Link Prediction for Complex Networks via Random Forest

Kuanyang Li and Lilan Tu
College of Science, Wuhan University of Science and Technology, Wuhan 430065, China.
E-mail address: tulilan@wust.edu.cn

Abstract. In this paper, based on four existing similarity indexes (CN, LHN-II, COS+ and MFI), we obtain eigenvectors by extracting the features of two arbitrary complex network nodes. The core idea of this paper is to use decision trees to handle these four different indexes which are not strongly related. After training and learning with the random forest algorithm, a new link prediction algorithm for complex networks is proposed. We prove by conducting some numerical simulations, using the US aviation network as an example, that the proposed link prediction algorithm is more accurate and stable than other similar algorithms.

1. Introduction
Following on the rapid development of research regarding complex networks, link prediction has recently begun to attract much more attention across a number of fields and it has achieved a series of significant research results [1-10]. Existing research mostly focuses on the development and application of link prediction algorithms. Various researchers [1-6] have examined in detail how to construct link prediction algorithms using local path similarity indexes, local naive Bayes models, local degree blocking models, information allocation indexes, local weighted paths, and matrix perturbation and decomposition. In [7-8], the research has focused on the similarity measures for each meta-path in a topological structure, using this as a possible solution to the link prediction problem in heterogeneous complex networks. In [9], a plurality of features in a topological structure was combined to form the basis of an adaptive model for link prediction. In [10], the research for using link prediction algorithms for community detection was analyzed.

It can be both costly and difficult for us to obtain the attribute information of the nodes in a network. Existing link prediction algorithms have therefore mostly been based on similarity definition, using the structural information about a network [1-2,4]. An important assumption underpinning all such algorithms is that the greater the similarity of the nodes in the network, the greater the possibility of links existing between them. Many algorithms have thus been based on individual similarity indexes, such as the common neighbors similarity index (CN) [11], the path similarity index (LHN-II) [12], the random walk similarity index (COS+) [13], and Matrix Forest Theory (MFI) [14]. The CN assumes that the more neighbors shared by two nodes, the more likely it is that they will be connected. Its advantage lies in its low computational complexity. However, whilst its prediction accuracy is high in networks with a high agglomeration coefficient, it is low if the agglomeration coefficient is low. The LHN-II was proposed as a means of drawing upon general equivalence. General equivalence focuses on whether two nodes have the same role in a network, even if there is no common neighbor node between them. However, if neighbor nodes have a similar status, the two nodes will also be similar and therefore tend to be connected. The COS+ is derived from a random walk process. As a result, its computational complexity is high. The MFI, however, has managed to achieve some good results in collaborative recommendation systems.
In this paper, we report on an approach we have adopted to improve upon the algorithms mentioned above. Primarily, this involves the comprehensive consideration of all four similarity indexes (CN, LHN-II, COS+ and MFI) together. This combination can then be applied to link prediction. First of all, for all four indexes, we acquired the scores for the connection between any two nodes in a network according to the similarity features when each similarity index is seeking. Eigenvectors were then constructed for these arbitrary node pairs. After this, the edge-to-edge prediction between each pair of nodes formed the basis of a supervised learning approach derived from machine learning that is founded upon a random forest method. On the basis of all these, we then propose a new link prediction method that recognizes the complementarities of the different indexes and combines them together. Finally, some simulations using the US aviation network is reported that, when compared to the results produced by using any one single local similarity index-based algorithm, the algorithm proposed here exhibits significantly better prediction accuracy.

2. Link Prediction Algorithm based on Random Forest

In this section, we will now propose a new link prediction algorithm for complex networks using the eigenvectors of any two network nodes.

The goal of this section is to propose a new link prediction algorithm. To do this, a score will be assigned to each pair of nodes in a network. This score can be understood to be a kind of proximity that is positively related to the connection probability existing between any two nodes. All unconnected node pairs are sorted according to the score values, from the largest to the smallest. The probability that the top-ranked node pairs are connected to each other will then be the largest, and so on. At the same time, in this section, we will create random forests to avoid over-fitting.

Let us consider an undirected network $G(V, E)$, where $V$ and $E$ are the sets of nodes and links, respectively. $U$ is a complete set of $N(N-1)/2$ node pairs. In order to measure the algorithm’s accuracy, the set $E$ is randomly divided into two parts: the training set $E^t$ (treated as known information); and the probe set $E^p$ (which is used for testing). From here onwards, the links in $E^p$ will be called missing links and the links in $U - E^t$ will be called non-observed links.

The algorithm has the following four steps:

**Step 1:** Construct a data set $E$, an attribute set $F$ and a class set $C$.

Given a complex network with $N$ nodes, the elements in data set $E$ can be constructed from the arbitrary node pairs $(v_i, v_j)$ of the network. For any node pair $(v_i, v_j)$ of the network, the four similarity indexes, CN, LHN-II, Cos+, and MFI can be calculated and these can then be viewed as four features of the node pair. All of these features of each node pair are used to compose an attribute set $F$. For any node pair $(v_i, v_j)$, if there is a connection between them in the original network, they will be given the class label 1. Otherwise, they will be given the class label 0. In this way, the class set $C$ is obtained.

**Step 2:** Divide the data set $E$.

In order to test and compare the performance of the proposed algorithm, we need to select a part of the known data set $E$ as a test set $E^t$. For the purposes of this paper, we will randomly select 70% of the links in $E^t$ to build the training set $E^t$. The remaining 30% of the links will form the probe set $E^p$.

**Step 3:** Construct the random forest.

(1) Select $n$ samples from the training set $E^t$ using the bootstrap sampling method. Bootstrap sampling is a method that selects $n$ samples repeatedly. Breiman [15] found that if $n$ tends to infinity, about 36.79% of the samples will not be selected. This is called out-of-bag data (OOB). This data can be used to replace the test set $E^t$ when evaluating the accuracy of the algorithm.

(2) Randomly select $K$ attributes in the attribute set $F$, calculate the information gains and select the best segmentation attribute as a node to create a decision tree.

(3) Repeat the preceding two steps $m$ times to establish $m$ decision trees. These $m$ decision trees constitute the random forest.

**Step 4:** Evaluate the algorithm.
After the previous three steps, in order to evaluate the link prediction model for the given complex network, we can use the ‘area under the receiver operating characteristic curve’ (AUC) [16] and the precision [17] to establish the performance of the model. AUC is the most commonly-used method for measuring the stability of a link prediction algorithm. Precision, however, relates to the proportion of accurate predictions in the set.

3. Results

In this section we will be testing and evaluating the link prediction algorithm proposed above by applying it to the US aviation network. This includes 332 airports and 2126 airlines [18], with each node in the network corresponding to an airport. If there are direct flights between two airports, then there is an edge between the two nodes corresponding to those two airports. According to the algorithm step 1 proposed in the section 2, let the number of positive samples with connections between nodes be 2126. And then, the number of negative samples without connections between nodes is $C_{332}^2 - 2126 = 52820$. At this time, the computational complexity is $O(n^2)$. In the following simulations, while optimizing random forest parameters, we take the strategy of pruning but not undersampling or oversampling. For a decision tree, we assume that the minimum number of split samples in a node is 10. If the number of samples included in a node is greater than 10, then (possibly) branch, otherwise, no branch is performed.

The precision of the OOB, the training set, and the test set based on the decision tree model are shown in Table 1. The table shows that the method has a high level of precision when constructing the decision tree model. This applies also to the precision of the training set and the test set. Table 1 also indicates that OOB data can be used as an approximate substitute for the test set when measuring the model’s precision.

![Figure 1](image)

**Figure 1.** ROC curves and AUC values for the different decision trees

Fig. 1 shows the ROC curves for different combinations of the four attributes. ROC curves can be used to evaluate the classification effect of a classifier. The AUC value can be used to measure the accuracy of a link prediction algorithm. Looking at Fig. 1, it can be seen that where the area enclosed by the ROC curve is larger, so, the predicted results will be better.
Table 2. AUC for different algorithms

| Index   | CN   | LHN-II | COS+ | MFI  | CN+LHN |
|---------|------|--------|------|------|--------|
| AUC     | 0.954| 0.610  | 0.958| 0.941| 0.954  |
| Index   | CN+COS| CN+MFI| LHN+COS| LHN+MFI| COS+MFI|
| AUC     | 0.973| 0.958 | 0.960| 0.956| 0.971  |

In Table 2, the AUC evaluation results of the models with single index provided by [2] are shown between the 2nd and 4th column in the 2nd row. The other results are obtained using the method proposed in this paper. This makes it evident that the decision trees constructed according to this paper achieve better results. The model itself also shows better stability and better accuracy.

4. Conclusion

In this paper, we have therefore used four existing algorithms in combination to construct a more effective similarity-based link prediction algorithm. We have investigated the viability of combining a machine-learning-based random forest algorithm with the various similarity indexes. Taking the real example of data pertaining to the US aviation network, we have conducted some numerical simulations that illustrate that the proposed algorithm has an improved AUC value and better stability as a consequence. The results presented in this paper constitute an important development in the in-depth research of link prediction for complex networks. The paper provides new ways of revealing the behavior and evolution of complex networks and can also serve as the theoretical basis for the actual networks.

5. Acknowledgements

The authors acknowledge the interesting comments of anonymous referees. This work is supported by the National Natural Science Foundation of China under Grant 61473338.

6. References

[1] Lü L Y and Zhou T 2009 Physical Review E. 80 046122
[2] Lü L Y and Zhou T Z 2011 Epl. 96 48007
[3] Liu Z, Dong W and Fu Y 2014 Chaos An Interdisciplinary Journal of Nonlinear Science, 25 1150-1170
[4] Pei P, Liu B and Jiao L 2016 Physica A: Statistical Mechanics and its Applications. 470 1-11
[5] Yao Y, Zhang R and Yang F 2017 International Journal of Modern Physics C. 28 4
[6] Xu X, Liu B and Wu J 2017 Scientific Reports. 7 14724
[7] Shakibian H, Charkari N M and Jalili S 2016 Journal of Computational Science. 17 73
[8] Shakibian H and Charkari N M 2018 Scientific Reports. 7 44981
[9] Ma C, Bao Z K and Zhang H F 2017 Physics Letters A. 381 3369-76
[10] Cheng H M, Ning Y Z and Yin Z 2017 Modern Physics Letters B. 32 1850004
[11] François L and White H C 1977 Social Networks. 1 67-98
[12] Leicht E A, Holme P and Newman M E 2006 Physical Review E. 73 026120
[13] Fouss F, Pirotte A and Renders J M 2007 IEEE Transactions on Knowledge and Data Engineering. 19 355-369
[14] Chebotarev P and Shamis E 2006 Automation & Remote Control. 58 1505-14
[15] Breiman L 1996 Machine learning. 24 123-140
[16] Hanley J A and McNeil B J 1982 Radiology. 143 29-36
[17] Herlocker J L, Konstan J A and Terveen L G 2004 ACM Transactions on Information Systems (TOIS). 22 5-53
[18] U.S. Air Network data http://vlado.fmf.uni-lj.si/pub/networks/data/mix/USAir97.net