INF554 - Team TVRPZ

Romain Fouilland - Philémon Gamet - Jacques Song

École polytechnique

January 11, 2019

Plan

1. Basic features, citations graph and first predictive model
   - First features
   - Graph-based features
   - First predictive model
2. Solving the overfitting issue
3. Final model and parameters tuning

Raw data & data exploration

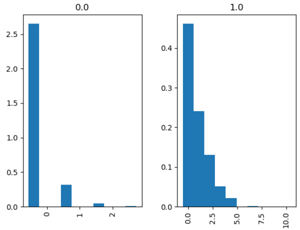


Figure 1: Overlap between titles

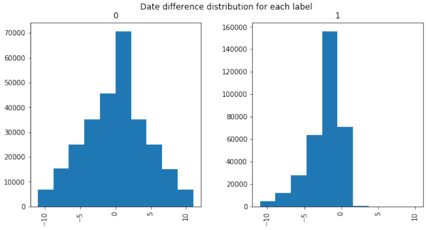


Figure 2: Temporal difference

- Raw data: date, title, authors, journal, abstract
- Features: difference between dates, overlap between titles, authors in common...

Abstract embedding

How to use the abstract?

- Overlapping words between articles
- Embedding: a representation of text in high dimensional vector space

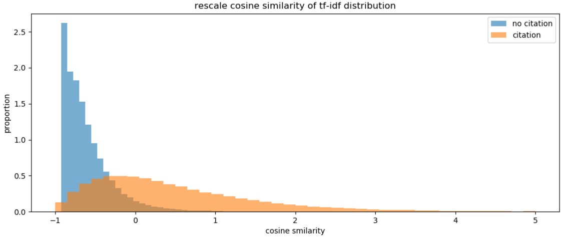
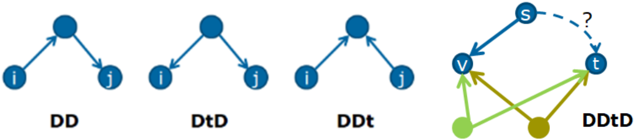


Figure 3: Tf-idf cosine similarity

The graph of documents

Nodes: documents, edges: citations



D the adjacency matrix of the graph:

- DD^T: number of common neighbors
- DD^TD: quoted simultaneously

Compute the degrees: $d_{in}(s), d_{in}(t), d_{out}(s), d_{out}(t)$

First neural network

Score with SVM classifier trained on 5% of the dataset: **96,52 %**
$\rightarrow$ Computed features are meaningful

Multi Layer Perceptron

- 1 hidden layer of 16 perceptrons
- Trained on 5% of the dataset
- Accuracy: **97,01%** on kaggle

Bad results with all the data

To improve predictions: train over a larger part of the dataset

- Good results on our validation dataset
- Very poor results on kaggle (around 85%)...

Plan

Autoencoder

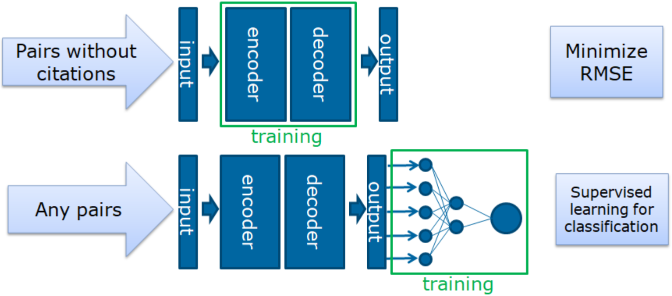


Figure 4: Autoencoder pipeline

- No overfitting thanks to compressed information
- Accuracy: **95,24%** on kaggle

Distributions

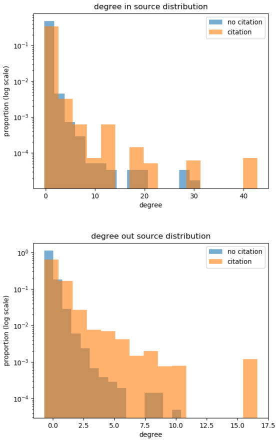


Figure 5: Source degrees

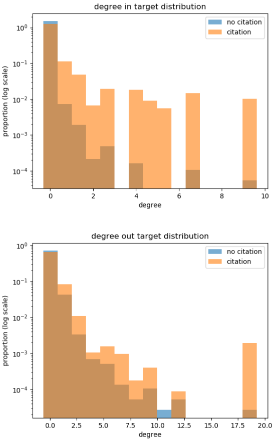


Figure 6: Target degrees

Correlations

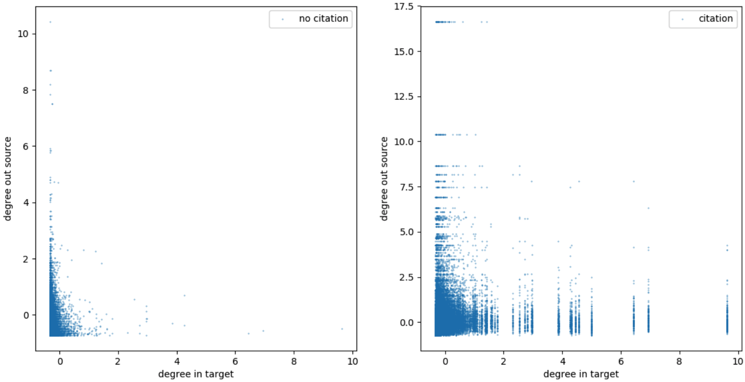


Figure 7: Correlation between source degree out and target degree in

Decision Tree analysis

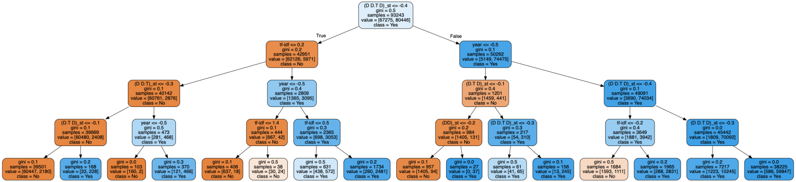


Figure 8: Decision Tree understanding of the data

3 features: DDtD, year, tf-idf and 95.62% accuracy

Random forest results

Random forest on basic features: 77.9%

Random forest on all features:

- 99.7% on validation
- 80.1% on Kaggle

$\Rightarrow$ Is our validation set correct ?

Data splitting issue

Figure 9: Initial data split

The results are already in the training data !

Data splitting issue

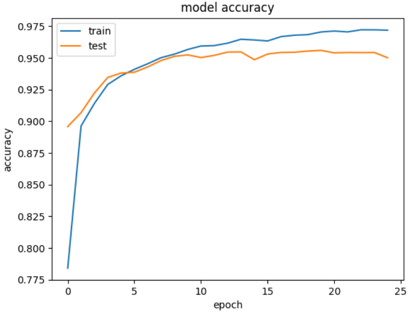


Figure 10: No overfit on few data (5%)

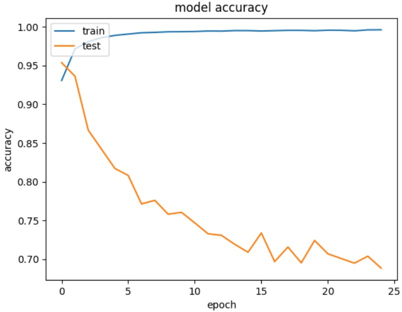


Figure 11: Overfit with more data (70%)

Data splitting issue

Figure 12: Initial data split

The results are already in the training data !

Figure 13: Final data split

It's now safe to learn.

Random Forest analysis

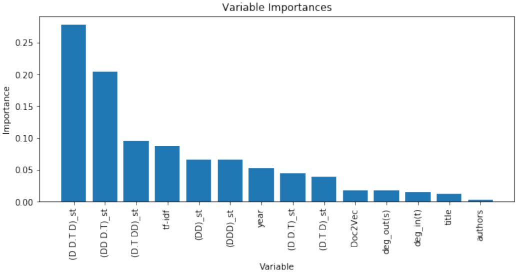


Figure 14: Features importance in a random forest

Well performing forest (96.59% accuracy)
⇒ Add new features: abstract and authors graph

Random Forest analysis

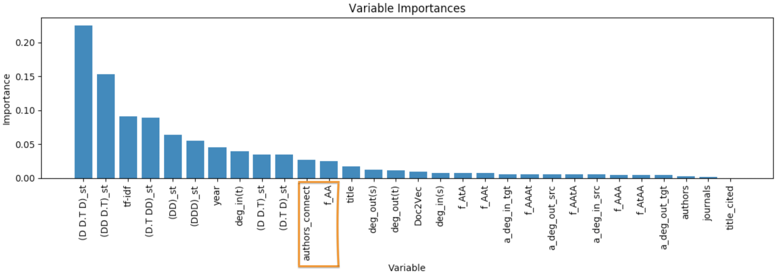


Figure 15: Features importance in our best random forest

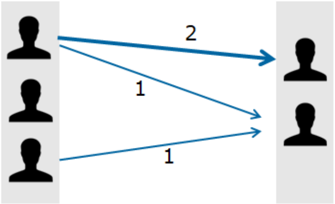
High performing forest (97,11% accuracy)

Plan

The graph of authors

Build a citation network for the authors

- Nodes: authors, edges: number of citations
- Multiple authors for each paper



$$author\_connection = \sum_{s \in S} \sum_{t \in T} A_{st}$$

Convolution Network for abstract embedding

Idea: convert abstract into features maps, train CNN on these maps and collect intermediate layer output to reuse as features in the final model.

Input data:

- Word embedding: word2vec
- Abstract embedding: array of word2vec vectors

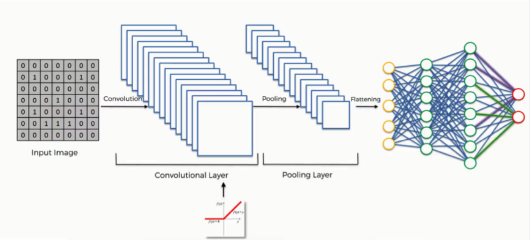


Figure 16: Structure of a CNN

Convolution Network for abstract embedding

Results of the CNN: too much overfitting

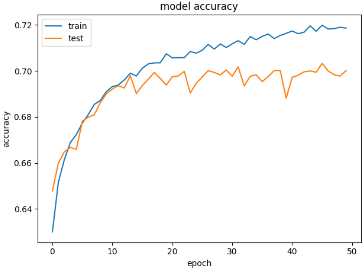


Figure 17: CNN model accuracy showing overfitting

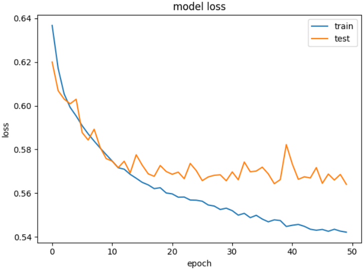


Figure 18: CNN model loss showing overfitting

Overfit vs. Regularization

How to reduce overfitting directly in the neural network ?

- Dropout
- Regularization
- Batch size

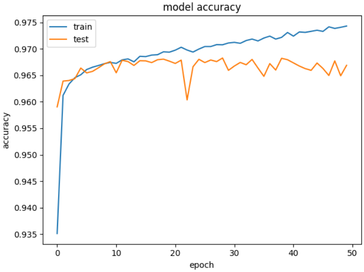


Figure 19: Model accuracy without regularization

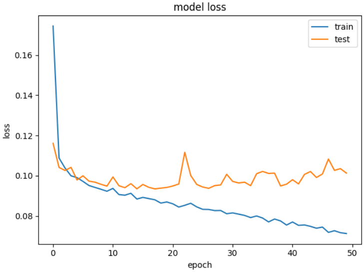


Figure 20: Model loss without regularization

Overfit vs. Regularization

- Dropout: add dropout layers between each dense layer
- Regularization: l_2-regularization
- Batch size: increase size

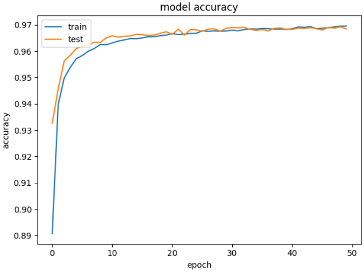


Figure 21: Model accuracy with regularization

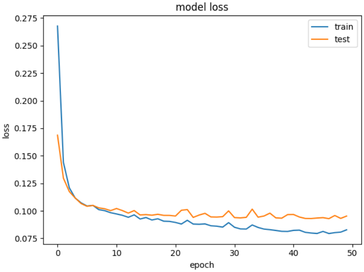


Figure 22: Model loss with regularization

Best results

38	—	ugfdd908	0.97035	7	6d
39	▾16	Pumpkin	0.97035	22	2d
40	▾14	TVRPZ	0.97010	28	3d
41	▴1	MBS	0.96986	23	2d
42	▾2	BDC	0.96961	17	4d
43	▴1	Navy	0.96955	10	2d