm is the transition probability matrix $\mathbf{P}$. $\mathbf{P}_{xy} = a_{xy}/k_x$ present the probability that a random walker starting at node x will walk to y in the next trap,

Table 1: The notation of symbols

Symbols	Description	Symbols	Description
$\mathbf{P}$=$\mathbf{P}_{xy}$	the transition probability matrix	k	the degree of the node
Π_{xy}	the probability of x and y in the process	α	the restart probability
S_{xy}	the similarity of the node x and y	$\Gamma(x)$	the set of the neighbours of x
N	the number of the node in the network	T	The threshold
E	the number of the edges in the network	L	the average path length
$\bar{k}$	the average degree of the network	C	the clustering coefficient
D	the diameter of the network	ρ	the density of the network
P	the complementary cumulative conditional distribution		

where a_{xy} equals 1 if node x and node y are connected, 0 otherwise, and k_x denotes the degree of node x. When set a random walker starting from node x, with the probability $\Pi_{xy}(t)$ this walker locates at node y after t traps, There is the formula

$$\vec{\pi_x}(t) = \mathbf{P}^T \vec{\pi_x}(t-1) \tag{1}$$

where $\vec{\pi_x}(0)$ is an $N \times 1$ vector with the x-th element equal to 1 and other all equal to 0, and T is the matrix transpose. The initial resource is usually assigned according to the importance of the nodes [12]. The initial resource of node x is set proportional to its degree k_x. Then, after normalization the similarity between the node x and node y is

$$S_{xy}^{LRW}(t) = \frac{k_x}{2\,|E|} \cdot \pi_{xy}(t) + \frac{k_y}{2\,|E|} \cdot \pi_{yx}(t) \tag{2}$$

where $|E|$ is the number of edges in the network. It is obvious that $S_{xy} = S_{yx}$.

2.2 Random walk with restart

This algorithm is also consider about a simple network G(V,E), There is also a transition probability matrix $\mathbf{P}$, with $\mathbf{P}_{xy} = a_{xy}/k_x$ present the probability that a random walker starting at node x will walk to y in the next trap, where a_{xy} equals 1 if node x and node y are connected, 0 otherwise, and k_x denotes the degree of node x. Differently, there is a additional restart probability, when a random walker starting from node x, who will iteratively move to a random neighbour with probability α and return to node x with probability $1-\alpha$, and denoting by q_{xy} the probability the walker locates at node y when it come to a steady state, then the formula is

$$\vec{q_x} = \alpha \mathbf{P}^T \vec{q_x} + (1-\alpha)\,\vec{e_x} \tag{3}$$

where $\vec{e_x}$ is an $N \times 1$ vector with the x-th element equal to 1 and other all equal to 0. We can get the solution straightforward, as

$$\vec{q_x} = (1-\alpha)(1-\alpha\mathbf{P}^T)^{-1}\,\vec{e_x} \tag{4}$$

Accordingly, the similarity between the node x and node y can be define as

$$S_{xy}^{RWR} = q_{xy} + q_{yx} \tag{5}$$

3 Our Algorithm

As we mentioned before, the node's attributes are very important to generate an edge, Because two nodes with similarity attributes are more similar. Considering most of the node's attributes are hidden, we find the common neighbors can be a attribute of the nodes, so we can combine it to the random walk algorithms.

3.1 CN-LRW and CN-RWR

Common Neighbour (CN) [4] is also a classical similarity algorithm. By common sense, two people, x and y, will be more likely to be friends if they have more common friends. For a node x, let $\Gamma(x)$ denotes the set of the neighbours of x, there is a simplest way to measure the CN algorithm. namely

$$S_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)| \tag{6}$$

Based on the CN algorithm, we make some improvement on that two algorithms based on random walk. As we introduced, both of the two algorithms need a transition probability matrix $\mathbf{P}$, but the matrix is just simply defined as $\mathbf{P}_{xy} = a_{xy}/k_x$, which is only related to the degree of the nodes. So we add the CN indic to the matrix. Let M denotes the max number of the CN in the networks. We have

$$\mathbf{P}_{xy} = (1 + \frac{CN}{M}) \times \frac{a_{xy}}{k_x} \tag{7}$$

Then we obtain two developed algorithms: CN-LRW and CN-RWR. Algorithm 1 gives the overview of our proposed method.

Algorithm 1 Compute the similarity matrix $\mathbf{S}$

```
1  INPUT: Datasets, α;
2  OUTPUT: the similarity matrix S;
3  Initialize: the transition probability matrix: P ← 0_{n×n};
4  for i ← 1 to n do
5      for j ← 1 to n do
6          P_ij ← (1 + CN/M) × a_xy/k_x;
7          P ← P_ij_{n×n};
8      end
9  end
10  Normalize P to obtain P^T;
11 for i ← 1 to n do
12     s_i ← e_i;
13     repeat
14     s_i ← α s_i P^T + (1 − α) e_i;
15     until convergence;
16 end
17 return S;.
```

From the 4-th line to the 9-th line, we obtain the transition probability matrix: $\mathbf{P}$, the 11-th line to the 16-th line does the random walk, finally we obtain the similarity matrix $\mathbf{S}$. Actually, we also make some normalize to the $\mathbf{S}$.

4 Experiments

4.1 Data sets

In the following, we report on the main features of our data sets and we also analyze the networks later. In this paper, we collect four social networks and the weight and the direction of the edges are ignore: (i) the dolphin network: A network representing the dolphins social interactions. (ii) the football network: A network of football teams in a American college. (iii) the power-grid network: An electrical power grid of the western US, with nodes representing generators, transformers and substations, and edges corresponding to the high voltage transmission lines between them. For each data set, we collect some basic information about the social network. Table 2 summarizes the features of our data sets.

Table 2: Dataset statistics

Dataset	N	$\|E\|$	$\bar{k}$	L	D	C	ρ
dolphin	62	129	5.129	3.357	8	0.259	0.084
football	115	613	10.661	2.508	4	0.403	0.094
power-grid	4941	6594	2.669	18.989	46	0.08	0.001

After this, we make some analysis of the structure of the networks. When considering a network, we always discuss the property of the nodes at first. Accordingly, we first analyze the distribution of the degree of the node in the networks. Further more, we use a measure which is the average nearest neighbors degree of the node u.

$$k_{nn}^{u} = \frac{1}{k_u} \sum_{v \in \nu(u)} k_v \tag{8}$$

where the sum runs over the set $\nu(u)$ of the neighbors of u.

Fig. 1 shows the complementary cumulative conditional distribution $P(n|k_n)$ in the networks, where n is the number of degree of the node. And Fig. 2 shows the complementary cumulative conditional distribution $P(n|k_{nn})$ in the networks, where n is the number of average nearest neighbors degree of the node. From Figs. 1 and 2, we can see the degree of the nodes presents a steady distribution in the football club network with most nodes having at least 10 connections to other nodes. But in the USA power-grid network, the degree of the nodes presents a wide distribution with the increase of the degrees, while in the dolphin network, no such phenomenon is observed. The degree distribution seems in a uniform way.

4.2 General consideration

First we evaluate two aspects of our algorithm: (1) the choice of random walk restart parameter α in the RWR. (2) the choice of the threshold of similarity score which we use to judge weather there is a edge between two nodes.

(1) Choice of α

To get a exhaustive consideration on the impact of the random walk restart parameter α, it is necessary to think of the extreme cases. When $\alpha = 0$, the PageRank of a node in an undirected

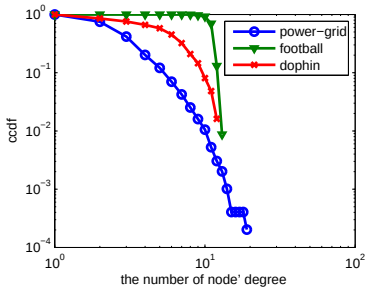

Fig. 1: Complementary cumulative conditional distribution of the degree

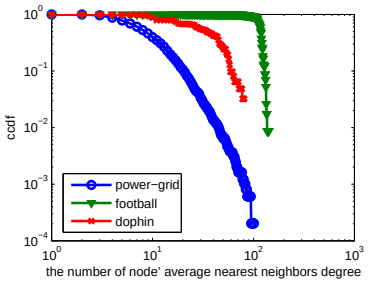

Fig. 2: Complementary cumulative conditional distribution of the average nearest neighbors degree

graph is simply its degree. On the other hand, when approaches 1, the score will be exactly proportional to the "Random-Random" model [13] which simply makes two random hops from s, as of random walk greater than 2 become increasingly. Intuitively, the value of α controls for how "far" a wanders from the initial node before it restarts and jumps back to the initial status. From the empirical study, the bigger the value of α, the shorter the local random walks, while the smaller α value makes longer walk.

We evaluate on the networks to see the impact of the α. Fig. 3 shows the results. From the definition, F-measure is a composite metric of precision and recall. In this paper, we use it to compare the different performances with various parameter settings.

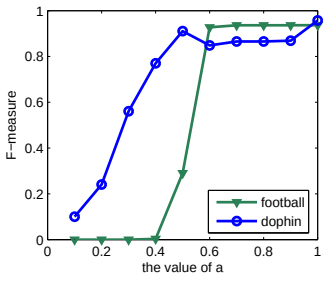

Fig. 3: Impact of random walk restart parameter α

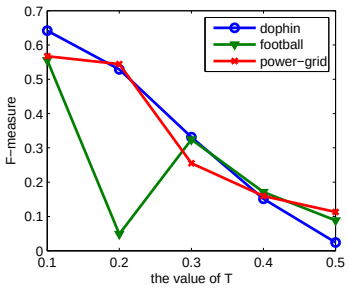

Fig. 4: Impact of random walk restart threshold T

We can observe that α really play an important role in the method, in the football network, the α performs better from 0.6 to 0.9, while in the dolphin network, the α performs best when α equals 0.5. So we can find out the α paly better from 0.5 to 0.9 in general. In this paper, we choose α equals 0.9.

(2) Choice of threshold T

The similarity score means the strength of the relationship between two nodes. With the different threshold, we may get variety result. Once the threshold T set, when the similarity score is less than T, we will define the link does not exist. Fig. 4 shows the influence. In this paper, we choose T equals 0.1.

4.3 Results and discussion

After discussing the α and T, we then compare the classical LRW and RWR with our improved method besides other three classical algorithms with different data sets. For co-author network, we divide it to two parts based on the time. One is the training set, and the other is the testing set. And for other networks, we remove 20% of the links in the network and let the remaining network as the training set, the whole network as the testing set. One prerequisite thing is to keep the connectivity of the network. In other words, the connection of the newly generated network must be guaranteed. In practice, we do check the connectivity after each deletion.

To evaluate the algorithms' performance, we use the classical metrics-precision, recall and F-measure.

Clearly, higher value of the metrics means higher prediction. But Precision and Recall always can not achieve a high value both in generally.

Tables 3, 4, 5, show the comparison of the algorithms' performance. From the result, we

Table 3: Comparison of performance Precision

Precision	RWR	CN-RWR	LRW	CN-LRW	ACT	LP	RA
dolphin	0.6880	**0.7015**	0.2297	0.2214	0.4566	0.6111	0.3613
football	0.4372	**0.7783**	0.3274	**0.7347**	0.5593	0.7710	0.7341
power-grid	0.4527	0.4373	0.1902	0.2093	0.1754	0.4800	0.1438

Table 4: Comparison of performance Recall

Recall	RWR	CN-RWR	LRW	CN-LRW	ACT	LP	RA
dolphin	0.5409	**0.5912**	0.5503	**0.6761**	0.6289	0.5535	0.3522
football	0.2580	**0.4323**	0.4233	0.2643	0.6770	0.8238	0.5090
power-grid	0.7887	**0.8076**	0.3628	0.3546	0.2153	0.6460	0.1366

Table 5: F-measure evaluation

F-measure	RWR	CN-RWR	LRW	CN-LRW	ACT	LP	RA
dolphin	0.6056	**0.6416**	0.3241	0.3336	0.5291	0.5809	0.3567
football	0.3245	**0.5558**	0.3693	0.3887	0.6125	0.7965	0.6012
power-grid	0.5753	0.5673	0.2496	0.2632	0.1935	0.5508	0.1381

can see our improved algorithms' performance is better than the classical algorithms' in most cases some metrics seem to lag behind those of the original methods, especially the power-grid network. Otherwise, LP algorithm do well in the the power-grid network. We try to find out why it is the case. Look back to the data sets statistics details in Table 2 and the degree distribution together with the average nearest neighbors degree distribution, it is clearly that in the power-grid network, both two distributions are both more widely than those of other two networks. And from the analysis of the basic information of the networks, we can also find out the power-grid network is more sparse than other two networks. We were speculating that, the sparsity of the network is a crucial factor and may influence the performance of the algorithms.

Generally, our algorithms worked properly as expected with better performance on precision metric, are in a better position with advantageous over the traditional methods especially from precision metric perspective. In other cases, the performances of our methods are very close to those of traditional ones.

5 Conclusion

In this paper, we presented two algorithms based on random walk. We analyzed the structure of the networks through the distribution of the degree of the nodes, and discussed how they influence the performance of the algorithms. Experiments reveal that our methods can achieve a better performance in link prediction tasks.

In the future work, we plan to apply algorithms in larger network with more node attributes.

References

[1] A. Clauset, C. Moore, M. E. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (7191) (2008) 98–101.

[2] R. R. Sarukkai, Link prediction and path analysis using markov chains, Computer Networks 33 (1) (2000) 377–386.

[3] L. Linyuan, Link prediction in complex network, University of Electronic Science and Technology 39 (5) (2010) 652.

[4] W. Liu, L. Lü, Link prediction based on local random walk, EPL (Europhysics Letters) 89 (5) (2010) 58007.

[5] D. Coppersmith, P. Doyle, P. Raghavan, M. Snir, Random walks on weighted graphs and applications to on-line algorithms, Journal of the ACM (JACM) 40 (3) (1993) 421–453.

[6] B. Zhao, Z. Zhang, Y. Gu, X. Gong, W. Qian, A. Zhou, Discovering collective viewpoints on micro-blogging events based on community and temporal aspects, in: Advanced Data Mining and Applications, Springer, 2011, pp. 270–284.

[7] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 635–644.

[8] Y. Yang, Z. Zhang, Random walks in unweighted and weighted modular scale-free networks with a perfect trap, The Journal of chemical physics 139 (23) (2013) 234106.

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

[10] J. G. Kemeny, J. L. Snell, Finite markov chains, vol. 28, Springer New York, 1976.

[11] J. R. Norris, Markov chains, no. 2008, Cambridge university press, 1998.

[12] E. Nummelin, A splitting technique for harris recurrent markov chains, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 43 (4) (1978) 309–318.

[13] J. Leskovec, L. Backstrom, R. Kumar, A. Tomkins, Microscopic evolution of social networks, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 462–470.