Report - Advanced graph and text learning

Fabrice ZAPFACK, Sylvain PUYRAVAUD, Hugo MARIVAL

April 3, 2016

Contents

1 Part 1 : Feature design

The **node_information.csv** document gives us valuable information about the papers that can certainly help us predict citation links between two papers. However, the information given in this document, for example the article abstract it title, is raw data that can't be used in learning algorithms without processing. Therefore we had to use text mining methods to extract relevant information from **node_information.csv**. Moreover, **training_set.txt** and **node_information.csv** allow us to construct the graph of citations for the papers and for the authors - paper A (resp. author A) is linked to paper B (resp. author B) if A has cited B. Features can then be extracted from these graphs. Some of the features computed were straight-forward but we got the idea of computing others from some papers we read especially Paper 1 and Paper 2.
We have computed many features and decided to keep 15 of them. Those ones can be grouped in 4 categories:

1.1 Document similarity / Topic modelling

- **comwordintitle** : number of common words in the titles of two papers.
- **titles_dist** : cosine similarity between titles using Word2Vec (we averaged over all the word vectors of the title).
- **abstract_dist** : cosine similarity between two abstracts using Word2Vect (we averaged over all the word vectors of the abstract).
- **numofhightfidfwordsincom** : number of words with a high tfidf score (>1.7) in common in the two papers' abstracts.

It is important to stress out the idea that those features were important to determine if the documents were about the same topics. Others appraoches were tested for topics modelling (LDA, LSI form gensim) but they weren't helpful for the classification.

1.2 Graph features

- **dfs** : length of the shortest path between two nodes (papers).
- **same_conn_comp** : 1 if the two papers are in the same connected component of the paper citation graph, 0 otherwise.
- **in_degree** : number of nodes pointing to the given node.
- **out_degree** : number of nodes to which a given node is pointing.

1.3 Paper/author autority

- **paper_pagerank** : pagerank score of the papers in the paper citation graph.
- **author_authorities** : authority scores of the authors in the authors graph using HITS.

- **author_hubs** : hubness score of authors in the author graph using HITS.
- **author_pagerank** : author pagerank scores in the author graph.

1.4 Others features

- **num_com_aut** : number of common authors.
- **tempdiff** : source publication date - target publication date (can be <0).
- **samejournal** : 1 if both documents were published by the same journal, 0 otherwise.
- **nbdefoisautcitejournarttarget** : number of times the author of the source paper has cited an article in the same journal that published the target paper.

2 Part 2 : Learning algorithms

2.1 Data preprocessing

The 15 features computed were store in a numpy array. There were no missing values. Scaling (min/max) and normalization were discarded after we found they were decreasing the classifiaction performance

2.2 Models Selection

In this phase the entire training dataset was used to compare 5 classification models. That comparaison was based on the f1-score obtanained by cross-validation (5 stratified k-folks with test size= 30%). The best parameters for each model was obtained using a gridsearch (due to computation time, only few values were tested). The best score obtained by each model are presented in the table below:

	Linear SVM	Random Forest	Scikit Gradient boosting	XGboost Gradient boosting
F1-score	0.9639	0.9747	0.9752	0.9749

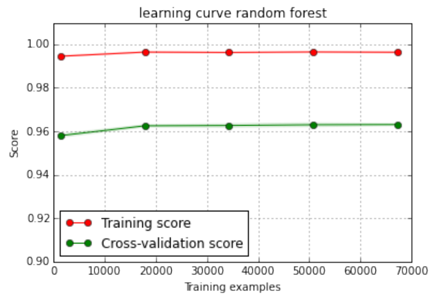
We can notice that there is no algorithm that clearly outperform the other ones. From that situation, we thought it could be a good idea to do the prediction by aggregating the predictions of those 4 algorithms.

2.3 Models ensembling

The method we chose was a weighted majority vote. The weight of each model is obtained for it's f1-score on an independant dataset (balanced). In order to give more credits to good predictors, an exponential function was applyied: $coef(i) = \frac{exp(score(i)*100)}{\sum_i coef(i)}$ This leaded to a sligth improve of the score of the best model between the 4 listed above.

2.4 Training dataset size

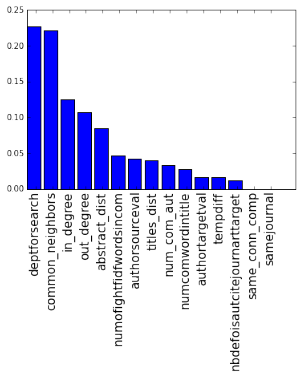
During this project, one of the major difficulty we faced was the time needed to compute the features for the entire training dataset. We then asked ourselves if it was possible to achieve good results using only a subset of the training dataset. For that, we plot the learning curve of the different algorithm. We realised that, after around 30000 traing samples the increase of the score was almost insignificant.



That's why in the final the final file (start.py) the parameter size is use to select the number of training of samples (we recommend 100.000). A dataset containg 50000 samples with label=1 will be computed. after the features are extracted, half of the dataset will be used to train the models, and the other to compute the validation set needed to calculate the coefficients for aggregating the models.

2.5 Features Selection

Another important part of our analysis was the features selection step. After we computed the 15 final features, a greed approach would have been to compared the prediction scores for the $2^1$5 combinaisons of features possible. However it was impossble to compare those $2^1$5 models.
Another way was to use a bottom down approach : Start with all the features and at each step, check if the score improve if we remove one feature. We were happy to see that none removal made the score increase, which means that all features play a role in the decision function.
We also decided to see the importance given by the random forest to each feature. We were surprise to see that some features had very low importance.


deptforsearch
common_neighbors
in_degree
out_degree
abstract_dist
numofightfidfwordsincom
authorsourceval
titles_dist
num_com_aut
numcomwordintitle
authortargetval
tempdiff
nbdefoisautcitejournaltarget
same_conn_comp
samejournal