Improving upon Link Prediction using Supervised Learning

Cuevas López Victor
García Carrillo Jennifer Itzel
Garrido Czacki Mario Horacio
Hernandez Rodriguez Alejandro

January 2021

1 The Link Prediction Problem

Link prediction problems are often a point of interest arising from social network analysis. By modeling interactions between groups of people, it is possible to analyze and predict the behavior of individuals within such groups. Due to the influence of distinct social dynamics within populations, social network analysis is a complex problem that draws interest to this day and remains a contemporary research topic.

This work seeks to recreate the methods of *Link Prediction using Supervised Learning* while also trying to improve upon the results presented by its authors. Due to the oftentimes unclear methodology used, recreating the results presented poses a challenge to anyone interested in the problem. We will strive to clear any possible misunderstandings that may cause suboptimal results in the methods presented by the original paper, while adding our current understanding of Machine Learning methodology in order to improve upon the original results.

2 The Dataset

The original paper develops a Link Prediction methodology for two different databases, BIOBASE and DBLP. While the former has more useful information such as keywords that may add more predictive power to the models, we will only center our efforts on the DBLP database due to its ease of access and our interest in the pure topological analysis of graphs.

The DBLP dataset is a coauthorship network for scientific papers. It contains an entry for each publication, and each entry in turn has a title, a list of authors, a publication date and other data pertaining to the type of publication. There are three kinds of publications, *Inproceedings*, *PHD Theses* and *Articles*.

3 Methodology

For the original work 15 years of this dataset were used, from 1990 to 2004. The first 11 years were used as a training set and the last 4 as a test set. Pairs of authors were selected randomly, and the link-prediction problem was posed as a supervised binary classification problem in which the classes predicted were *Collaboration* or *No collaboration*. For each pair of authors, a feature vector was constructed with the following statistics per author:

- **Sum of papers -** The sum of papers published by both authors. This is a measure of prolificity.
- **Sum of neighbors -** The sum of neighbors each author has. This is a measure of popularity of within a social graph.
- **Log secondary neigbors -** The logarithm of the sum of neighbors of neighbors each author has. This measure allows the model to identify relatively poorly connected authors that have collaborated with highly conected authors. Because the number grows exponentially, the logarithm is used.
- **Clustering index sum -** The sum of the clusterig indexes for each author. This reflects the topology surrounding each author.

A detailed summary of all steps taken is presented in the next section.

4 Data Pipeline

4.1 Data Preprocessing

4.1.1 XML Splitting

Before any exploratory work was done on the database, we decided to split the XML into three distinct files, each containing the data pertaining from 1990 to 2000, 2001 to 2002 and 2003 to 2004 respectively. Because of the nature of the algorithms used, we turned each of these into a structured comma separated value file with the relevant data for each publication.

While the dataset contains three types of publications, data analysis on the support for each class led us to only use the **inproceedings** and **article** classes. For each of these we generated an entry containing article id, date of publication and authors. This database has very high quality, so cleaning was kept to a minimum beyond solving issues related to the XML's codification and duplicate entries. Note that the one attribute that seems to present extensional inconsistencies is year, as sometimes it seems to contradict the date attribute. Manual inspection led us to notice that date corresponds to the date when the item was added to the database, and not the publication date. On the other hand, year is the attribute that reflects the original collaboration date, so we decided to use this attribute in order to split the XML file.

4.1.2 Nodes Catalog

Due to the nature of prediction over a closed set of authors, we decided to generate a catalog of the authors present within the first training file (corresponding to the years 1990 to 2000) in order to bound the problem. As to assure that every author had a sufficient amount of data in order to solve the link prediction problem, the catalog was filtered to only contain authors with three or more publications.

4.1.3 Data Filtering and Graph Generation

For each of the csv files generated from the dataset we filtered in order to only include authors within the catalog generated in the previous step. Afterwards, a multigraph was generated in order to represent the collaborations between authors. Each author was assigned a node, and each collaboration (entry in the original database) was split into edges between all the pairs of authors that collaborated.

4.2 Sampling and Feature Generation

Coauthorships are not exclusively a relation between two authors. Due to the multiple collaborations between more than two authors it is possible to extract huge quantities of different samples from the dataset. Because of this, a naive random sampling algorithm is enough to generate a big enough dataset in order to properly capture the behavior of authors within the database.

Because we not working with a densely connected network, choosing a random pair of authors would most likely yield a pair that hasn't collaborated. In order to prevent the generation of an unbalanced dataset, we synthetized equal amounts of samples representing both connected and unconnected authors. Of the 1990 to 2000 dataset, 149,312 samples were generated. In order to test our model's performance, we generated a test set consisting of 38,103 samples from the 2001 to 2002 and 2003 to 2004 datasets.

As explained at the beggining of the document, we generated four features for each pair of authors:

- **Sum of papers -** The sum of papers published by both authors. This is a measure of prolificity.

- **Sum of neighbors** - The sum of neighbors each author has. This is a measure of popularity of within a social graph.
- **Log secondary neigbors** - The logarithm of the sum of neighbors of neighbors each author has. This measure allows the model to identify relatively poorly connected authors that have collaborated with highly conected authors. Because the number grows exponentially, the logarithm is used.
- **Clustering index sum** - The sum of the clusterig indexes for each author. This reflects the topology surrounding each author.

These four features - in addition to the label - are the input for each of our models.

4.3 Models

Before training, we preprocessed both datasets by standarizing the features. We found that this step greatly increases model performance and becomes vital in order to get optimal results.

The following models were trained. 10 Fold Cross-Validation with hyperparameter grid search was used when possible and sklearn optimal parameters are shown when relevant:

- **Decision Tree** - Min. Samples Leaf: 60, Max Features: Sqrt, Max Depth: None, Criterion: Entropy
- **SVM - Linear Kernel** - Dual: False, C: 0.01
- **SVM - RBF Kernel** - Gamma: Auto, Class Weight: None, C: 5.3
- **K-Neighbors** - Weights: Distance, P:1, Metric: Minkowski, N. Neighbors: 48
- **Gaussian Naive Bayes**
- **Multi-Layer Perceptron**
- **RBF Network**
- **Bagging**

The RBF Network was implemented using Keras and Bagging was simply implemented as the label assigned by a simple majority of all the other models.

Classification Model	Accuracy	Precision	Recall	F1 Score	Squared Error
Decision Tree	82.41	82.43	82.41	82.41	0.1758
SVM - Linear Kernel	82.40	82.62	82.4	82.32	0.1759
SVM - RBF Kernel	83.72	83.72	83.72	83.72	0.1627
K-Neighbors	82.14	82.19	82.14	82.15	0.1785
Naive Bayes	76.60	79.00	76.60	75.85	0.2339
Multi-Layer Perceptron	83.75	83.75	83.75	83.10	0.1625
RBF Network	83.08	83.36	83.08	83.09	0.1691
Bagging	83.53	83.53	83.53	83.52	0.1626

Table 1: Table for weighted performance scores.

For the most part, we were able to match the results presented in the original paper. While we were able to get better results in models such as the Multi-Layer Perceptron, RBF Network and Bagging, we were not able to reach the performance of the reference Naive Bayes algorithm. We believe this to be due to a difference in implementation between Sklearn and the systems used in the original work's modeling. As for the other minor differences in performance, we theorize that these could be caused by a difference between the support we used in relation to the original's, and the possibly different ratios of support between train and test sets. There was also no clarification as to whether balanced metrics were used for evaluation. We defaulted to using weighted metrics for precision, recall and F1 score, and this might have been different in the original paper. As to better show our models' performance, we have included a table consisting of positive-class metrics only.

Classification Model	Accuracy	Precision	Recall	F1 Score	Squared Error
Decision Tree	82.59	83.60	83.13	83.68	0.1741
SVM - Linear Kernel	82.41	80.44	87.85	83.97	0.1759
SVM - RBF Kernel	83.72	83.97	86.29	84.62	0.1627
K-Neighbors	82.14	83.90	81.64	82.75	0.1785
Naive Bayes	76.61	71.45	92.32	80.56	0.2339
Multi-Layer Perceptron	84.04	85.76	83.44	84.59	0.1625
RBF Network	83.48	88.81	78.40	83.28	0.1691
Bagging	83.80	84.13	85.20	84.66	0.1626

Table 2: Table for positive class performance scores.

As shown in this positive-class evaluated table, metrics improve significatively in most cases. Even without the evlation metric clarification, our models' performance gets remarkably close to those presented in the original paper. This leads us to believe that the results presented by the authors are valid and effectively achievable.

5 Conclusions

While our results differ slightly from the original work's, we believe that we achieved an acceptable result in the link prediction task when constrained to topological features only.

We believe there is still untapped data within the database that may improve the metrics, such as the use of article titles. This is why now, after understanding the scope of the original work, we will seek to improve upon it with our own methods. While that is certainly a topic for another paper, we conclude by saying that the optimal recreation of the original work was not at all a trivial task, and one must take care to not make careless mistakes which may lead to suboptimal results.