Advanced Learning for Text and Graph Data
Project report

Maha ELBAYAD
(Kaggle team : ELBAYAD)
Ecole Normale Superieure de Cachan - Master MVA

maha.elbayad@student.ecp.fr

1 Introduction

This project aims at predicting missing links on a citation network where edges have been deleted at random. The citation network is set to be a graph where nodes are research papers and an edge (u, v) exists if article u has cited article v.

2 Outline

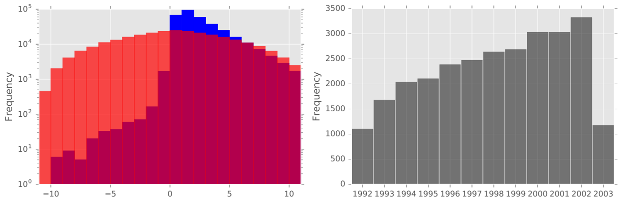
We distinguish three different phases, first the pre-processing and feature engineering where we design features based on meta-data provided on each article of the network as well as textual information extracted mainly from the articles' abstracts and topological information from the main citation network or other auxiliary graphs introduced below. The second phase is learning a classifier on these engineered features to predict whether an edge exists or not. We will compare different classifiers and features selection techniques to boost the classification performance evaluated by the F1-score.

3 The datasets

Our articles corpus consists of 27770 articles where we're provided with the paper title, its publishing year, its authors with their affiliations, the publishing journal and the abstract/review of the article. We format the author name in the following format:

```
Allen C. Hirshfeld (University of Dortmund)
author name : ['a.c.hirshfeld']
affiliation : ['univ dortmund']
```

The training set is of 615512 pairs of articles of which x are citations link. The articles in our corpus are published between 1992 and 2003 (cf. figure 1b) and are published in different journals although a considerable number of journals are missing in the nodes information due mainly to the errors in parsing the `arXiv` meta-data, some of which were manually corrected (cf. figure 2)



(a) *Temporal diff* **(b)** *Articles publication year*

Figure 1: *Articles publication years and the temporal difference between the citing and cited articles (cf. attribute features)*

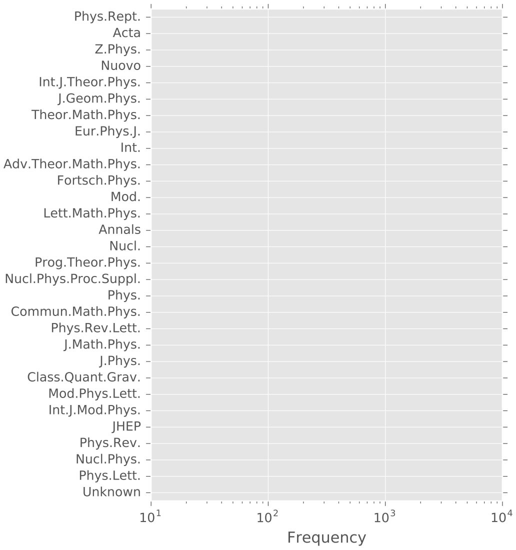


Figure 2: *Top 30 publishing journals in the corpus*

4 Features engineering

We consider the features chosen in Shibata et.al [2012], there are mainly three different groups of features:

4.1 Attribute features:

Extracted from the nodes information

- `(title overlap)` the titles overlap as the number of common words between the titles of the citing and cited papers.
- `(temporal diff)` the difference in publication years. In fact researchers tend to cite recent papers with state of the art techniques to make relevant contributions (cf. figure 1a). Although our graph is supposed to have edges from the citing to the cited articles which contradicts the fact that we have negative temporal difference, flipping those edges lowers the model performance.
- `(self citation)` is it a self citation i.e. is there a common author between the two articles. It's likely for an author to mention her/his previous work to show the continuity of the research.
- `(same journal)` are both articles published in the same journal.
- `(common authors)` the number of the articles' common authors.

- (same affiliation) Do authors have the same affiliation. To accurately compare the affiliations we map them to a features space via the tf-idf vectorizer or the word2vec deep learning model (cf. semantic features).

4.2 Topolgical features:

The main graph on which we predict the missing links is **G** whose nodes are the articles with edges from the citing to the cited article. We also considered two other graphs **GAA** with the authors as nodes ans links between co-authoring authors and **GAC** with edges from the citing authors to the cited ones.

- (common neighbors) the number of common neighbors between the two articles u and v in undirected **G** i.e $|\Gamma_u \cap \Gamma_v| = |\Gamma_{uv}|$.
- (dispersion) the dispersion between the two nodes in **G** that characterizes the edge's strength defined as:

$$\text{disp}(u, v) = \sum_{s,t \in \Gamma_{uv}} d(s, t)$$

  where Γ_{uv} is the set of common neighbors between nodes u and v and $d(s, t)$ is the distance between the two nodes.

- (Jaccard coefficient) the Jaccard coefficient defined in the undirected graph version of **G** as:

$$\text{Jacc}(u, v) = \frac{|\Gamma_u \cap \Gamma_v|}{|\Gamma_u \cup \Gamma_v|}$$

- (Adar) the Adamic/Adar ceffcient defined as:

$$\text{Adar}(u, v) = \sum_{z \in \Gamma_{uv}} \frac{1}{\log |\Gamma_z|}$$

  where having low-degree neighbors in common indicates a stronger connectivity. This feature is implemented for both the **G** graph and the **GAA** graph.

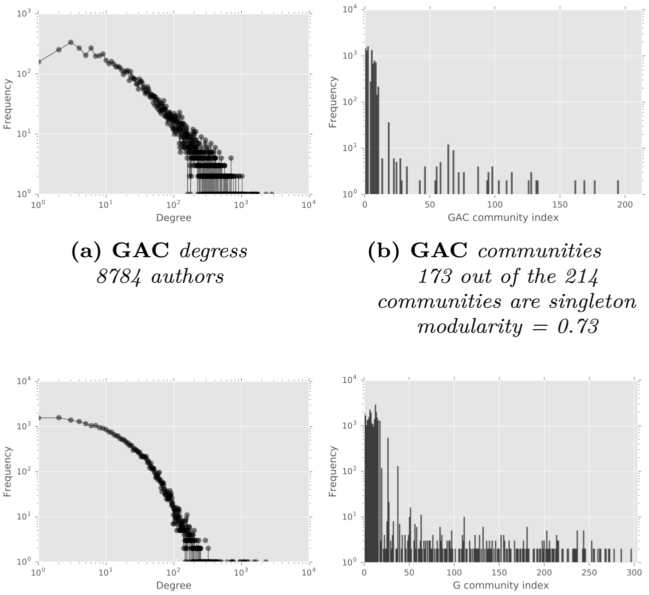
- (diff inlinks) difference of in-degrees between the citing article u and the cited article v:

$$\text{diff}_{\text{in}}(u, v) = \text{in}(v) - \text{in}(u)$$

  where in is the in-degree of a node in **G**.

- (to cited) number of the articles that cited the cited article v i.e $\text{in}(v)$.
- (cited authority) the cited authors ranks on **GAC** defined as the maximum Pagerank of the cited authors. These two last features reflect the fact that famous/successful articles are more likely to be cited and get more successful.
- (authors same co) if either one of the citing and cited authors are in the same community in undirected **GAC**. For the logical reason that a community of researchers in the graph share the same interest in a specific topic.
- (articles same co) if the citing and cited articles are in the same community in undirected **G**.

For the community detection we compute the partition of the graph nodes which maximizes the modularity using the dendrogram generated by the Louvain algorithm (community networkx API)



(a) GAC *degress*
8784 authors

(b) GAC *communities*
173 out of the 214 communities are singleton
modularity = 0.73

(c) G *degress*
27770 articles

(d) G *communities*
123 out of the 305 communities are singleton
modularity = 0.97

Figure 3: *Communities detection in the citations graphs*

4.3 Semantic features:

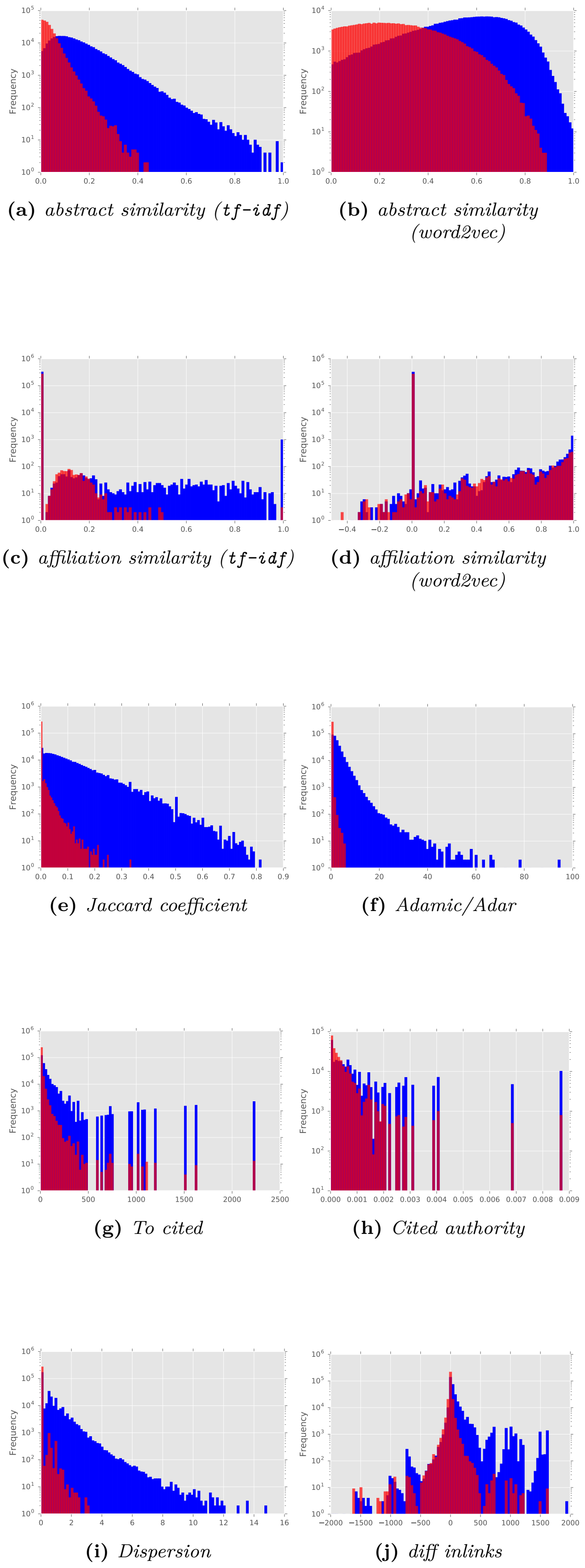
For the semantic features we focused mainly on the abstract, measuring the cosine similarities between vector representations of the abstracts. We experimented with both a Tf-Idf vectorization and a word embedding via the word2vec model.

parameter	description	abstract	affiliation
window	The maximum distance between the current and predicted word within a sentence	15	3
size	The dimensionality of the feature vectors	200	100
min_count	Minimum total frequency to consider a word	20	4
sample	Threshold for configuring which higher-frequency words are randomly down-sampled	1e-3	1e-3

Table 1: *Word2vec models' parameters (Gensim)*

5 Features visualization

To assess the quality of each feature we visualize an approximation of its density on the two different classes (existing citation - nonexistent citation)



(a) *abstract similarity (tf-idf)*

(b) *abstract similarity (word2vec)*

(c) *affiliation similarity (tf-idf)*

(d) *affiliation similarity (word2vec)*

(e) *Jaccard coefficient*

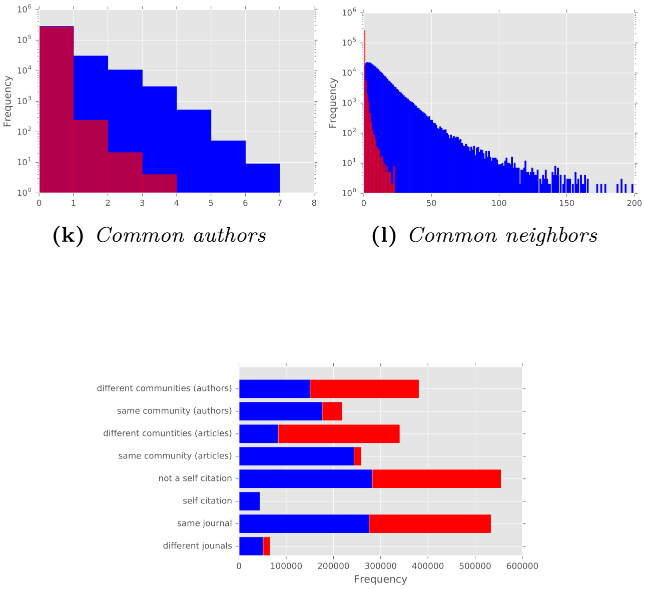
(f) *Adamic/Adar*

(g) *To cited*

(h) *Cited authority*

(i) *Dispersion*

(j) *diff inlinks*



(k) *Common authors*

(l) *Common neighbors*

(m) *Boolean features*

Figure 4: *Features distribution in both classes. In red the negative class and in blue the positive class statistics*

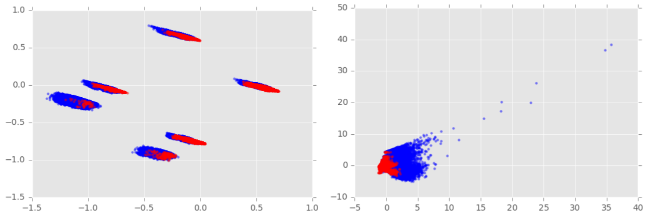
We note that refined coefficients such as the Jaccard, Adamic/Adar and Dispersion allow us to separate the two classes compared to simply counting the nodes difference of in-edges where the two classes heavily overlap. The cited authors authority evaluated by the Pagerank algorithms yields equivalent results to simply counting the times the target article was cited.

6 Classification and Model Selection

For this project we experimented with different machine learning algorithms namely the C-SVM, Neural networks and random forests as well as logistic regression. To tune the model parameters for the SVM, RandomForest and Logistic Regression (from the Scikit learn library) we use the grid search to choose from a set of plausible parameters.

To select the most significant features we scale the design matrix (each feature would range between 0 and 1) and test the following techniques:

- `Recursive feature elimination:` we introduce all the above parameters to the model and then recursively remove the less significant features.
- `Extremely Randomized Trees:` where random subsets of candidate features are tested
- `Factor analysis:` to detect structure in the relationships between features.
- `PCA:` To reduce the dimensionality of the problem.



(a) *PCA first plane* **(b)** *Factor Analysis first plane*

Figure 5: *Dimensionality reduction: Factor analysis yields better combined features adapted to the discrete features*

Feature	rank(1)	importance (2)
dispersion	1	0.025259
common neighbours	1	0.076234
Jaccard coefficient	1	**0.115636**
Adar	1	**0.082322**
Adar authors	1	0.012043
diff inlinks	1	0.024143
to cited	1	0.051572
cited authority	5	0.026481
articles same co	4	**0.284356**
authors same co	7	0.053184
title overlap	1	0.017469
temporal diff	2	0.060224
self citation	1	0.019679
same journal	8	0.005588
common authors	3	0.003748
same affiliation word2vec	6	0.000389
same affiliation tfidf	1	0.000168
abstract tfidf cosine	1	0.064645
abstract word2vec cosine	1	**0.076861**

Table 2: *(1) Recursive feature elimination with linear C-SVM (C=10) & (2) Extremely Randomized Trees: The selected features were as predicted the ones where the classes don't overlap that much. The two variables cited authority and to cited cancel out each other as they reflect the same measure*

As for neural networks, we used `Caffe` BVLC framework to test a network of 3 layers each with [256, 120, 2] neurones and regularize with a dropout layer of tuned ratio and a small weight decay. The loss of the neural network could either be a multinomial logistic loss or a hinge loss with an L1/L2-regularization.
We trained different classifiers on 600,000 edges and kept around 15,511 edges for validation. The performances of each are shown in the tables below.

	precision	recall	f1-score	support
0	0.9549	0.9755	0.9651	7050
1	0.9792	0.9616	0.9703	8461
Avg/total	0.9681	0.9679	0.9679	15511

(a) *kernel SVM (RBF, C=10)*

	precision	recall	f1-score	support
0	0.9596	0.9796	0.9695	7050
1	0.9827	0.9656	0.9741	8461
Avg/total	0.9722	0.9720	0.9720	15511

(b) *Random Forest (100 trees)*

	precision	recall	f1-score	support
0	0.9528	0.9753	0.9639	7050
1	0.9790	0.9597	0.9693	8461
Avg/total	0.9671	0.9668	0.9668	15511

(c) *Neural network (Multinomial loss, dropout = 0.8)*

	precision	recall	f1-score	support
0	0.9467	0.9779	0.9620	7050
1	0.9810	0.9541	0.9674	8461
Avg/total	0.9654	0.9649	0.9650	15511

(d) *Logistic regression (Regularization = 10.L1)*

References

Adamic, L. A., and Adar, E. 2003. Friends and neighbors on the web. *Social networks 25*, 3, 211–230.

Backstrom, L., and Kleinberg, J. 2014. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, ACM, 831–841.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology 25*, 2, 163–177.

Liben-Nowell, D., and Kleinberg, J. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology 58*, 7, 1019–1031.

Shibata, N., Kajikawa, Y., and Sakata, I. 2012. Link prediction in citation networks. *Journal of the American society for information science and technology 63*, 1, 78–85.