Statistical Machine Learning Project 2

Chi Zhang 1067750 & Haitong Gao 1068254 & Qianjun Ding 1080391

1 Introduction

The machine learning model(s) developed for authorship attribution to a given document is essential in nowadays society, as it can be applied as a tool for detecting plagiarism or as contribution to the research of privacy protection. In this project, we aim at attributing the prolific author(s) to a given paper with its publication information based on the fitted model with the training set. We will first describe the data structure accompanied by the illustration of the preliminary data analysis. Then, preprocessing steps, including self-implemented methods such as TF-IDF and One-hot Encoding, are presented, followed by the elaboration of selected model(s) based on the data characteristics and the related research. Finally, the evaluation of the model is provided, and the report concludes by choosing the optimal approach(s) with supporting evidence. Besides, alternative approaches which are not optimal for this project are illustrated for reference and comparison.

2 Data Description

Based on the specification, the first 100 authors are prolific authors, which will be potentially assigned to each of the papers. The rest of the authors are treated as co-authors, which is applied as a predictor in constructing the model. In addition to the co-authorship, the year of the publication and venue are also investigated in this project as predictors, which might provide extra information for predicting. Moreover, each instance (paper) has included its title and abstract content as a predictor with different lengths of list recorded in the provided data set, which could allow us to capture specific writing patterns or research domains that could be assigned to the possible prolific author(s).

Based on the provided data and features description, we have collected 25800 papers (instance) for training and 800 papers as the test set. After a preliminary investigation of the data, the issue is that some instances include new information that is not observed from the training set (e.g. co-author X only appears in the test data but not in the training data). That is, the feature is new to the fitted model. In this case, the generic model is required to avoid distortion of the authors' attribution. The pre-encoded 'title' and 'abstract' data also introduce trouble in detecting the redundant words that contribute less to the model's prediction.

3 Data Preprocessing

By manually selecting the prolific and the co-author from the 'Author' feature of each instance in the dataset, the 'Author' feature is split into the response feature (prolific author) and predictor feature (co-author) based on the index criterion, which is mainly considered in constructing the model. The target result contains one or more prolific authors in a specific instance, indicating a multi-label classification task is proceeding. Since the prolific authors are assumed to be equally important, one-hot encoding is applied to convert the predicted result into a 1x100 vector with 1(s) representing a specific prolific author(s) who is involved in writing the paper, while 0 means not involved. A similar strategy can be applied to the predictor features by converting each instance's feature into a vector with respect to the length of the unique integer value list (e.g. co-authors as a 1x21146 vector from values in $\{100, \ldots, 21245\}$) according to the selected model's preference.

The "year" factors from the test dataset are identical, thus it can be excluded from the training set. In addition, since both 'title' and 'abstract' would contain many redundant words, such as "a", "the", "it", we further discard them based on their high frequency of appearance.

4 Baseline Models

First base model we use is the Gaussian Naive Bayes (GNB) classifier. From the project introduction we could recognise that this is a classification problem, so the GNB classifier becomes our first choice. It is one of the useful classification techniques that is suitable for multiclass classification and efficient and easy to implement (Jahromi & Taheri, 2018). However, in this project, the output of the result is in multi-label format, which means it may contain more than one prolific author. In this case, the label powerset in the skmultilearn package will be used to transform the multi-class classifier to solve multi-abels problems. It enables the GNB classifier to train on all unique combinations of labels found in the training data.

Secondly, MLP, as a neural network-based model, could capture the underlying complex relationships in the data. We apply the MLP deep learning model with 20 and 200 as numbers of two hidden layers, using the RELUs function for both inputs to the first hidden layer and the first to the second hidden layer. Notably, sigmoid is used as our final activation function in order to generate multi-class output in a probabilistic manner. Here, any probabilities greater than default threshold 0.5 would append prolific authors to the predicted result. Firstly, only "coauthors" is included as our primary factor since it directly relates to the response variable. Each prolific author is more likely to collaborate with similar people from a particular field. Meanwhile, we also add 'title' or 'abstract' to form new models. They are less important because all information has been encoded into a range of integers with respect to the 'title' and 'abstract' features that contain a sequential logic (e.g. contents of 'abstract' are mapped into an index in 1, . . . , 4999), causing difficulty in capturing sequentiality. After embedding them into useful words, they still produce extremely low training accuracy when using the Long term short memory model (Zaheer et al., 2017).

5 Final Approaches

5.1 MLP - Threshold Controlling Number of -1 Outputs

From the characteristic of data, among all the instances, 18333 of them have no prolific author (nearly 75% of the train data), which means all the authors are coauthors. Such a large number of empty author instances may cause the model to skew more towards a prediction class -1, resulting in a degradation of the model's performance. Due to the large number of "empty author instances" problems, we try various thresholds from 0.01 to 0.5 for finding an optimal one that can produce the best training f1-score by using MLP. According to the plot, the resultant f1-score converges when the threshold equals 0.29. We use the smallest one because our purpose is to minimize the number of '-1' outputs, meaning that the probabilities generated using the sigmoid function must be less likely to be smaller than the threshold, getting as similar to the test samples as possible. This model predicts 371 '-1's, which is less than the original model with 465 '-1's predictions. As a result, it achieves the highest test score with 0.53164. However, obviously, there is an overfitting problem for the MLP model, as seen in Figure 1, applying the model to the "train_test_splitting" data the score will go up to 0.56, which is higher than the score on Kaggle.

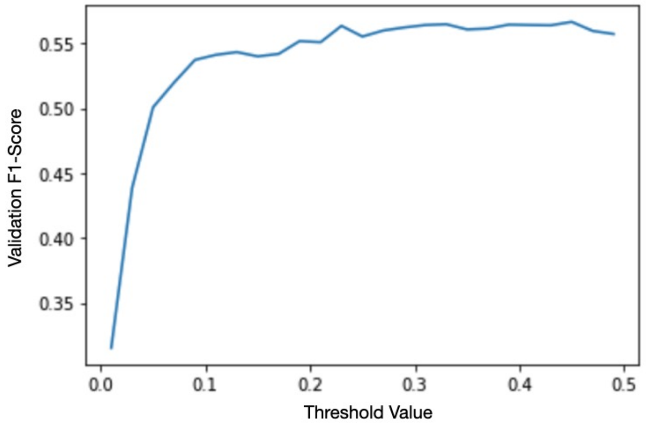

Figure 1: Thresold vs. F1-Score

5.2 GNB - Majority Voting

Due to generating relatively low f1-scores by using our existing models, we aim to combine these weak models together to form a stronger one through the majority voting technique. Here, we randomly split the training set to 67% training and 33% validation sets 100 times and temporarily save these results altogether. This will further reduce overfitting, as it reduces the chance of generating mis-classification by the minority of the models. Subsequently, the "prolific authors" with the highest frequency are treated as our final prediction corresponding to each instance. As a result, GNB will produce the highest test score with 0.54522, which will be considered our final approach. However, this approach is much more computationally expensive due to training models multiple times.

6 Alternative Approaches

6.1 GNB Grid Search & Pipeline

Parameter tuning is applied with the grid search function that we implemented, a 3-fold cross validation is included. In GNB classifier, the var_smoothing parameter controls what is the largest variance among all features that is added to variances for the purpose of calculation stability. In the baseline model this parameter is 1e-9 by default, we first use grid search to experiment in the test set to narrow down the optimal parameter to between 1e-4 and 1, with a length of 15 var_smoothing parameter lists, equal length from 1e-9 to 1e5 as input parameters. Then use a pipeline again to locate the optimal parameter as 0.0015 and obtain a result score 0.53755. However, since grid search only optimise based on the training set, which lead to a potential overfitting issue, as well as the large computational time, we decide not to apply this method as the final approach.

6.2 Random Drop Empty Prolific-Authors Instances

To deal with the large amount of “empty author instances” from the training set, we also designed a “drop empty author instances” strategy to balance the distribution of classes in the training set. In this strategy, parameter ’n’ controls how many instances that will be dropped. We randomly delete multiple (n) instances with no profile authors from the training set to match the test set approximate amount of no prolific author instances. After several experiments by dropping off instance before training, the best model delivers a score as 0.53672. However, removing 50% of the empty prolific author instances will make the model lose too much of the training set, which will result in a drop in the final score.

6.3 Single Label Prediction for Conservatism

Since f1-score is our evaluation index, we try to manually force the outputs to be presented in one dimension as a more conservative approach. For instance, given the true label is [1, 2], returning only [1] or [2] as prediction will result in a higher f1-score than returning a totally mistaken estimation, such as [3,4]. Therefore, before the one-hot encoding process, we manually transfer the multi-label response variable to a single label by only selecting the starting “prolific author” in advance. However, this is a double edge sword as the accuracy will be further reduced if the ground truth contains many “prolific authors”, which can also be shown by the results.

7 Conclusion & Future Direction

In conclusion, after trying various approaches, our final approaches are GNB with majority voting and MLP with threshold tuning. Not only do they produce the best testing f1-score, but also have advantages as described above. The Table 1 & 2 of the test set F1-score illustrate our process in implementing until we reach the final approach. For future studies, potential feature engineering can be applied on “title” and “abstract” variables to observe their sequential property. Therefore, these variables would also be useful for training.

Model	Baseline	Coauthor + Title	Coauthor + Abstract	Majority Voting	Single Label	Majority Voting + Threshold	**MLP Final Approach**
F1-Score	0.50689	0.37620	0.30741	0.47666	0.51141	0.51089	0.53164

Table 1: MLP Models Evaluations

Model	Title	Abstract + Coauthor	Venue + Coauthor	Drop 50% Non Prolific Authors	Grid Search	**GNB Final Approach**
F1-Score	0.09750	0.21916	0.48541	0.53672	0.53755	0.54522

Table 2: GNB Models Evaluations

Reference

[1] Jahromi, A. H., & Taheri, M. (2017). A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features. In *2017 Artificial intelligence and signal processing conference (AISP)* IEEE.

[2] Zaheer, M., Ahmed, A., & Smola, A. J. (2017). Latent LSTM allocation: Joint clustering and non-linear dynamic modeling of sequence data. In *International Conference on Machine Learning.* PMLR.