Physica A xx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Physica A

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

Q1 Bounded link prediction in very large networks

Q2 Wei Cui [a], Cunlai Pu [a,*], Zhongqi Xu [a], Shimin Cai [b], Jian Yang [a], Andrew Michaelson [c]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, PR China

[b] Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China

[c] Department of Biomedical Engineering, Stony Brook University, NY 11790, USA

HIGHLIGHTS

- We propose a fast algorithm for calculating the number of common neighbors.
- We propose a new metric to evaluate the performance of a link prediction method.
- Our link prediction method is applicable to enormous real-world networks.

ARTICLE INFO

Article history:
Received 7 July 2015
Received in revised form 25 February 2016
Available online xxxx

Keywords:
Link prediction
Complex networks
Parallel computing

ABSTRACT

Evaluating link prediction methods is a hard task in very large complex networks due to the prohibitive computational cost. However, if we consider the lower bound of node pairs' similarity scores, this task can be greatly optimized. In this paper, we study CN index in the bounded link prediction framework, which is applicable to enormous heterogeneous networks. Specifically, we propose a fast algorithm based on the parallel computing scheme to obtain all node pairs with CN values larger than the lower bound. Furthermore, we propose a general measurement, called self-predictability, to quantify the performance of similarity indices in link prediction, which can also indicate the link predictability of networks with respect to given similarity indices.

1. Introduction

The scale and complexity of real-world systems grow unprecedentedly, which makes the prediction of such systems challenging, and on the other hand attracts increasing attention from both industry and research [1–6]. The link prediction Q3 problem is described as quantifying the likelihood of unknown associations between individuals in networks. Generally, there are two kinds of link prediction problems [7,8]. The first one is predicting future links given the current topology of a network. The second one is inferring the missing links in a static snapshot of a network. Link prediction has wide applications in social systems [9,10], biological systems [11,12], scientific systems [13,14], etc. For example, in recommender systems [15,16], they could either suggest people (items) whom you might find interesting enough, or the friends (items) you have already known, but just not yet connected online. In national safety applications, link prediction helps to identify hidden terrorist groups or criminal relationships [17,18]. In bioinformatics, link prediction is employed to find interactions

* Corresponding author.
E-mail address: pucunlai@gmail.com (C. Pu).

http://dx.doi.org/10.1016/j.physa.2016.03.041
0378-4371/

between proteins [19] or the side effects of drugs [20]. In scientific research, link prediction is used to find the scientific experts [21] or potential coauthors [22,23] based on current co-authorship networks. In all these applications, link prediction facilitates the evolution of related systems or enhances the integrality of related systems.

Link prediction methods may be broadly divided into three groups [8]: maximum likelihood algorithms, probabilistic models, and similarity-based strategies. Compared with the first two types of approaches, similarity-based strategies seem more promising, due to their simplicity and lower computational cost. In similarity-based methods, unconnected node pairs are assigned similarity scores, and node pairs with high similarity scores are assumed to be linked by edges with high probability. However, it is hard to define node similarity indices, since node attributes are not easy to obtain. Thus, many researchers propose similarity indices only with the knowledge of network structure, which is further categorized into three types according to the amount of information used in the similarity computation: local indices [24–27], global indices [28–31], and quasi-local indices [32–34]. Local indices require the information of the local structure of nodes to determine the similarity of nodes. The first and most widely studied local index is the common neighbors (CN) index [7], which quantifies the similarity of a pair of nodes as the number of neighbors they have in common. Many other local indices are extensions of the CN index [8].

Although CN-based indices are promising in link prediction, they are hard to evaluate in very large real-world networks. The current evaluation framework requires the calculation of similarities for a large number of node pairs with the time complexity of $O(|V| * \langle k\rangle^2)$, where V is the node set and $\langle k\rangle$ is the average node degree [8]. For dense networks in which the maximum node degree is comparable to $|V|$, the time complexity reaches $O(|V|^3)$, which is not acceptable for super large networks. On the other hand, current prediction accuracy metrics such as precision [35] and area under the curve (AUC) [36] have limitations when applied to very large networks. For example, the number of links in the complement graph of a large real-world network is usually much larger than that of the original network, which makes the value of precision tend to zero, and leads to a large variance of AUC. Recently, several fast link-prediction algorithms [34,37] based on the MapReduce computational model [38] were proposed in the literature. These algorithms work efficiently in the calculation of CN-based similarity indices, since they divide the computation into the map phase and the reduce phase, and the computation may be parallelized into clusters of many machines. However, these fast algorithms do not reduce the time complexity essentially, and are still not efficient for large dense networks with millions of nodes.

In fact, most of the real-world networks have heterogeneous topological structures, and link recommendations usually happen in relatively dense areas of the networks. For example, we are more interested in individuals with a large number of friends, and we recommend the "hot" individuals to others. Based on these facts, we propose a bounded link prediction framework, in which node pairs with similarity scores larger than the lower bound are selected with priority for prediction. Also, in the bounded link prediction framework the evaluation of similarity indices is with respect to a specific lower bound. In this paper, we focus on the CN index in the bounded link prediction framework. With the lower bound of CN scores, we propose a fast parallel algorithm for calculating CN values based on the MapReduce model. Moreover, we propose a new metric, self-predictability, for evaluating the performance of similarity indices. With our fast algorithm and the new metric, we can efficiently and precisely evaluate the CN-based indices for very large real-world networks.

2. Traditional algorithms based on MapReduce

In this section, we discuss the traditional MapReduce based algorithms [34,37] for calculating CN values and discuss their limitations from the perspective of computational complexity.

MapReduce [38] is a widely used programming model for dealing with searching, sorting, and other tasks related to large-scale datasets. Programmers find the MapReduce system easy to use since it automatically parallelizes tasks across large-scale clusters of machines, handles machine failures and schedules inter-machine communications. The user only needs to specify the computation task in terms of a map and a reduce function. The map function takes an input key/value pair and produces a set of intermediate key/value pairs. Then, all the intermediate pairs are grouped by the key and passed to the reduce function. In the reduce function, the values for a key are merged together to form a smaller set of values.

In the original pair generating algorithm [34], node indices and node adjacencies are specified as the key/value inputs of the map function. In the phase of "map", neighbors of a node are paired with each other, and each node pair is taken as the key and assigned a value (score) "1". In the phase of "reduce", the values for each key are summed which are the desired CN values.

The pair generating algorithm is further improved by means of vectorization [37]. In the vectorization algorithm, the value of a key is set as an accompanied group (see Fig. 1 for the illustration of accompanied groups). Thus, the times of key/value pairs transmission from the map function to the reduce function are greatly reduced. Assuming there are several node pairs incident with node $v7$ which are $(v1, v7)$, $(v2, v7)$, $(v4, v7)$, and $(v5, v7)$. For the pair generating algorithm, these node pairs are all different keys and are sent to the reduce function one by one. However, in the vectorization algorithm, these node pairs are packaged into a key/value pair, where the key is node $v7$ and the related value is $v7$'s accompanied group $\{v1, v2, v4, v5\}$, and then the intermediate key/value pair are transferred to the reduce function. The pseudocodes for the pair generating algorithm and the vectorization algorithm are in Appendix A.

We present a small example to further illustrate the above two algorithms. As shown in Fig. 1, the given network contains 8 nodes and 15 edges. First, we find the node adjacencies based on the given network. Then, with the node adjacencies as

Please cite this article in press as: W. Cui, et al., Bounded link prediction in very large networks, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.03.041