INF 554 Machine and Deep Learning

Data Challenge: H-index Prediction. Team NLD

Leder Aguirre[1], Denis Mbey Akola[2], and Nicolas Espina[3]

[1]M1 Computer Science-Optimisation, leder-yhivert.aguirre-ramirez@polytechnique.edu.
[2]M1 Cyber-Physical Systems, denis.akola@ip-paris.fr
[3]M1 Applied Mathematics,nicolas.espina-quilodran@polytechnique.edu

December 8, 2021

Abstract

The goal of this data challenge is to predict accuractely the h-index of scientific authors in computer science using Machine Learning techniques. The dataset consist of: (i) a collaboration graph where each node represents an author and edges between two nodes indicates a coauthorship on at least one paper, (ii) a list of the top-cited papers of some of these authors, and (iii) a list of abstracts of some of these papers. 75% of the data set is dedicated to the training phase and the metric to evaluate each model is the Mean Squared Error (MSE). For all the approachs proposed in this project, we construct an author embedding from the abstracts, another from the collaboration network and also structural features of that graph. Our best model achieved a score (MSE) of 49.64101 and ranked 10 on Kaggle private leaderboard. All deliverables for this project are available on Github.

1 Introduction

The problem of predicting the h-index of an author belongs to the class of supervised learning problems. To solve this kind of problems, there is a plethora of models proposed in the literature. We can classify them into two big groups: naive approaches like lasso regression and k-Nearest Neighbours, tree-based like random forest and neural network-based like deep neural networks. Thus,considering this problem, we decided to use both regression approaches and deep neural networks to predict the h-index of the authors. We therefore trained several regression models and a deep neural network. For the regression task, we trained the model using Random Forest, Lasso Regressor, LightGBM, Xgboost, and K-Nearest Neighbors.The deep neural network was based on a recurrent neural network architecture.

2 Data preprocessing and Analysis

The data that was provided for this challenge included the inverted abstracts of papers the authors wrote, the co-authorhsip network graph, and the list of authors and their top cited papers. To a get a good model that works better on this prediction task ,we spent a great amount of time cleaning and preparing this data. First, we converted the abstract from the inverted format to normal text strings without the indexes. Upon further analysis of the abstract, we discovered that not all the abstracts were in english. About 75% of the abstracts provided were in english,however,the remaining abstracts were in other languages (French,Spanish,German,Portuguese etc). In oder to classify the abstracts into other languages, we decided to use a threshold which was computed based on the number of non-english words which were found in a single abstract. A threshold of 0.50 was used in this exercise. It is common to have some words from foreign languages infiltrated into other languages so setting a threshold help us to avoid classifying every abstract that had a foreign word which might have been adopted into English to be a foreign abstract. We leveraged the Google translator python api to convert all these abstracts to English. The graph below in figure 1 shows the distribution of the various languages in the abstracts. We using NLTK library to remove stop-

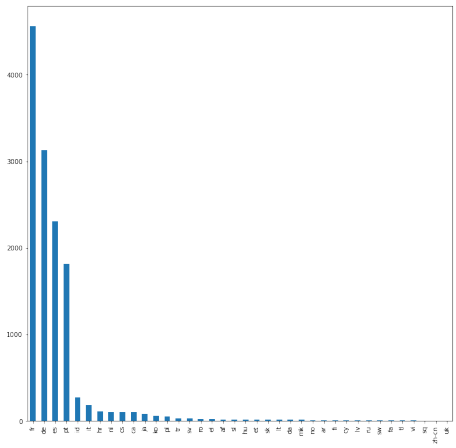

Figure 1: Various languages in abstracts

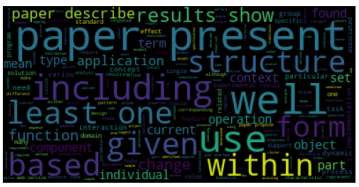

Figure 2: Other common words detected in abstracts

words. Also,we took advantage of WordCloud Python package to visualise the abstracts data and we discovered some more words which were frequent in all abstracts and we likewise treated them as stopwords. A sample of the visualisations with wordcloud is shown in figure 2 above.

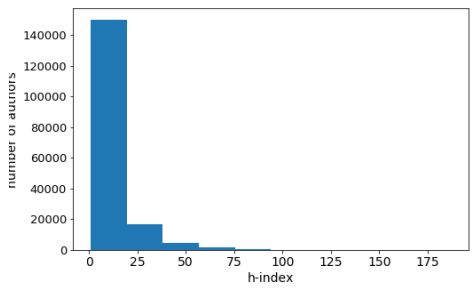

Figure 3: Histogram of the h-index.

3 Data analysis

4 Feature selection and feature extraction

4.1 Abstract Embeddings

In this step,we used Doc2Vec from gensim NLP Python Library to transform our processed abstracts into word embeddings. Doc2Vec is an NLP tool for representing documents (text corpus) as a vector. It is a generalisation of word2Vec.

4.2 features

We decided to use as features the embeddings of abstracts, the embeddings of the graph of collaboration between authors, some gnerated features as

5 Model tuning and comparison

We first try the run all the models with parameters close to the default value. The results are shown in Table 1.

Model regressor	**MSE train**	**MSE test**
KNN with k=7	111.269	119.863
LassoCV	77.026	75.995
FFNN	55.897	63.690
Random Forest	8.869	62.594
XGBoost	46.192	59.648
LightGBM	14.230	51.689

Table 1: Comparison of the models before parameter tuning.

The performance of the first two models gives us a new baseline for the other models in this section. The Feed-Forward Neural Network (FFNN), constructed as a set of sequential layers, don't give us satisfactory results but the execution is fast using the GPU. Finally, since Random Forest had the lowest MSE test compared to the formers, we tested more powerful models derived from Random Forest. So, the improvements with XGBoost and with LightGBM only confirms the theoretical progress on this field of Gradient Boosting model.

6 Discussion

After several experiments,the LightGBM models performs better than the recurrent neural network and the other regressors. Thus,the Light-

Learning_rate	activation_function	units_HL	MSE_val
0.01	nn.LeakyReLU(0.1)	[20,4]	60.0822515
0.1	nn.ReLU()	[20,4]	61.1567066
0.01	nn.Tanh()	[20,4]	64.9793999

Table 2: Best results of parameter tuning for the Feed-Forward Neural Network.

GBM model is best for the prediction task for the h-index prediction.

Conclusions

One of the major setbacks which could account for why our model had a best MSE of 49.641 instead of a lower value could be attributed to the language translation. Though google translator works better in translating words.These are times were trhe context of the sentences is lost in the process of the translation and this error has a multipling effect on other aspects of the model.

References