**Abstract**

Link prediction in complex networks is always a key research direction. The current mainstream algorithms based on similarity index have an obvious drawback that they perform well only in some specific network. In this paper, we study link prediction as a supervised learning problem. Along the way, we identify a set of features according to the similarity index we proposed and five other well-known similarity indices, and adopt supervised learning algorithms in six networks from different fields. The high prediction accuracy compared with other five well-performed methods demonstrates that link prediction using supervised learning algorithm can be applied in various networks.

**1 Introduction**

Many systems in real world, such as social, power, transportation system, can be described by networks, in which nodes represent entity (individuals, power station, etc.), and edges illustrate the relations between nodes. In recent years, researchers have paid more attention on the evolution of networks, where the most fundamental problem is link prediction. Based on the observed structure of network, link prediction aims at discovering the links missed and the high probability links in the future.

Many frameworks of link prediction algorithms proposed before are mostly based on Markov chains and machine leaning. Another group of algorithms proposed recently are based on the maximum likelihood estimation. These algorithms have an obvious advantage that can offer an excellent insight into the network organization. However, they are time-consuming and can’t handle the case when network consists of millions of nodes. Besides, they can’t get a relative high accuracy compared with other algorithms.

Currently, researchers tend to design the algorithms upon the node similarity, namely two nodes have a high chance to be connected if they’re similar to each other. These algorithms are so-called similarity-based algorithms. The measurement on node similarity can be determined by attributes of node and network structure. However, it’s usually difficult to collect the information about the attributes of nodes. Therefore, similarity indices mainly depend on network structure. According to the amount of information used on the network, similarity indices could be sorted by three types, local indices, global indices, and quasi-local indices. Local indices need less network information, while global indices ask for the whole information, making it time-consuming but accurate. Quasi-local indices act a tradeoff role between local indices and local indices.

There are still many problems pointing to the similarity indices to study. At present, most similarity indices seldom consider the subgraphs in local structure, taking triangle for example. Moreover, a lot of similarity indices, such as common neighbors (CN), Katz Index, Local Path Index, Adamic-Adar Index, perform well only in some specific networks. For example, the common neighbor index performs better when the network’s cluster coefficient is high, while worse when applied to networks like power or router whose cluster coefficient is low. So it’s natural to come up the idea that combining these similarity indices together would make up each index’s drawback. Machine learning is a good way to accomplish it.

In this paper, we proposed an index to characterize the node similarity from the view point of subgraph. Based on this index and five other famous similarity indices, we study link prediction as a supervised learning task by extracting a set of features from them. In this way, we get competitive performances in various networks.

**2 Problem Description and Experimental Setup**

This paper mainly focuses on the unweighted and undirected network , where represents the set of nodes and  denotes the set of the links observed. We study the link prediction as a supervised learning task. The following explanations introduce the process to construct dataset.

For a pair of node, we define the label  as follows:



It interprets that the links observed in current network are positive class, while the non-observed links are negative class. Furthermore, we divided  into two parts,  and . They stand for the positive class of the training dataset and the test dataset respectively. Apparently,  and. In the same way, we can get the negative class by dividing the  ( is the universal set). The ratio between training set and test set is 0.9 initially.

We obtain a binary classifier to predict whether there exists link relation between a pair of node  by training on the dataset, and verify the classification model on test data.

**3 The Approach**

**3.1 Similarity index based on local structure**

The similarity between two nodes is also related to their local structure. Triangle, a kind of subgraph in local structure, is among the most special one. Enlighted by the Preferential Attachment Index, we propose a similarity index upon local structure.

Let  be the total number of nodes in network. According to the graph theory, a network could be represented by an  adjacency matrix, where the entry  if a pair of node  and  are directly connected and  otherwise.

The diagonal element of the three power of the adjacency matrix represents two times the total number of triangle that contains a specific node. For a certain node , we define the total number of triangle containing  as, so the similarity index reads,



Similar to Preferential Attachment Index, the mechanism of this index can also be used to generate the scale-free network. Above all, it does not need too much information, which would greatly reduce the computational complexity. The index is then used to construct the feature of dataset in next chapter.

**3.2 Feature Set**

The feature extracted is essentially important to the effect on classifier, and it should reflect the proximity between a pair of node to some extent. This chapter provides a simple description of features we used for supervised learning. Due to the difficulty in acquiring the node attributes, the features are all extracted from the network’s topological and could be applied to any other networks. The first five features are based on the five different similarity indices. The last feature is based on the similarity index we proposed. These features are extremely cheap to compute. They are listed as follows.

1. common neighbors feature

We let  denotes the set of neighbors of a node, so for a pair of node, the common neighbors feature is defined:



1. sum of neighbors feature

The common neighbors feature is the intersection between two nodes, while sum of neighbors feature is the union of two nodes:



1. Preferential Attachment feature

Preferential Attachment feature uncovers such a phenomenon that the more neighbors the entities owns, the higher connection possibility will occur between them. It can be calculated by the product of the neighbors of two nodes.



1. Katz feature

Before giving the definition of Katz feature, we first introduce a kind of global similarity index, Katz index. The mathematical expression reads,



Where  denotes the path with length  between and, represents the damping factor. Katz feature is a special case whenand, that is



1. Adamic-Adar feature

The main idea of Adamic-Adar feature is that common neighbors with low degree (the number of nodes directly connecting with a specific node) would contribute more than those with high degree. Adamic-Adar feature can be obtained by follows:



1. Triangle feature

Based on the similarity proposed above, we define the feature as follows:



**3.3 Classification Algorithms**

There is a variety of classification algorithms for supervised learning. Some may work better than other for a specific dataset. In this paper, we experimented with five different classification algorithms. They are SVM, Random Forest, Naïve Bayes, Boost, KNN.

**4 Results**

In this paper, we use six representative datasets from different fields. For more details, see xxx. The basic topological of these networks are summarized in Table 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| network | V | |E| | C | <d> | <L> |
| USAir | 332 | 2126 | 0.749 | 12.807 | 2.46 |
| Jazz | 198 | 2742 | 0.617 | 27.697 | 2.23 |
| NetScience | 379 | 941 | 0.798 | 4.823 | 4.93 |
| Celegans | 297 | 2148 | 0.308 | 14.456 | 2.46 |
| PB |  |  |  |  |  |
| KJ |  |  |  |  |  |

Table 1 basic topological of the seven networks. and  denote the total numbers of nodes and links.  is the clustering coefficient.  represents the average degree of the network.  is the average path length of the network.

In this paper, we compared the performance of five classification algorithms with five similarity-based methods on the datasets mentioned above. The similarity indices include two local ones: common neighbors (CN), Adamic–Adar Index (AA), two global ones: Katz index, Average Commute Time (ACT), and a quasi-local index: Local Path (LP). These indices are detailed introduced in xxx.

As shown in Table2, We use the standard metric AUC to verify the performance of the algorithms.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AUC | SVM | RF | NB | Boost | KNN | CN | AA | LP | Katz | ACT |
| USAir | **97.3** | 94.45 | 94.47 | **96.39** | 92.23 | 95.4 | 95.7 | 94.2 | 94.4 | 90.05 |
| Jazz | **96.16** | 96.47 | 94.6 | **96.58** | 92.23 | 95.3 | 96.1 | 94.7 | 95.1 | 78.3 |
| NetScience | **99.66** | 99.59 | 99.54 | 99.43 | 98.72 | 99.2 | 99.2 | 99.5 | **99.9** | 58.1 |
| Celegans | 84.76 | 77.74 | 85.18 | 84.72 | 78.06 | 73.97 | 84.2 | **85.9** | **86.6** | 73.98 |
| PB | 92.91 | 88.05 | 92.43 | **92.93** | 87.82 | 92.18 | 92.44 | 92.83 | **93.3** | 89.65 |
| KJ | 97.2 | 96.28 | 97.86 | **97.91** | 92.14 | 97.49 | **98.35** | 97.58 | 97.47 | 79.6 |

Table2: Comparison of algorithms’ performance measured by AUC on eight networks. Each algorithm’s AUC is calculated by averaging 1000 times experiments. The two highest AUC value is emphasized by boldface.

The result in Table 2 shows that link prediction using supervised learning performs better than the methods based on similarity index on most networks. Among all the classifiers, SVM and Boost outperform all the others.

For further analyzation, we compare the performance of the classifiers using performance metrics, accuracy, precision, recall, F1 value, in Jazz network. The results are shown in Table 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SVM | RF | NB | Boost | KNN |
| Accracy | 91.46 | 84.29 | 86.83 | 90.1 | 88.28 |
| Precision | 91.61 | 96.31 | 89.32 | 89.45 | 87.21 |
| Recall | 91.27 | 71.27 | 83.64 | 90.91 | 89.67 |
| F1 | 91.44 | 81.92 | 86.38 | 90.17 | 88.43 |

It clearly shows that the classifier SVM and Boost outperform most other classifiers.

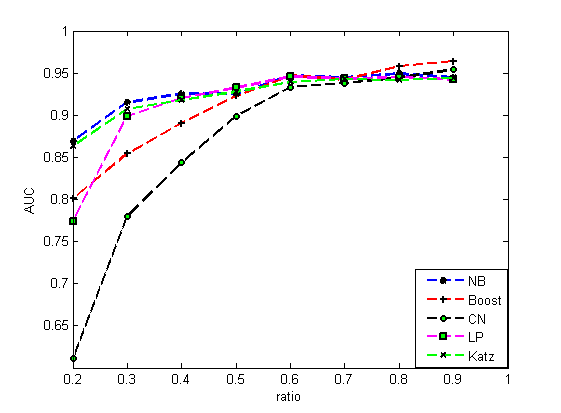
To illustrate the importance of the feature based on the similarity index we proposed, we leave this feature out, and reconstruct the feature set to experiment in PB network. The results are presented in Table 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AUC | SVM | RF | NB | Boost | KNN |
|  | 92.62 | 83.54 | 92.11 | 92.81 | 86.06 |
|  | 92.91 | 88.05 | 92.43 | 92.93 | 87.82 |

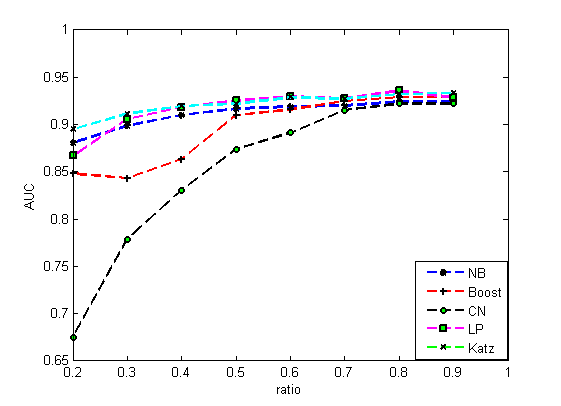
Table4: Comparison of algorithms’ performance measured by AUC on two cases that whether the triangle feature is included or not. The last row shows the result when this feature is considered.

We could make a conclusion that the triangle feature has a positive effect on most networks. It could make the classifier have a relative good discriminating ability. Though the accuracy is not improved remarkably, it means a lot for a large scale network.

For each network, the ratio between the size of training set and test set is always 0.9. In order to analyze the robustness of the classification algorithm, we change the proportion between them from 0.2 to 0.9, and obtain the following figures showing the performance measured by AUC on USAir network and PB network.



Comparison of algorithms’ performance measured by AUC on condition that ratio between the size of training set and test set changes from 0.2 to 0.9.



It can be inferred that the methods based on supervised learning are not that sensitive to the experimental setups mentioned above, while similarity-based methods, especially common neighbors index, performs worse on such a condition.

**5 Conclusion**

In this paper, we proposed a similarity index for link prediction based on local structure of the network. According to this similarity index and five other well-known similarity indices, we extract six topological features which are then used to construct feature set for five classification modes. We compared the classification algorithm with five well-performed methods. The results show that our algorithm can give a competitive accuracy, demonstrating that it’s feasible to mode the link prediction as a classification problem.

**References**