Improving Link Prediction Algorithms in Complex Networks

Hamed Zamani, Masoud Asadpour
School of Electrical and Computer Engineering
University of Tehran
January 26, 2013

Abstract
Nowadays, the link prediction problem in complex networks has attracted much attention. There are some difficulties in solving this problem, such as scarcity and huge size of networks. Most of the previous works have low efficiency. There are some solutions for this problem and we try to combine these solutions to find a better one. Our experiments in coauthorship networks show the truth of our combination formula.

1 Introduction

Recently, huge amounts of data in social networks has dramatically increased and there appears many new links into these networks [11]. Therefore, only the algorithms with low time complexity can be run on social networks and the link prediction problem in complex networks has attracted much attention nowadays[5, 6].

The link prediction is used to prefetch the documents, predict the new relationship among people and determine the advertisements [9].

These days, coauthorship networks have been studied in some researches in different aspects. The papers [7] and [2] focused on the coauthorship networks, such as DBLP network. Most of these papers tried to design new link prediction methods. Also, the probabilistic models are an other approach to solve and map the link prediction problem [10].

Link prediction is a subset of link mining which includes hypertext and web mining, relational learning and inductive logic programming [4]. In these researches, the emphasis is on the links between nodes.
2 Problem description and evaluation metrics

Consider an undirected network $G(V, E)$, which $V$ is the set of nodes and $E$ represents the set of edges or links between them. Also it is an unweighted network and we assume there are no multiple links between each two nodes. Consider a set of links $U$ which contains all the possible links in $G$. So $\mid U \mid = \mid v \mid (\mid v \mid - 1)/2$, where $\mid U \mid$ denotes the number of elements in set $U$. As a result, $U - E$ represents the links which are not existed in $G$. We assume that some of these links may be appeared in the near future.

The link prediction problem is supposed to predict the links which may be appeared later. It is obvious that we don’t have any information about the network topology in the future so we should solve the problem only with current data.

To evaluate the link prediction algorithm, we assume that we know the number of links in future network. An algorithm result is better if the number of predicted links have more similarity to the real future network. So we should consider two datasets which represent the network in two different times. The link prediction algorithms should be run on the former dataset and its result will be compared to the later one [5].

3 Related works

3.1 Previous works

There are some heuristic algorithms to solve the link prediction problem in complex networks. Some of these algorithms were described in [5] such as graph distance, common neighbors, Jaccard’s coefficient, Adamic/Adar, hitting time and SimRank. Each one of these algorithms use the network topology to predict the existence of links between two unconnected nodes. The most important parameter of evaluating these algorithms is the similarity of their prediction with reality.
3.2 Baselines

Graph Distance Algorithm

Graph Distance Algorithm predicts the links according to nodes distance. As it was mentioned in Section 2, we have an unweighted graph so the distance between each two neighbors is 1. In this method, the two closest nodes to each other have the highest probability of having a link between them. As a result, we should find the shortest path between all nodes and sort them increasingly according to distances.

The Breadth-First Search (BFS) algorithm is used to solve the Single Source Shortest Path problem in unweighted graphs. So to implement the Graph Distance Algorithm, we should run the BFS algorithm for $V$ times. The time complexity of the Graph Distance Algorithm is $O(V^2 + VE)$.

Common Neighbors Algorithm

The result of Common Neighbors Algorithm is based on the number of common neighbors of two unconnected nodes. It means that the probability of existing link between two nodes in the future is correlated with the number of common neighbors of them. In this algorithm, we should calculate the number of common neighbors between all pairs of unconnected nodes and sort the related edges in descending order[8].

The most direct implementation of this algorithm is done by computing $|\Gamma(v) \cap \Gamma(u)|$ for any two unconnected nodes $u$ and $v$, where $\Gamma(v)$ denotes the set of nodes which are connected to $v$ [5].

Hitting Time Algorithm

The Hitting Time Algorithm is related to *random walks* in the network. The hitting time from node $u$ to $v$ is correlated with the number of random surfers starting from node $v$ and passing node $u$ plus the number of random walkers from $u$ to $v$. Growth in hitting time between two nodes increases the probability of existing link between them.
The random walkers will be terminated in a small probability, otherwise it will choose one of the neighbors of the nodes which contains it as a next node.

4 Methodology

As mentioned on [5], the similarity of results between Common Neighbors and Hitting Times algorithms are really low in experience. So we decided to combine these two algorithms with Graph Distance Algorithm which has a good result. To combine these three algorithms we use linear weighted combination and fusion methods. The idea of this methodology comes from the differences in the result of mentioned algorithms. Using fusion method helps us to collect the best result of each algorithm and combine them. Also we use linear weighted combination idea to determine the importance and coefficiency of each baseline algorithm. Assume that the lists \( CN \), \( HT \) and \( GD \) describe the result of Common Neighbors, Hitting Time and Graph Distance algorithms, respectively.

The result of our methodology is

\[
\text{ResultSet} = \text{top}_\alpha(HT) \cup \text{top}_\beta(CN) \cup \text{top}_\gamma(GD)
\]

where \( 0 \leq \alpha, \beta, \gamma \leq 100 \), \( \alpha + \beta + \gamma = 100 \) and the \( \text{top}_\alpha \) of a list denotes the first \( \alpha \% \) of list’s member.

5 Experiments

5.1 Dataset

The arXiv datasets were firstly used in [5] and we chose \texttt{astro-ph} and \texttt{cond-mat} datasets which are related to coauthorship networks in astrophysics and condensed matter publications. Figure 1 shows the density of these datasets which is visualized by Gephi, an open source graph visualization and manipulation software [3, 1]. As it shows, the coauthorship network is specified as a sparse network in most of its parts and in a few ones, it is dense. Table 1 shows the number of nodes and links in this network.
### Table 1: Dataset statistics

<table>
<thead>
<tr>
<th></th>
<th>Authors</th>
<th>Articles</th>
<th>Collaborations</th>
</tr>
</thead>
<tbody>
<tr>
<td>astro-ph</td>
<td>5343</td>
<td>5816</td>
<td>41852</td>
</tr>
<tr>
<td>cond-mat</td>
<td>5469</td>
<td>6700</td>
<td>19881</td>
</tr>
</tbody>
</table>

![Astrophysics Dataset](image1.png) ![Cond-mat Dataset](image2.png)

Figure 1: The coauthorship networks

#### 5.2 Experimental Results

The $\alpha$ and $\beta$ parameters and also the probability of correct predcitions which are discussed in 4 were shown in Figure 2. Also the result of our parameters tuning are shown in Table 2. The probability of prediction of correct links is shown in this table and as it shows, the result of link prediction in our methodology has been significantly improved.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HT result</th>
<th>CN result</th>
<th>GD result</th>
<th>optimum $alpha, beta$</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>astro-ph</td>
<td>0.035</td>
<td>0.057</td>
<td>0.032</td>
<td>0.1, 0.8</td>
<td>0.071</td>
</tr>
<tr>
<td>cond-mat</td>
<td>0.040</td>
<td>0.059</td>
<td>0.044</td>
<td>0.3, 0.6</td>
<td>0.072</td>
</tr>
</tbody>
</table>

### 6 Conclusion and future works

The goal of this work was to investigate some link prediction methods in social networks. Also the result of link prediction algorithms in presented methodology was significantly
Figure 2: Parameter tuning: the probability of correct prediction with $\alpha$ and $\beta$ parameters in both datasets improved. As a result, combining the link prediction methods helps to improve the efficiency in link prediction.

In this work, we assumed a linear weighted combination of three different algorithms. For future work, the nonlinear combination may have a better result. Also, we can combine more algorithms to improve the accuracy of the method. In addition, we assumed that the parameters of linear weighted combination should be constant numbers, while we can enhance these parameters through learning with some different datasets and improve its efficiency.

7 Acknowledgement

We thank Mr. Ehsan Sherkat for his useful suggestions. We are also grateful to Mr. Mostafa Dehghani and Mr. Hasan Nasr Esfahani for helping to discover the social networks field.
References


