Thematic classification of press article
Machine Learning for Natural Language Processing 2022

Hugo Henneuse
ENSAE-IP Paris
hugo.henneuse@ensae.fr

Martin Rouault
ENSAE-IP Paris
martin.rouault@ensae.fr

Abstract

In this project we tackle the problem of thematic classification of documents. For that, we choose a data set that contains articles from the BBC classified in 5 categories. We explore two methods of classification, with very different approaches. A first one, quite usual, with an embedding, and then training a logistic regression or a neural net on the vectorized data. And another one, more original, by looking at our documents corpus as a words-documents bipartite graph and using spectral clustering method as describe in (Dhillon, 2001).We compare some advantages and drawbacks of both methods.

1 Problem Framing

We look at the well-known problem in natural languages processing of document thematic classification. We choose a dataset[1] that contains 2225 articles from the BBC classified in 5 categories : business, entertainment, sport, technology, politics. Many approaches exist to tackle this kind of problems, we propose to explore two of them here. A first one, really classical, with an embedding of our data via Doc2Vec. We then train, a logistic regression and neural net. Une second one, less usual, describe in (Dhillon, 2001), that we summarize and implement in the notebook[2]. The main idea is to look at our documents corpus as a bipartite words-documents graph and applying a spectral method that coming from the community detection in graphs litterature. This elegant method, benefit from a rich connex litterature, it's theoritically well-understood, and so is less a "black box" method than the previous one. Moreover it give, in addition of a documents clustering, a words clustering which is intersting in term of interpretability. Also it doesn't require to split the data in train and test sets.

2 Experiments Protocol

For the first approache, we split the data in two same-sized sets, with the same distribution along the classes, a train set and a test set. we train Doc2Vec embedding on the train set, then the logistic regression on the vectorized data of the train set. We also propose the neural net, trained on the vectorized data of the train set, with the following parameters : 1 input layer (300), 2 hidden layers (1000 and 100), and an output layer (5). We choose a L2 loss, that gives in practice best results for the 0-1 loss we want to look at, even if the L1 loss is the convexification of the 0-1 loss.We choose Adam as solver we batch by groups of 100 articles, we realise 100 optimization iterations.

For the second one, the methode used implies to compute inverses and SVD decompositions of huge matrices, and so require an important RAM capacity. We work under colab, and so we have to restrain ourselves to a two classes problem, by considering only tech and business. We apply the multipartition algorithm proposed in (Dhillon, 2001).

3 Results

Modèle	Erreur
Doc2Vec + Reg. Log.	3,6%
Doc2Vec + NN	2,9%
Cluster. Spectrale	2,6%

Table 1: Scores summary

As said earlier, we choose the 0-1 loss, that we normalized by the number of articles considered, this permit to interpret the results in term of percentage of missclassified. For the first approache,

[1] https://www.kaggle.com/datasets/shivamkushwaha/bbc-full-text-document-classification

[2] https://github.com/BabaUtah/NLP-Rouault-Henneuse

we obtain an error of 3,6 % with the logistic regression on the test set, and we improve it to 2,9% by choosing the class with the highest value given by the prediction of the neural network on the test set.
For the second one, we obtain an error of 2,6 %, on the total of considered data. Warning ! This result isn't really comparable to the previous ones, the considered problem here is restrained.

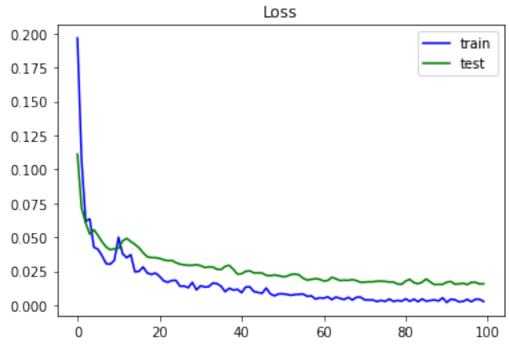

Figure 1: Loss L2 train and test NN

4 Discussion/Conclusion

So, we have explored here, two really different methods, that give both satisfaying results. The first one, however suffering from general problems of deep learning methods : "black box", lack of interpretability, superficial theoretical understanding, necessity to split the data... but efficient on our problem with architecture with low (reasonable) computational cost (spatially and temporally). The second one, far more well understood theoritically, doesn't need to split the dataset, give elements of interpretability, but is more costly (mainly spatially), it will be interesting to compare those methods with more computational power. An other possibility would be to trim the dictonnary (delete usual words, or problematic vertices in the graph), to make adjacency and Laplacian matrices smaller and so make more easily computable the multipartion algorithm, but it may need some more linguistics knowledges or empirical results to be done properly.

References

Inderjit Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.