(Ewen) Yuxuan Wang

Authorship Classification Project Updates

In all my text data from 11 authors combined, there are over 10,000 unique words. Rather than extracting the average length of the sentences and the variety of words used, I found more meaning in extracting the words and relationships between words after reading articles about NLP. Currently, I use a Tfidf vectorizer to vectorize unigrams and bigrams that make up at least 0.1% of the document and a maximum of 10% to filter out words that are either too rare or too common. Although I initially planned to split the data by paragraphs, I neglected the problem that many paragraphs are dialogues that are only one sentence long. Therefore, I decided to more evenly create the samples by separating each sample by 2,000 characters (including spaces and punctuation) ending on the closest full word. The result is a numerical matrix that is 11028 x 18048, where each sample is a vector of the weighted frequency of that word in the 2,000-character document and the label is the author. Then, I randomly split the data into 75% training and 25% testing.

To reduce the enormous number of columns and the sparsity of the matrix, I deviated from my original plan and performed PCA with a parameter to keep 90% of the original variance. The PCA is fitted on the training data (the covariance matrix is calculated from the training), then applied on the training and testing data to ensure that the testing data doesn't influence the result of the PCA. The trade-off of PCA is that there is no longer interpretability of which words or phrases distinguish the authors; however, now the number of columns is reduced to only 4,336. Furthermore, I filtered out the features that have low correlation to the label by selecting the 1,000 features that have the lowest p-values in an ANOVA F-test. Thus, predictors have the most different means grouped by the labels are kept.

After processing the features, I tested the accuracy of four different classifiers with default parameters from scikit-learn. Despite reducing the number columns by 18 folds, training the random forest and SVM classifiers took over 20 minutes each. Therefore, I do not plan to perform cross-validation on the models. Here are the preliminary accuracies:

Logistic Regression: 0.923

Random Forest: 0.860

Gaussian Naïve Bayes: 0.892

Linear SVM: 0.220

Compared to the accuracy of random predictions of 1/11 = 0.09, the first three classifiers performed quite well. SVM performed surprisingly poorly most likely because I did not tune the C parameter. So, moving forward, I will compute the other metrics (F1 score, precision, recall and error) for the models, tune the parameters of each model, and compare the performance of a neural network and AdaBoost with logistic regression base classifiers. Perhaps I will perform manual hyperparameter tuning on the classifiers by visualizing the metrics for different settings. I would also like to compare the results of training a model using the original data (before PCA and filtering) with those of the reduced dimensions and interpret words that help differentiate between authors the most.