A Novel Link Prediction Algorithm Based on Deepwalk and Clustering Method

Lijun Cai¹, Jibin Wang¹², Tingqin He¹, Tao Meng¹ and Qi Li¹

¹College of Information Science and Engineering, Hunan University, Changsha 410082, China
E-mail: ¹jbwang@hnu.edu.cn

Abstract. Link prediction has attracted more and more attention due to its wide application in social network analysis, bioinformatics, and personalized recommendation. One of the methods for judging whether two nodes have connections in the network is to calculate the similarity. This method not only has low computational complexity, but also can achieve better prediction results, and it’s more suitable for large-scale networks. There are many similarity indexes proposed until now, but most of them only consider degree of the node and its common neighbors. With the proposal of Deepwalk, many people applied it to link prediction. However, the method of judging the similarity between two nodes simply by their distance is also one-sided. In this paper, we propose a new similarity index, called Deep Affinity (DA) index, through combining the traditional similarity index with the distance index obtained by Deepwalk and introducing the idea of clustering at the same time. After conducting experiments on different network datasets, the results show that DA-based link prediction algorithm has greatly improved the prediction accuracy, especially for large-scale network datasets.

1. Introduction
Link prediction [1] in the network is to predict the possibility of connection between two unconnected nodes in the network through limited informations. We always apply it to find out missing link, future link and spurious link. The main ideas of early research on link prediction are based on Markov chain and machine learning. Sarukkai used Markov Chain to do link prediction and path analysis [2]. Zhu et al. extended it to the website prediction on WWW [3]. The link prediction algorithm in recent research is divided into two types: one is the algorithm based on maximum likelihood estimation, while it can’t be widely used because of its huge computational complexity; the other is based on the similarity index, it’s favored by scholars because of its linear computational complexity and good prediction accuracy [4].

The key to the similarity algorithm is similarity index and there are dozens of them proposed up to now. In 2014, Bryan Perozzi et al. proposed Deepwalk [5], a method to express nodes in network as vectors. Afterwards many scholars use these vectors to compute nodes’ distances as similarity index to predict links. In 2017, Sun et al. proposed the LAS algorithm [6] and make improvement on three indexes previously mentioned.

In this paper, we propose a new link prediction index, DA index, and implement the link prediction algorithm based on Deepwalk and clustering methods. Experiments have been performed on four real network datasets and our results show that it outperforms traditional link prediction algorithms mentioned above.
2. Preliminaries

2.1. Deepwalk

Word2vec became the basic part of deep learning in Natural Language Processing since Tomas Mikolov presented it in [7]. Its basic idea is to express every word in the natural language as a short vector of unified meaning and unified dimension.

Based on this idea, Bryan Perozzi et. al. proposed a method called Deepwalk, which uses skipgram in word2vec to represent the nodes in the network as words in the natural language.

Random Walk is a basic tool used by Deepwalk to extract node information. Deepwalk first walks node \( v_i \) to generate a sequence \( W_i = W_i^1, W_i^2, \ldots, W_i^k \) and then iteratively updates the sequence as a sentence of language model. The language model uses the skipgram and its principle is mainly to find the optimal solution model \( \Phi : V \mapsto V \in R^{V \times d} \), aiming to maximize the average probability of \( v_i \) walking to its neighbors. The model is defined as follows:

\[
\begin{align*}
\min_{\Phi} & \quad - \log \Pr(\{v_{i-w}, v_{i-1}, v_{i+1}, v_{i+w}, \ldots, v_{i+kw}\} | \Phi(v_i)) \\
\end{align*}
\]

2.2. K-means

K-means is an algorithm that finds K clusters of a given dataset, and the number of clusters K is specified by users. Here we use Elbow Method to determine the best K value for each dataset. Here is the main workflow:

1. Randomly find k initial nodes as centroids.
2. Each node in the dataset is assigned to one cluster of the nearest centroid, and then the centroid of each cluster is updated to the average of all nodes in the cluster.
3. Loop step 2 until the cluster allocation result no longer changes.

The algorithm inputs the position vector of each node in the network which we use the vector set obtained by Deepwalk in this paper. The output is the cluster number set to which each vector belongs. Two nodes have same cluster number means they are in the same cluster.

3. Deep Affinity Similarity Model

Our paper has carried out some analysis and improvement based on the above work. Deepwalk uses the distance computed by vectors as the index of similarity to predict. The traditional link prediction method uses local similarity index, which only needs to obtain information of neighbors and degrees of two nodes to calculate the possibility of their connection. In [6], Sun et al. proposed an idea that suppose a, b have n common neighbors and b, c also have n. Although a, b, and c are in the same basketball club, c also joins football club because he loves sports and is closely related to them. It can be seen obviously that the connection between a and b is much stronger than that of b and c, so they proposed LAS index.

Sun et al. consider the case where three nodes are in the same cluster, but sometimes there are a, b in the same group, and b, c are not in the same cluster. Although similarity values calculated by LAS index meet \( S_{ab} = S_{bc} \), the possibility of b, c knowing each other is still lower than that of two people in the same club.

Based on above point of view, for any nodes a, b, we define the Deep Affinity (DA) index as follows:

\[
S_{ab} = \alpha \times \left( \frac{\Gamma(a) \cap \Gamma(b)}{k_a} + \frac{\Gamma(a) \cap \Gamma(b)}{k_b} \right) \times \text{flag}(a,b) + \beta \times \frac{1}{\text{dis}(a,b)+1}
\]

We decided to cluster nodes in the network first. There are many clustering algorithms and in this paper we use K-means method to introduce a parameter flag. When nodes a and b belong to the same cluster, the value of flag is 1, otherwise it’s -1. In this way, even if they have the same number of common neighbors, similarity of the two nodes in different cluster will be significantly lower than the
two nodes in the same. We also found that adding the distance of vectors generated by Deepwalk to this index also improves accuracy of the prediction algorithm and introduced two parameters $\alpha$ and $\beta$, where $\alpha+\beta=1, \alpha \in [0,1]$. As shown in Figure 1, nodes 15 and 19 in the same cluster have more probability to have connection than nodes 15 and 23, our prediction result also proved that.

Figure 1. Testing example

4. Prediction Algorithm

According to our proposed DA index, our link prediction algorithm is designed as follows:

- Step1: Traversing the graph with Deepwalk to generate vectors
- Step2: Clustering nodes by using K-means
- Step3: Determining values of $\alpha$ and $\beta$ to calculate $S_{ab}$

Algorithm 1 Link prediction algorithm based on DA index

**Input:** Graph $G(V,E)$

- embedding size $d$
- id of test nodes $a,b$

**Output:** $S_{ab}$

1: Use Deepwalk to get the representation vector $v_i \in \mathbb{R}^{1 \times d}$ of each node in graph $G$
2: cluster(a)=K-means($v_a$), cluster(b)=K-means($v_b$)
3: if cluster(a)=cluster(b) do
4: flag $\rightarrow 1$
5: else
6: flag $\rightarrow -1$
7: end if
8: $\text{dis}(a,b) \rightarrow [(v_{a1} - v_{b1})^2 + \ldots + (v_{a_d} - v_{b_d})^2]^{1/2}$
9: for $\alpha = 0$ to 1 do
10: $\beta \rightarrow 1 - \alpha$
11: Calculate $S_{ab}$ using formula (2)
12: $\alpha \rightarrow \alpha + 0.1$
13: end for

5. Experiments

5.1. Datasets

The experimental datasets we selected are all from realistic network like Football [8], Jazz [9], Yeast [10], P2P [5] (see table 1).
Table 1. The size of each network set.

| Network | |V| | |E| |
|---|---|---|---|---|---|
| Football | 35 | 118 |
| Jazz | 198 | 2742 |
| Yeast | 2284 | 6646 |
| P2P | 6301 | 20777 |

5.2. Evaluation Method

In order to evaluate the quality of link prediction algorithm, we randomly divide the sample set E (edge set) into training set ET and test set EP on the basis of connectivity in training set. Obviously, ET ∩ EP = ∅, ET ∪ EP = E. We generally introduce AUC (Area Under the receiver operating characteristic curve) to judge the quality of link prediction algorithm. Its calculation process is to randomly select one edge at missing edges’ set (EP) to compare with a randomly selected edge in non-existent links’ set (U-E), and to compare n times independently. The number of situations where the score of missing edge is greater than non-existent link is denoted as n’, and the number of situations with the same score is denoted as n”. AUC is calculated as follows:

\[ AUC = \frac{n' + 0.5n''}{n} \]  

(3)

If the score of link is generated randomly, the probability of occurrence of the case where the missing edge’s score is smaller than the non-existent link’s is the same, then n’=n”=1/3n, and AUC=0.5 is easily obtained. Therefore, how greater AUC is than 0.5 measures the prediction accuracy of the algorithm.

5.3. Results

We set n=100 to do experiments on four network datasets and the ratio of training set is 0.1, 0.2, 0.3, and 0.4. From the experimental results, we can see that AUC decreases with the increase of missing edges. As shown in Figure 2, DA index is more accurate than LAS, Jaccard, Salton, and Sorensen, even it’s more obvious in larger networks.
Figure 2. Experimental results for each dataset

| Fraction of Test Set | Football | Jazz | Yeast | P2P |
|----------------------|----------|------|-------|-----|
| AUC                  |          |      |       |     |
| $\alpha = 0.1$       | 0.98     | 1.00 | 1.00  | 1.00|
| $\alpha = 0.0$      | 0.99     | 0.99 | 0.99  | 0.99|
| $\alpha = 0.5$      | 0.95     | 0.95 | 0.95  | 0.95|
| $\alpha = 0.6$      | 0.9      | 0.9  | 0.9   | 0.9|
| $\alpha = 0.7$      | 0.9      | 0.9  | 0.9   | 0.9|
| $\alpha = 0.8$      | 0.9      | 0.9  | 0.9   | 0.9|
| $\alpha = 0.9$      | 0.9      | 0.9  | 0.9   | 0.9|

Experiments were conducted when $\alpha$ ranged from 0 to 1 and the interval was set to be 0.1. We found that in most of the cases, the prediction effect is the best when $\alpha = 0.9 - 0.7$, as shown in Table 2.

### 6. Conclusion

We propose Deep Affinity, a novel similarity index for link prediction which represents the affinity between nodes a and b in network. By introducing Deepwalk and K-means, we fully take advantage of network topology information, and the prediction accuracy has increased obviously. In the future, we may try better clustering method to improve results and do some change to make it adapt to practical applications.

### 7. References

[1] Liben-Nowell D, Kleinberg J. The link prediction problem for social networks [J]. Journal of the American Society for Information Science and Technology, 2007, 58 (7):1019 -1031.

[2] Sarukkai R R. Link prediction and path analysis using Markov chains 1[J]. Computer Networks, 2000, 33 (1): 377 - 386.

[3] Zhu J. Using Markov Chains for Structural Link Prediction in Adaptive Web Sites [J]. Lecture Notes in Computer Science, 2001, 2311: 60 -73.

[4] Lü L, Zhou T. Link prediction in complex networks: A survey [J]. Physica A Statistical Mechanics & Its Applications, 2010, 390 (6):1150-1170.

[5] Perozzi B, Alrfou R, Skiena S. DeepWalk: online learning of social representations [J]. 2014:701-710.

[6] Sun Q, Hu R, Yang Z, et al. An improved link prediction algorithm based on degrees and
similarities of nodes[C]/ Ieee/acis, International Conference on Computer and Information Science. IEEE, 2017:13-18.

[7] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [J]. Computer Science, 2013.

[8] Constantinou A, Fenton N. Improving predictive accuracy using Smart-Data: The case of football teams’ evolving performance [J]. Knowledge-Based Systems, 2017.

[9] Pattuelli M C, Miller M, Lange L, et al. Crafting Linked Open Data for Cultural Heritage: Mapping and Curation Tools for the Linked Jazz Project [J]. Code4lib Journal, 2013 (21).

[10] Tanjung D, Saleh W K R, Yulia P S. Graph pattern matching in yeast dataset [J]. Journal of Theoretical & Applied Information Technology, 2016.