INF554 - Machine Learning 2018: Predicting missing links in citation network

Team Remontada
AYMEN AYADI, aymen.ayadi@polytechnique.edu
Mohamed Bayrem Kaabachi, bayrem.kaabachi@gmail.com
Private Leaderbord Score : 0.97059

January 2019

1 Overview

Edges have been deleted at random from a citation network. Our mission is to accurately reconstruct the initial network. We define a citation network as a graph where nodes are research papers and there is an edge between two nodes if one of the two papers cite the other.

2 Data Description

- Training set : 615,512 labeled node pairs (1 if there is an edge between the two nodes, 0 else). One pair and label per row, as: source node ID, target node ID, and 1 or 0. The IDs match the papers in the node information file (see below).
- Testing set : 32,648 node pairs. One pair per row, as: source node ID, target node ID.
- Node information : for each paper out of 27,770, contains the paper (1) unique ID, (2) publication year (between 1993 and 2003), (3) title, (4) authors, (5) name of journal (not available for all papers), and (6) abstract. Abstracts are already in lowercase, common English stopwords have been removed, and punctuation marks have been removed except for intra-word dashes.

3 Feature Selection

We have defined 12 features, as follows:

- overlap_title: number of overlapping words in title, represents the number of common words between source and target.
- temp_diff: Difference in publication year.
- comm_auth: The number of common authors, represents the number of common authors between source and target. This number is expected to have a similar effect to that of self-citation.
- comm_journal: Is published in same journal. This value is whether the two papers are published in the same journal. The value is 1 if the same, and 0 otherwise. As researchers tend to work in specific scientific communities, two papers published in the same journal are likely to be connected.
- comm_neighbours: The number of common neighbours, defined as the number of nodes that is connected to both source and target. We hypothesize, as we observe in the social networks in our life, that the more common the neighbours are, the more likely a citation exists.
- same_author: Is self citation. This value represents whether the two papers are written by at least one author. Self-citations are often observed because researchers are more familiar with their own research than the research of others. The value is 1 if self-cited, and 0 otherwise.
- jaccar: Link-based Jaccard coefficient, defined as the size of the intersection divided by the size of the union of the neighbours of from and to.

- cos_similarity: similarity measure for abstracts. In the calculation of similarity measures, nouns and nominal phrases are extracted from each abstract by linguistic filtering to represent a paper as a tfidf vector. We hypothesize that the more similar the topics of two papers are, the more likely that a citation exists.
- to_link: Represents the number of times the target is cited.
- cocitation: Two papers are cocited if there is another vertex citing both of them. Cocitation simply counts how many times two vertices are cocited.
- eccentricity: All papers represent a vertex in a graph. The eccentricity of a vertex is calculated by measuring the shortest distance from (or to) the vertex, to (or from) all other vertices in the graph, and taking the maximum. We calculate the difference between the eccentricity of 2 papers to present a relational value.
- in_link_diff: It is the difference in the number of in links, i.e the difference between the times a paper is cited and the times it cites other papers. We calculate the difference (in_link_diff(source) - in_link_diff(target))to present a relational value between both papers.

We plotted a correlation matrix showing correlation coefficients between the features. This allowed us to see which pairs have significant correlations.

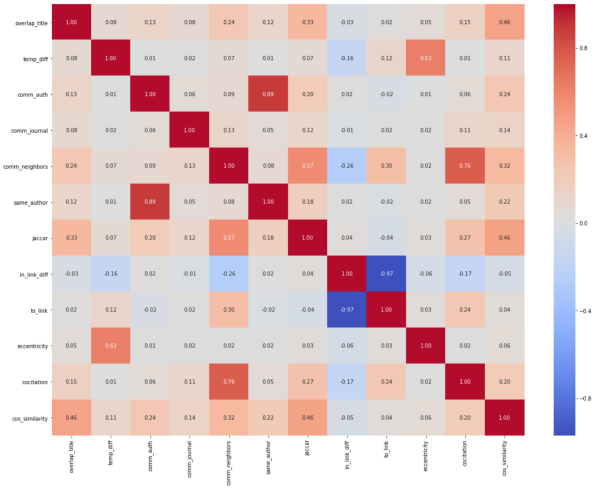

Figure 1: Cross matrix of features.

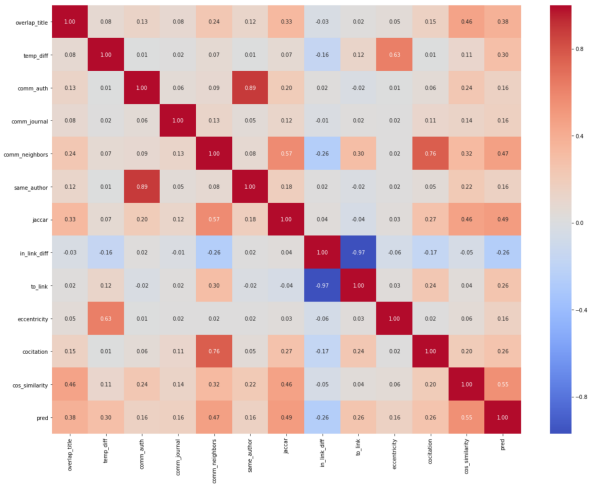

Figure 2: Cross matrix of features and predictions.

Adding the prediction as an element of the correlation matrix, we get the following information and make the following assumptions: comm_neighbors, jaccar and cos_similarity are the most important and impacting features on prediction.

4 Preprocessing

Considering the problem, we found that transforming the data into a graph would be useful. The graph would be populated by nodes ids as vertices, and there would be an edge between two vertices if one cites the other. The creation of the graph allows us to select topological features.
We dealt with the missing values in journal and author on the go while extracting our features by considering that if a paper has no author, its intersection with other papers in terms of author are 0. (and the same goes for journals).
We created a tfidf vector for each abstract in our paper, and we put all those in a matrix in order of IDs.

5 Choosing our model

We implemented a 5-fold cross validation while using the f1 score as a metric to test different classifiers(XGBoost, Logistic Regression, Random Trees). Below the cross validation scores diagram illustrating f1 scores of our tested classifiers.

As a result we found that XGBoost was the best performing one out of those three.

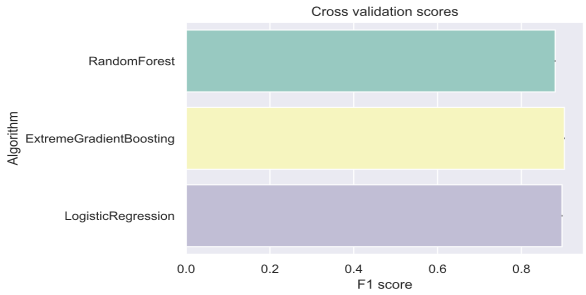

Figure 3: 5-fold cross validation scores.

5.1 XGBoost model

XGBoost (eXtreme Gradient Boosting) is an advanced implementation of gradient boosting algorithm. It has many advantages:

- Regularization, that helps to reduce overfitting.
- Parallel Processing
- Tree pruning: XGBoost make splits upto the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

Before running XGBoost, we set three types of parameters: general parameters, booster parameters and task parameters.

- General parameters relate to which booster we are using to do boosting, commonly tree or linear model.
- Booster parameters depend on which booster we have chosen.
- Learning task parameters decide on the learning scenario. For example, regression tasks may use different parameters with ranking tasks.

We do so by performing a gridsearch on parameters, with f1 score as a metric and choosing the best scoring ones gradually, we sometimes do the search on a pair of parameters if both are correlated.

```
param_test7 = {
 'reg_alpha':[0, 0.001, 0.005, 0.01, 0.05]
}
gsearch7 = GridSearchCV(estimator = gsearch4.best_estimator_,
param_grid = param_test7, scoring='f1',n_jobs=4,iid=False, cv=5)
gsearch7.fit(train[predictors],train[target])
gsearch7.grid_scores_, gsearch7.best_params_, gsearch7.best_score_
```

```
([mean: 0.90507, std: 0.00015, params: {'reg_alpha': 0},
  mean: 0.90509, std: 0.00015, params: {'reg_alpha': 0.001},
  mean: 0.90509, std: 0.00016, params: {'reg_alpha': 0.005},
  mean: 0.90507, std: 0.00015, params: {'reg_alpha': 0.01},
  mean: 0.90512, std: 0.00013, params: {'reg_alpha': 0.05}],
 {'reg_alpha': 0.05},
 0.90511622397162639)
```

Figure 4: Tuning example on reg_alpha parameter

5.2 Shapley values: Features impact

To get an overview of which features are most important for XGBoost model, we plotted the SHAP values of every feature. The plot below sorts features by the sum of SHAP value magnitudes over all sets, and uses SHAP values to show the distribution of the impacts each feature has on the prediction. We choose shapley values to visualize feature importance instead of XGBoost feature importance method because shapley values are more consistent,locally accurate and do not depend on feature order.[1] [1]

As we predicted, comm_neighbors is the most important feature and has most important impact on prediction.

[1]SHAP Values (an acronym from Shapley Additive exPlanations) break down a prediction to show the impact of each feature. They interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value.

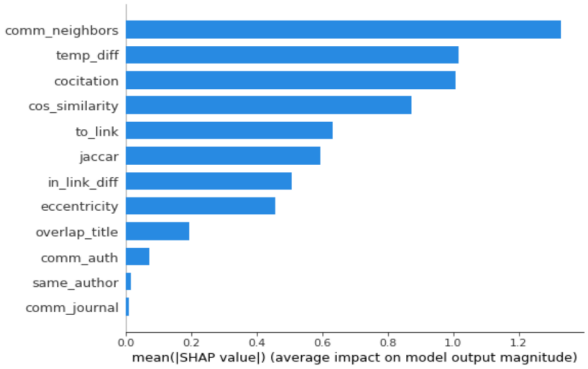

Figure 5: distribution of the impact each feature has on the model

5.3 Results

Our submission scored 0.97059 on Kaggle private leaderboard and 0.96710 on the private leaderboard using the F1 score as a metric. According to our own scoring on the test data we split from the training set, precision outperformed recall which seemed to bring the F1 score down.

6 Discussion

We also consider using SVC, but the large training set proved to be difficult to deal with since its runtime is in $O(n^3)$ according the the number of samples, to circumvent that we took 10 times 10% of our training set randomly, fit our model according to it and built a new classifier that takes the mean of the prediction probability of the others, but the score still didn't beat XGBoost.

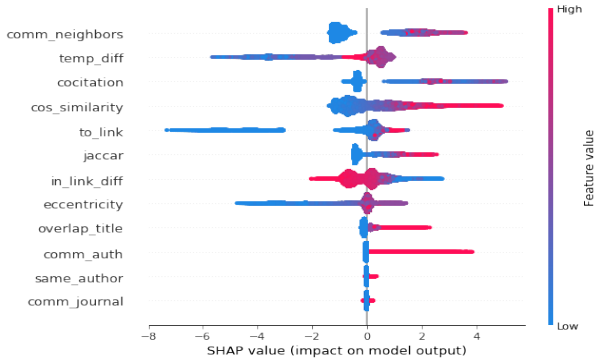

Figure 6: distribution of the impact each feature has on the model

In our model, especially 4 features largely affected the prediction of citations: *common neighbors, temporal difference, cocitation* and *cos similarity*. We can see that most papers cite locally so those who have the most common neighbors are more likely to cite each other. Also we can see from Figure 6 that we are more likely to predict 1 if temporal difference is close to 0 on the positive side, i.e papers are more likely to cite other papers that are recent in their fields.
Cos similarity of tfidf vectors also affected the prediction positively when it went up, which means when the abstracts are based on the same subjects it is more likely that 2 papers cite each other.
eccentricity : the blue lane in Figure 6 indicates that when the target eccentricity is high(thus the value of the feature is low) we are more likely to predict that the papers don't cite each other, which is consistent.
It is also worth noting that there is strong colinearity between common-neighbors and jaccard coefficient, so

References

[1] Scott M. Lundberg, Su-In Lee *Consistent feature attribution for tree ensembles*,