Physica A 419 (2015) 260–267

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

# Predicting link directions using local directed path

Xiaojie Wang [a], Xue Zhang [a], Chengli Zhao [a], Zheng Xie [a], Shengjun Zhang [c], Dongyun Yi [b,a,*]

[a] *College of Science, National University of Defense Technology, Changsha, 410073, China*
[b] *State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, 410073, China*
[c] *International School, Beijing University of Posts and Telecommunications, Beijing, 100876, China*

## HIGHLIGHTS

- We propose a method to predict link directions.
- Our method is stable and robust in many kinds of directed networks.
- We discuss the potential role of ground node in link direction prediction problem.

## ARTICLE INFO

*Article history:*
Received 9 April 2014
Received in revised form 13 August 2014
Available online 18 October 2014

*Keywords:*
Link prediction
Directed network
Local directed path

## ABSTRACT

Link prediction in directed network is attracting growing interest among many network scientists. Compared with predicting the existence of a link, determining its direction is more complicated. In this paper, we propose an efficient solution named Local Directed Path to predict link direction. By adding an extra ground node to the network, we solve the information loss problem in sparse network, which makes our method effective and robust. As a quasi-local method, our method can deal with large-scale networks in a reasonable time. Empirical analysis on real networks shows that our method can correctly predict link directions, which outperforms some local and global methods.

## 1. Introduction

Many real systems can be well described by networks [1], where nodes represent individuals or actors and links denote interactions or relations between them. In modern society, especially in IT industry [2], the scale of network is becoming larger and larger, which gives rise to rapid development of complex networks [3]. Recently, the research of complex networks has extended to many areas, such as biological networks [4], social communications networks [5], power grid networks [6], traffic networks [7], and so on. As a famous application, Information Retrieval (IR) [8,9] which focuses on mining unseen but useful information, has attracted much attention. One important aspect in IR is link prediction [10,11], which aims at predicting the underlying nature of networks. In fact, the problem of link prediction has been long studied in computer and information science, and many machine learning based algorithms [12,13] have been proposed. Though those algorithms are effective, their complexities become inevitable drawbacks when dealing with large-scare networks. Therefore, researches turn to making use of the structure and dynamic of networks to build methods that bear the characteristics of low computational complexity and time efficiency.

In [10], Lü et al. made a survey about link prediction algorithms for undirected networks. They classified them into three categories, similarity-based algorithms, maximum likelihood methods and probabilistic models. Similarity-based

[*] Corresponding author at: College of Science, National University of Defense Technology, Changsha, 410073, China. Tel.: +86 13755083447.
*E-mail addresses:* wangxiaojie0817@gmail.com (X. Wang), dongyun.yi@gmail.com (D. Yi).

http://dx.doi.org/10.1016/j.physa.2014.10.007
0378-4371/© 2014 Elsevier B.V. All rights reserved.

algorithms are inspired by node similarity and structural similarity. The idea about node similarity is quite straightforward: two nodes are considered to be similar if they share many common features. While, structural similarity considers more about the internal structural nature of networks, and lots of well known indices fall into this category such as *Common Neighbors* [14], *Preferential Attachment* [15], *Adamic–Adar* [16], *Resource Allocation* [17], *Katz* [18], and *Leicht–Holme–Newman* [19]. Maximum likelihood methods often try to describe the network structure with some likelihood model. Two typical algorithms of this kind are *Hierarchical Random Tree* [20] and *Stochastic Block Model* [21,22], which describe the hierarchical and community nature of networks, respectively. As another category, probabilistic models [23] pay more attention to the underlying relationship in networks. Recently, some methods considering the internal mechanic of networks have also been proposed [24–27] to solve the link prediction problem.

However, in directed networks, how to predict missing links, especially the link direction, is more complicated. Some metrics like *in-degree*, *out-degree*, *PageRank* and *LeaderRank* can be used to rank nodes in a specific order, then the link direction could be predicted as stemming from a lower-ranked node and pointing to a higher-ranked one. Besides these, Guo et al. proposed a recursive *Subgraph-based Rank* [28] method to predict link directions. These ranking-based methods are often very intuitive, but they cannot be used in a broad range of networks. Especially when the structure of network is not a tree-like one, these methods may fail to give a good prediction. Via considering the quasi-local information of network, Zhou, Lü et al. proposed the *Local Path* [17,29] method. Despite LP is often effective, it will meet serious information lost problems in some sparse networks and cannot perform well.

In this paper, we extend *Local Path* and propose a new method named *Local Directed Path*. Extensive numerical simulations on real networks show that our algorithm is promising and stable. Besides, we further analyze the parameter sensitivity and the effect of ground node in Discuss section.

## 2. Method

Consider a directed network $G(V, E)$ [30], where $V$ is the set of nodes and $E$ is the set of directed links. For simplicity, we suppose that the network is unweighted, and multiple links and self-connections are not allowed. Assuming that $|V|$ denotes the number of nodes, $|E|$ denotes the number of links, $A = (A_{ij})_{N\times N}$ denotes the adjacency matrix of the network, and $k^{in}$, $k^{out}$ denote the in-degree and out-degree, respectively.

In this paper, we compare seven algorithms for link direction prediction: *in-degree*, *out-degree*, *PageRank*, *LeaderRank*, *Subgraph-based Rank*, *Local Path* and our *Local Directed Path*. They fall into three categories: the former two are local methods, the middle three are global methods and the last two are quasi-local ones.

### 2.1. Local method

As the name implies, local method just considers the simple characteristic of node itself. In-degree and out-degree are two basic node indices, while they can be used to predict link directions easily. In a sense, if lots of nodes point to some node $i$, other nodes is more likely pointing to it. Similarly, if some node $j$ points to many nodes, it will point to more others no surprisingly. Then link directions could be predicted as stemming from low in-degree nodes to high in-degree ones, or from high out-degree nodes to low out-degree ones.

### 2.2. Global method

Local methods are often simple and easily understood, but they cannot well capture the whole structural characteristic of the network. For example, in an air transportation network, some busy transit stations may have lots of airlines from and to others airports. Even though they are quite busy and important, they are just local transitions which cannot influence the whole network more significantly than international stations. So is in internet autonomous systems network, where the degree of some local hub nodes may be higher than that of central ones, but their influence is obviously lower than the latter.

Compared with local methods, global methods care more about global structural nature of the network, which may give more reasonable solution than local ones. Here we consider three methods: *PageRank*, *LeaderRank* and *Subgraph-based Rank*.

#### 2.2.1. PageRank

*PageRank* (PR) [31] is a random walk based ranking method that forms the elemental basis of the famous Google search engine. In PR, every node $i$ is given a score PR($i$) to represent the probability of randomly browsing. It is assumed that the PR score of a node $i$ is contributed by two parts: all the neighbors pointing to it, and the random jump. A dumping parameter $d$ is introduced to describe the contribution of those two parts. With probability $d$ a walker at node $i$ can walk to some of $i's$ neighbors via directed links, and with probability $1 - d$ it jumps to a random node. The chosen of $d$ is important in the algorithm, because it not only contributes to the basic PR score of any node, but also influences the algorithmic convergence. In real implements, we use the recommended parameter [31] $d = 0.85$. The PR score of a node $i$ is calculated as follows:

$$\mathrm{PR}(i) = \frac{1-d}{N} + d\sum_{j=1}^{N}\frac{A_{ji}}{k_j^{out}}\mathrm{PR}(j) \tag{1}$$