# Supervised Learning for Link Prediction Using Similarity Indices

Sergey Korolev[1] and Leonid Zhukov

Higher School of Economics, Moscow, Russia,
sokorolev@edu.hse.ru

**Abstract.** The abstract should summarize the contents of the paper using at least 70 and at most 150 words. It will be set in 9-point font size and be inset 1.0 cm from the right and left margins. There will be two blank lines before and after the Abstract. ...

**Keywords:** network analysis, graph theory, link prediction

## 1 Introduction

Over the last few years the topic of link prediction has become very popular due to the rise of recommendation systems and social networks. The

The aim of this paper is to try to devise an approach to predict the missing links in the network using only structural information of said network.

## 2 Similarity Indices

In this section, we will describe most popular local and global similarity indices and measure their accuracy in ranking links according to the probability that said links were removed from the network.

We'll start with describing the method for measuring said accuracy. We start with the set of edges $E$ of existing graph $G$. Then we shuffle it and split it into $n$ parts $E'_i,\ i \in \{1, \dots, n\}$. Then for each step we subtract $E'_i$ from $E$ and look at the complement of the remaining set of edges $\bar{E}_i$ to the set of edges of complete graph on these nodes, so that $E^* = E \cup \bar{E} = E_i \cup E'_i \cup \bar{E} = E_i \cup \bar{E}_i$, where $E^*$ is the set of edges of complete graph and $\bar{E}$ is complement of graph $G$.

The main tool for measuring the accuracy of ranking is AUC score, which can be described as the probability that for pair of edges $(e_a, e_b)$ $e_a \in E'_i,\ e_b \in \bar{E}$, the ranking is such that $score(e_a) > score(e_b)$ plus 0.5 times the probability that $score(e_a) = score(e_b)$. Or as formula – $AUC = \frac{n' + 0.5n''}{n}$, where $n$ is number of pairs $(e_a, e_b)$ $e_a \in E'_i,\ e_b \in \bar{E}$, $n'$ is number of pairs such that $score(e_a) > score(e_b)$ and $n''$ is number of pairs such that $score(e_a) = score(e_b)$.

We'll first describe all indices used for the prediction of links and then compare their AUC scores on different networks.

### 2.1 Local Similarity Indices

**Common neighbours (CN)** This is the most obvious approach to the idea of capturing the similarity of two nodes in the network. As such, number of indices described below will use this measure with different weights. So if we denote $\Gamma(x)$ the number of neighbours of node $x$ in the graph, the formula looks like

$$s_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)|, \tag{1}$$

where $s_{xy}$ is the index.

**Salton Index (SaI)** This index is also called cosine similarity and is defined as

$$s_{xy}^{Salton} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x \times k_x}}, \tag{2}$$

where $k_x$ is the degree of node $x$.

**Jaccard Index (JI)**

$$s_{xy}^{Jaccard} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \tag{3}$$

**Sørensen Index (SoI)**

$$s_{xy}^{Sørensen} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k_x + k_y}. \tag{4}$$

**Hub Promoted Index (HPI)**

$$s_{xy}^{HPI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{min\{k_x, k_y\}}. \tag{5}$$

**Hub Depressed Index (HDI)**

$$s_{xy}^{HDI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{max\{k_x, k_y\}}. \tag{6}$$

**LeichtHolmeNewman Index (LHN1)**

$$s_{xy}^{LHN1} = \frac{|\Gamma(x) \cap \Gamma(y)|}{k_x \times k_y}. \tag{7}$$

**Preferential Attachment Index (PAI)**

$$s_{xy}^{PA} = k_x \times k_y. \tag{8}$$

**AdamicAdar Index (AAI)**

$$s_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log k_z}. \tag{9}$$

**Resource Allocation Index (RAI)**

$$s_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}. \tag{10}$$

### 2.2 Global Similarity Indices

**Katz Index (KI)**

$$S^{Katz} = (I - \beta A)^{-1} - I. \tag{11}$$

**LeichtHolmeNewman Index (LHN2)**

$$S^{LHN2} = 2m\lambda_1 D^{-1}(I - \frac{\phi A}{\lambda_1})^{-1} D^{-1}. \tag{12}$$

**Average Commute Time (ACT)**

$$S_{xy}^{ACT} = \frac{1}{l_{xx}^{+} + l_{yy}^{+} - 2l_{xy}^{+}}. \tag{13}$$

**Cosine based on $L^+$ (CBL)**

$$s_{xy}^{cos^+} = cos(x, y)^{+} = \frac{v_x^T v_y}{|v_x| \cdot |v_y|} = \frac{l_{xy}^{+}}{\sqrt{l_{xx}^{+} \cdot l_{yy}^{+}}}. \tag{14}$$

**Random Walk with Restart (RWR)**

$$s_{xy}^{RWR} = q_{xy} + q_{yx}. \tag{15}$$

**Matrix Forest Index (MFI)**

$$S = (I + L)^{-1}. \tag{16}$$

**Table 1.** AUC scores of local indices. AN - Word adjacencies network[2], CN - Neural network[3], Dp - Dolphin social network network[4], FB - American College football network[5], LM - Les Miserables network[6], PB - Books about US politics network[7], KC - Zachary's karate club network[8], UA - US air transportation system network[9]. Best results for given network in bold.

| Net | CN | SaI | JI | SoI | HPI | HDI | LHN1 | PAI | AAI | RAI |
|---|---|---|---|---|---|---|---|---|---|---|
| AN | 0.662 | 0.606 | 0.603 | 0.603 | 0.618 | 0.603 | 0.566 | **0.744** | 0.662 | 0.659 |
| CN | 0.844 | 0.797 | 0.790 | 0.790 | 0.805 | 0.779 | 0.725 | 0.750 | 0.861 | **0.866** |
| Dp | 0.779 | 0.774 | 0.779 | 0.779 | 0.763 | 0.780 | 0.762 | 0.619 | **0.781** | **0.781** |
| FB | 0.846 | 0.856 | 0.858 | 0.858 | 0.856 | 0.857 | **0.859** | 0.270 | 0.846 | 0.846 |
| LM | 0.910 | 0.882 | 0.880 | 0.880 | 0.847 | 0.878 | 0.820 | 0.776 | 0.918 | **0.919** |
| PB | 0.887 | 0.884 | 0.875 | 0.875 | 0.894 | 0.863 | 0.848 | 0.653 | **0.897** | 0.890 |
| KC | 0.700 | 0.636 | 0.607 | 0.607 | 0.712 | 0.593 | 0.600 | 0.712 | 0.726 | **0.733** |
| UA | 0.934 | 0.908 | 0.897 | 0.897 | 0.869 | 0.891 | 0.767 | 0.885 | 0.945 | **0.951** |

**Table 2.** AUC scores of global indices.

| Net | KI | LHN2 | ACT | CBL | RWR | MFI |
|---|---|---|---|---|---|---|
| AN | 0.714 | 0.549 | **0.742** | 0.586 | 0.738 | 0.667 |
| CN | 0.852 | 0.717 | 0.738 | 0.848 | 0.725 | **0.865** |
| Dp | 0.799 | **0.825** | 0.760 | 0.791 | 0.649 | 0.804 |
| FB | 0.857 | 0.879 | 0.588 | **0.885** | 0.273 | 0.878 |
| LM | **0.884** | 0.813 | 0.863 | 0.825 | 0.794 | 0.867 |
| PB | 0.891 | 0.858 | 0.729 | 0.891 | 0.618 | **0.899** |
| KC | **0.755** | 0.610 | 0.666 | 0.739 | 0.618 | 0.749 |
| UA | **0.920** | NaN | 0.892 | 0.913 | 0.862 | 0.913 |

## 3 Method

We begin by defining the test dataset to check the performance of the algorithm. We shuffle the set of existing edges and split it into $n$ parts $E'_i,\ i \in \{1, \ldots, n\}$. It is said that $n = 10$ achieves the best complexity-precision tradeoff. Now we can use the set $\bar{E}_i = E'_i \cup \bar{E}$ as test dataset, where if $e \in E'_i$ then $class_e = 1$ and if $e \in \bar{E}$ then $class_e = 0$.

Now we need to construct training dataset to train our classifier on it and then check its performance on the test dataset. We take the remaining set of edges $E_i$, shuffle it and split it into $n$ parts $(E'_i)'_j$. To achieve balanced training dataset with the same amount of edges with class 0 and 1 we split the complement $\bar{E}_i$ into subsets of the same size as $(E'_i)'_j$. Now for each subset $(E'_i)'_j$ we take this subset out of $E_i$, then calculate indices on the complement of the remaining set and mark classes as $class_e$, where if $e \in (E'_i)'_j$ then $class_e = 1$ and if $e \in \bar{E}_i$ then $class_e = 0$. And to get balanced dataset, on each step we take edges from only one subset of $\bar{E}_i$ as examples of class 0.

After $n$ steps of the above process we get the trainig dataset of size $2|E_i|$. Now we can train our classifier on this dataset and then test its performance on the test dataset. After testing SVM with linear kernel, decision trees, k nearest neighbours and random forests it was observed that the best performance is achieved with classification using random forests (as suggested in [1] and also tested in [10]).

## 4 Results

**Table 3.** Scores of the algorithm performance on the class 1 of marked links.

| Net | F1 | Precision | Recall | ROCAUC | Accuracy |
|---|---|---|---|---|---|
| AN | 0.027 ± 0.003 | 0.014 ± 0.002 | 0.579 ± 0.072 | 0.638 ± 0.032 | 0.696 ± 0.023 |
| CN | 0.041 ± 0.001 | 0.021 ± 0.000 | 0.811 ± 0.018 | 0.809 ± 0.008 | 0.808 ± 0.004 |
| Dp | 0.049 ± 0.008 | 0.025 ± 0.005 | 0.635 ± 0.082 | 0.704 ± 0.040 | 0.771 ± 0.035 |
| FB | 0.199 ± 0.025 | 0.116 ± 0.017 | 0.721 ± 0.044 | 0.831 ± 0.019 | 0.939 ± 0.011 |
| LM | 0.163 ± 0.021 | 0.090 ± 0.013 | 0.831 ± 0.068 | 0.875 ± 0.032 | 0.918 ± 0.013 |
| PB | 0.080 ± 0.009 | 0.042 ± 0.005 | 0.780 ± 0.080 | 0.812 ± 0.040 | 0.842 ± 0.012 |
| KC | 0.092 ± 0.026 | 0.049 ± 0.014 | 0.641 ± 0.145 | 0.717 ± 0.082 | 0.790 ± 0.043 |
| UA | 0.083 ± 0.008 | 0.044 ± 0.004 | 0.901 ± 0.024 | 0.910 ± 0.011 | 0.919 ± 0.009 |

## 5 Conclusion

Proposed approach of supervised learning for link prediction showed interesting results on the selected networks. It appears that the algoritm classifies most of the removed links correctly, but also classifies a lot of links that were not present in the network in the first place as the ones that were removed.

## References


1. Cukierski, W., Hamner, B., Yang, B.: Graph-based Features for Supervised Link Prediction. Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA (2011)
2. Newman, M. E. J. Phys. Rev. E 74, 036104 (2006). Available at: http://www-personal.umich.edu/~mejn/netdata/.
3. Watts, D. J., Strogatz, S. H. Nature 393, 440–442 (1998). Available at: http://www-personal.umich.edu/~mejn/netdata/.
4. Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., Dawson, S. M. Behavioral Ecology and Sociobiology 54, 396–405 (2003). Available at: http://www-personal.umich.edu/~mejn/netdata/.
5. Girvan, M., Newman, M. E. J. Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002). Available at: http://www-personal.umich.edu/~mejn/netdata/.
6. Knuth, D. E. The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA (1993). Available at: http://www-personal.umich.edu/~mejn/netdata/.
7. Krebs, V. Datasets. Available at: http://www-personal.umich.edu/~mejn/netdata/.
8. Zachary, W. W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977). Available at: http://www-personal.umich.edu/~mejn/netdata/.
9. Batageli, V., Mrvar, A. Pajek datasets. Available at: http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm.
10. http://github.com/libfun/...