Link Prediction in Citation Network

Network Graph Science Analysis Assignment 2

Report

Kaggle Team: Bang (Score: 0.97351)

Members: Kimya DHADE, Jiaqi SHI, Xiaofeng XU

This report is about predicting link between node pairs in citation network. In the following, we will introduce the features and model we used, and do some comparison among performance of different classifier models.

1. Feature Engineering and Selection

We consider two kinds of features: textual features, which are generated from the similarity on titles, authors or abstract, and graphical features, which shows the connective property of nodes pairs.

1.1 Textual Features

The title, author and abstract provide useful information for link prediction. Before we generated textual features from them, we tokened sentences, removed words frequently appeared but without practical meaning (stop-words) and did stemming.

Overlapping words in titles/abstract/journal Basically, it makes sense that papers in the same academic sub-category are more likely to cite each other. Thus we consider that overlapping words in title and abstract would influence the linkage in citation network. Papers having more overlapping words in the title and abstract have larger linkage probability. Since papers in the same sub-category are always published in several journals in that field. The title of those journals typically contained same keywords so that the more common words in journal means the more likely those journals are about the same academic field. Papers with more common words in journal are more likely in same field and thus have larger probability of linkage.

Temporal distance between two papers We consider that people can note cite papers published later than theirs and people have larger probability to cite papers published lately. Thus we compute the temporal distance between the source paper and target paper. Negative temporal distance means that the source paper and target paper are impossible to link each other, while smaller temporal distance means larger probability to link.

Common authors People like to cite their own papers. Therefore, papers with common authors are more likely to link. The more common authors two papers have, the more likely they would link in the future.

Cosine similarity of title/authors/journal /abstract Besides, we also consider the cosine similarity between the title, authors, journal and abstract of source paper and those of target paper.

1.2 Graph-theoretical Features

There are some basic link predicting methods, such as methods based on node neighborhoods, methods based on the ensemble of all paths, and some meta-approaches.

Common neighbors /Jaccard similarity coefficient/Adamic Adar similarity/Preferential attachment A number of approaches are based on the idea that two nodes *x* and *y* are more likely to from a link in the future if their sets of neighbors have large overlap. The most direct implementation of this idea for link prediction is to define the number of neighbors *x* and *y* have in common. Thus we computed common neighbors. More advanced, we computed the Jaccard's coefficient and Adamic/Adar similarity, which measure the probability that two nodes have a common feature. Preferential attachment method suggests that a new edge involves node *x* is proportional to the current number of neighbors of *x*. This follows the natural intuition that people are more likely to cite popular papers.

Page rank for source node /target node / Shortest paths We considered that the shorter paths between two nodes, the more likely they form a link in the features, so we computed the shortest path between two papers. Since shortest paths contain *inf* value, we categorized the shortest paths into 4 categories: shortest paths lower than 3 ; shortest paths within set [3, 5], shortest paths within set [5,10] and shortest paths larger than 10 and use 1 to represent the existence of each category. However, we found that we can not compute the shortest paths for test set because all the shortest path in test set is infinite. So we dropped this feature. We also tried to adapt PageRank for link prediction[1].

There were huge gap in value between different features. In order to reduce the weight of large-value features, we scale all of the features.

1.3 Feature Selection

At the very beginning, we calculated the features importance of RandomForest Classifier. The results are as below (**Figure.1**). The feature importance graph provides us a benchmark to select proper features. Practically, the importance score didn't mean too much. We used a relative stupid way to find the best feature combination, which was sticking to a model and removing a feature each time to see the change in accuracy. We found that page rank features and Jaccard Similarity did not help much, so we removed these features.

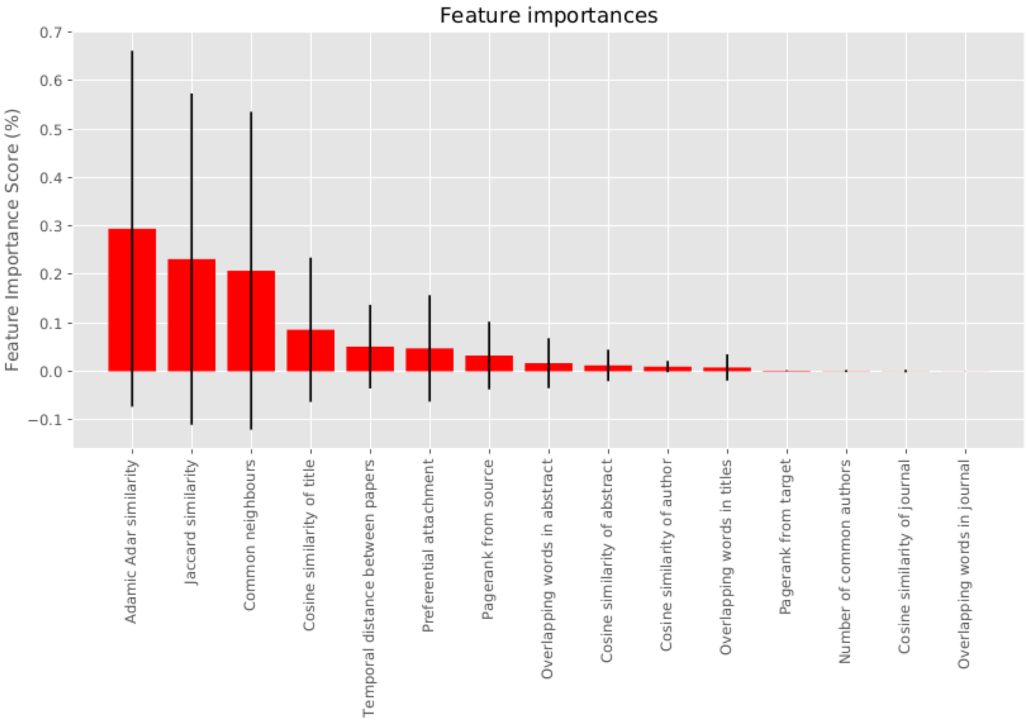

Figure 1. Feature Importance

2. Model tuning and comparison

2.1 Accuracy and Loss comparison

We randomly selected 80% of the train set for training and the rest of 20% for developing test. Ten different classifier models were implemented on the sampled training set and development set. We computed and plot the f1-Score (**Figure.2**) and loss (**Figure.3**) of each model. From the graph we could see that the f1-score of different models are very close, expect for the Linear Discriminant Analysis model. But the loss are quiet different.

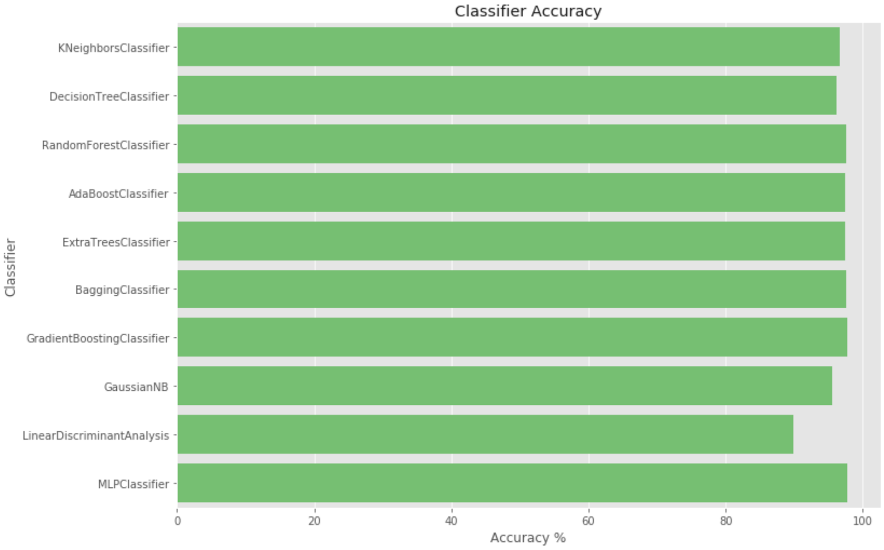

Figure 2. F1-Score of different classifiers

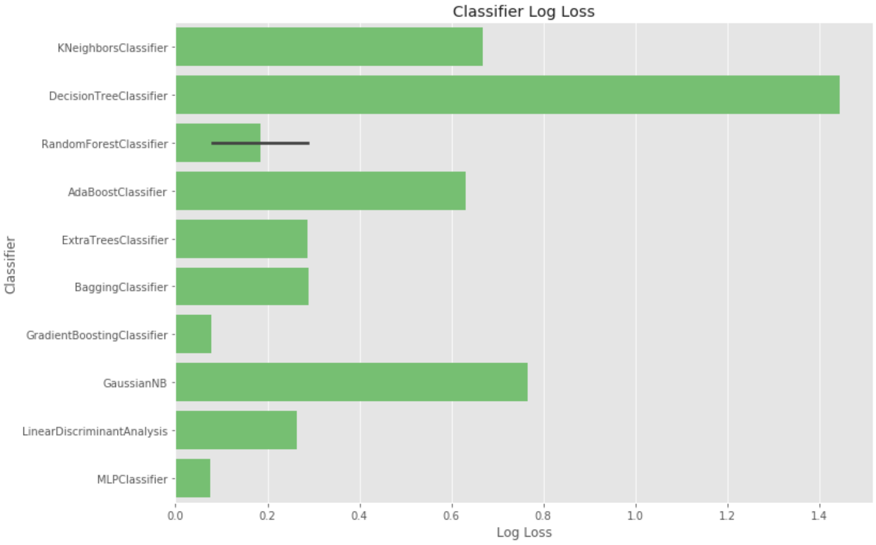

Figure 3. Loss of different classifiers

Both considering the accuracy and loss, we chose it as our final model. It achieved 97.351% accuracy rate in Kaggle competition.

REFERENCE

[1] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. 58, 7 (May 2007), 1019-1031. DOI=http://dx.doi.org/10.1002/asi.v58:7