**Tim Zhang (110746199)**

**CSE519 Final Report: Dating Documents**

**Overview**

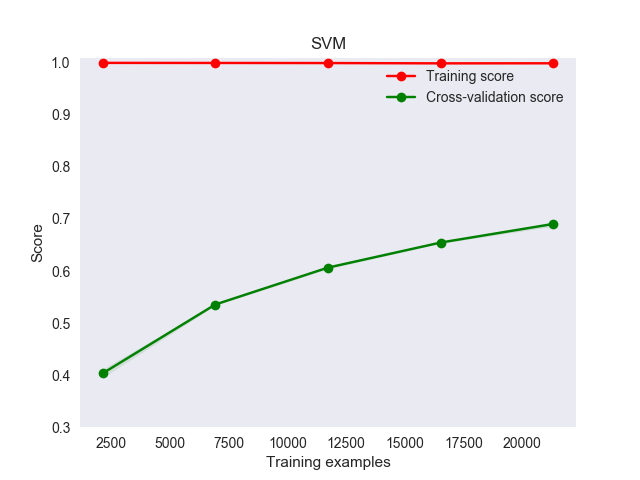
I have accomplished most of the goals which I previously introduced in my Progress Report. This includes extending my corpus, visualizing the model learning space, improving my models using cross validation for hyperparameter tuning, and performing an “apples to apples” comparison against the RNN implementation of Yingtao. In the direct comparison with the RNN model by Yingtao, I found my SVM model to perform favorably in each of the text domains (New York Times, COHA\_500, etc) with almost double the MAE scores in 4/5 domains. Unfortunately, due to time and computation constraints I was unable to experiment with CNN or Gradient Boosted Trees on this domain.

The remainder of this report will detail my process and implementations of: improving the model on COHA, extending the corpus, training a model for the extended corpus, and comparison with the RNN implementation. Progress from the previous report can be seen at:

<https://github.com/moduio/CSE519-2017-110746199/blob/master/Project/project.ipynb> starting from the section titled **“Refined Models”** which is halfway down the document.

**Improving SVM**

Having found very positive results on COHA using SVM I was interested in plotting how well the model was generalizing to the test set. Additionally, I wanted to know if my model would improve from incorporating more sources of data. To this end I implemented learning curves for SVM on COHA, which show the test and cross validation scores as a function of training set size.



We can interpret this plot as indicating that using more training data would lead to better model performance, although the gains may be less pronounced.

I was also interested in seeing how much more predictive power I could squeeze out of SVM. For this I implemented cross validated grid search for hyper parameters. I found that the optimal value of C for SVM resulted in:

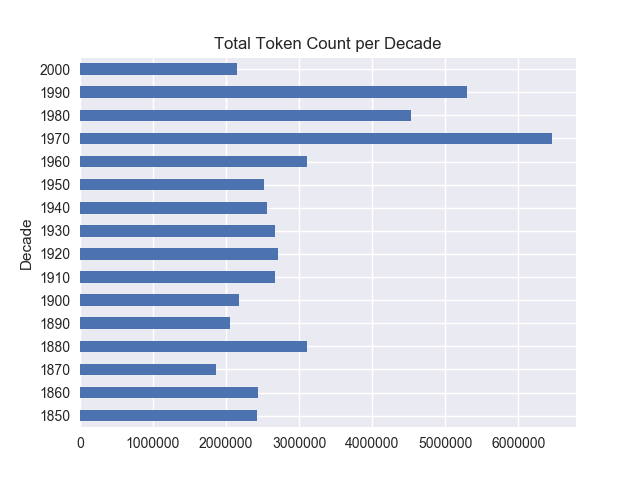
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MAE | Accuracy | Precision | Recall | F1 | Support |
| 6.09 | 0.733 | 0.73 | 0.73 | 0.73 | 8000 |

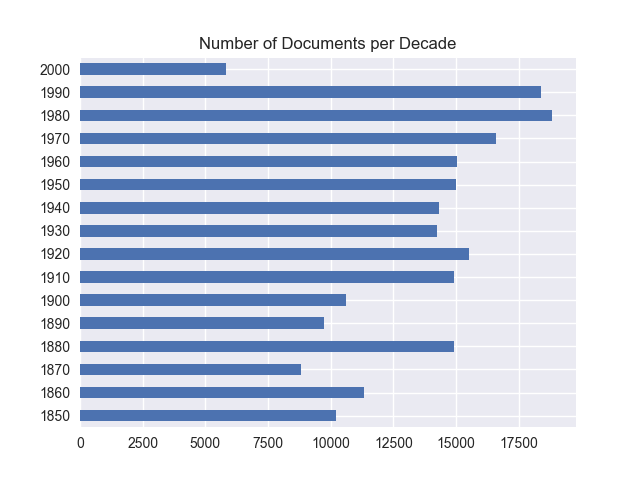
These scores are not significantly better than the performance of SVM using the default settings. While disappointing from my perspective, it does go to show how difficult this problem domain is.

It should be noted that although my Jupyter Notebook file only has a hyperparameter search range of [.001, .01, …, 1.5], I tried many other combinations which I omitted for cleanliness of the notebook. I also attempted cross validated grid search for Random Forest tuning the number of trees, but this approach did not perform better than SVM so I omitted these experiments from my final notebook.

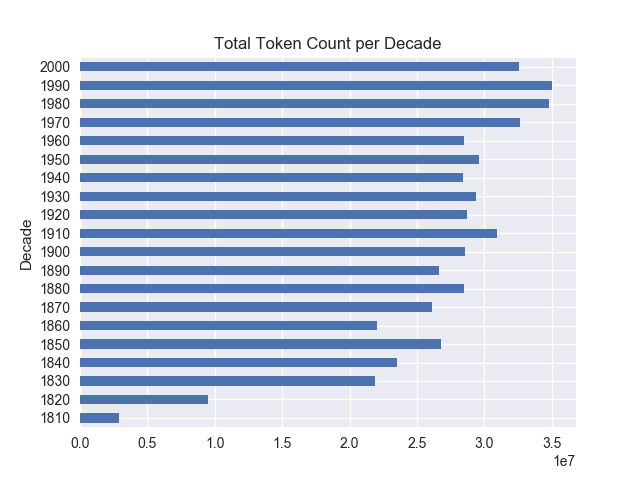
**Extending the Corpus**

Having seen the limits of my previous model along with the promise of more data, I felt confident in incorporating the New York Times dataset which I had painfully acquired via custom API scraping. This data included article snippets and summaries from each year available in their database from the 1850s – 2000s. To avoid some issues of noise, I only considered those “documents” which were at longer than 10 words.



From the above plot we can see that the token frequency distribution is relatively balanced except in the 1970-1990 range.

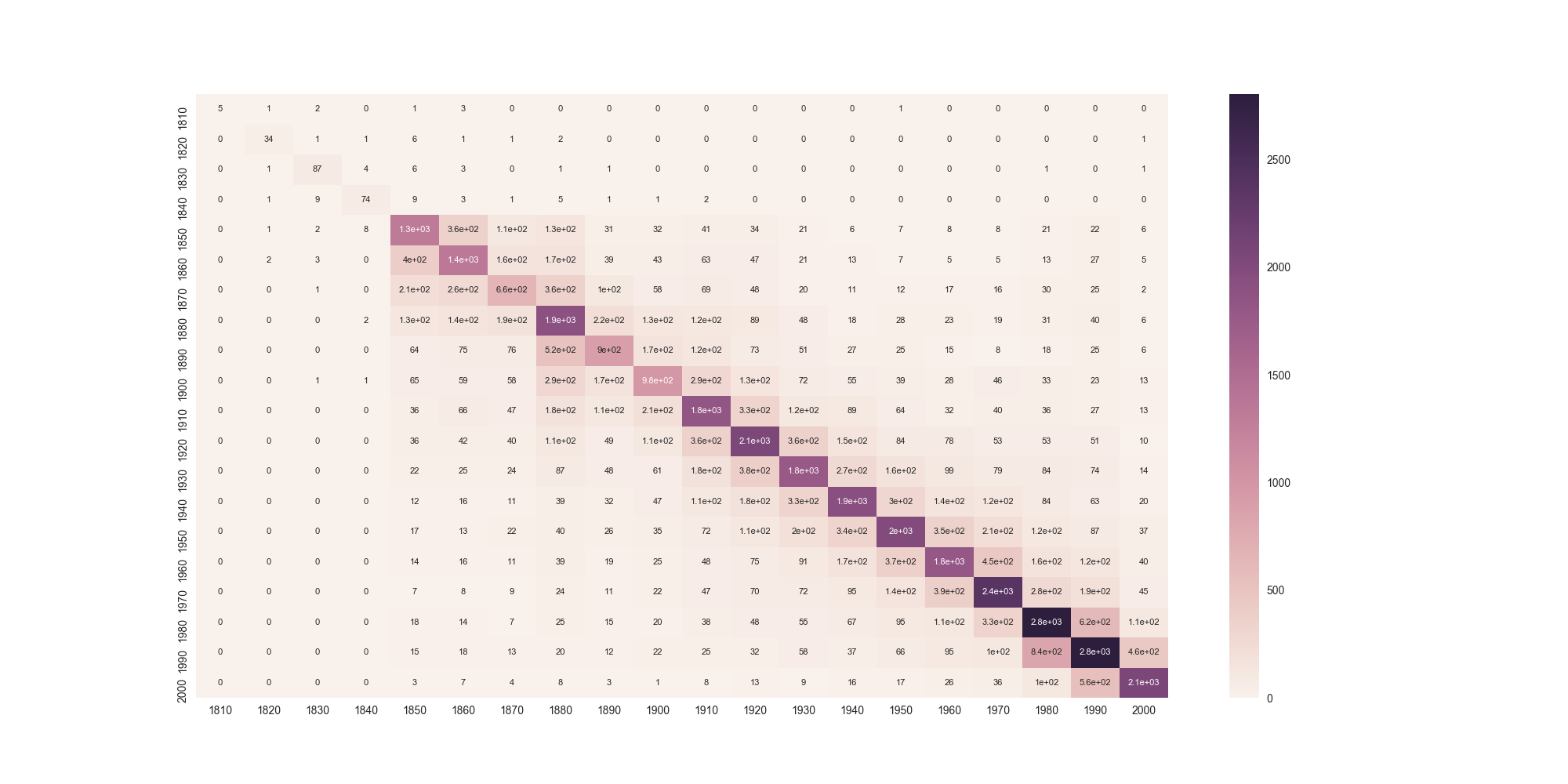
We also can see that the document frequency is similarly almost balanced. We would like to make sure that the token frequency of the merged dataset including both COHA and NYT would have relatively balanced token frequencies, as COHA was balanced via token frequencies.



We see that the token frequency of COHA/NYT is very similar in shape as that of only COHA. This indicates that our merging has maintained a distribution which has the same balance assumptions as the original COHA dataset.

**Modeling COHA/NYT**

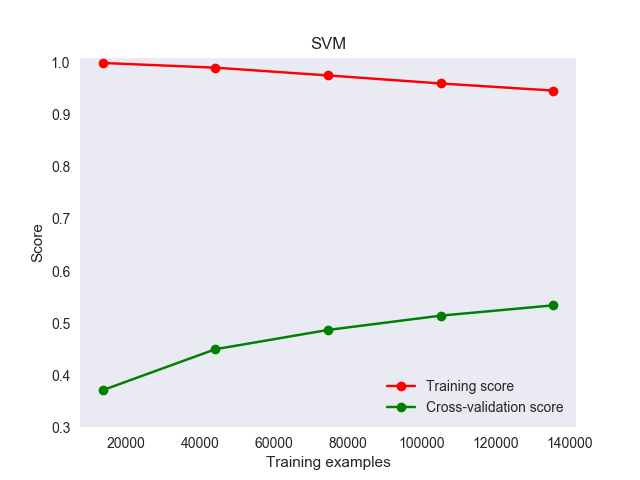
Having implemented cross validation, learning curves, etc in the COHA only case, I trained my SVM model on the merged COHA/NYT dataset without the need for specific implementations. First I found optimal hyper parameters via grid search using trial and error to determine reasonable hyper parameter ranges.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MAE | Accuracy | Precision | Recall | F1 | Support |
| 11.075 | 0.566 | 0.57 | 0.57 | 0.56 | 50844 |

We can see that SVM now performs almost 20% worse than on vanilla COHA. Additionally, the MAE score increased to almost double the value of 6.09 achieved prior.

When I visualized the learning curve for COHA/NYT I immediately realized that I had underestimated how much more difficult the NYT corpus made the document dating problem. SVM is unable to achieve perfect training scores on COHA/NYT. Additionally, the accuracy seems to be converging to 0.5 as a function of training size.



**Apples to Apples Comparison**

Although my model’s performance had dramatically decreased on the extended dataset, there was no good way to determine what the meaning of this was without some baseline comparison. To achieve this I would compare my model performance against the RNN implementation by Yingtao. During my extensive discussion with Yingtao on this subject we came to the agreement that the best way to directly compare our models was by running the trained RNN on the same test set which I used to evaluate my models. This was because the RNN was trained using Google Ngrams while my models were trained using TF-IDF BoW features extracted from the NYT/COHA corpus itself. Due to the differences in feature representation, there was no meaningful way to “train” either model on the other model’s training set.

To run the RNN on my test data I needed to extract the documents and labels and save as a new file in the same format that they were parsed in the RNN implementation. I then made some changes to the RNN code which would allow parsing and forward inference with the following results:

|  |  |  |
| --- | --- | --- |
| Dataset | MAE | Accuracy |
| NYTimes | 25.115 | 0.018 |
| COHA\_100 | 25.862 | 0.026 |
| COHA\_500 | 21.985 | 0.031 |
| COHA\_1000 | 21.377 | 0.032 |
| COHA\_2000 | 20.979 | 0.034 |

Next I would need to implement the same subdocument parsing logic to train my models on 100, 500, 1000, and 2000, subdocument splits of COHA. My tuned SVM trained on COHA/NYT results were:

|  |  |
| --- | --- |
| Dataset | MAE |
| NYTimes | 11.918 |
| COHA\_100 | 25.179 |
| COHA\_500 | 14.190 |
| COHA\_1000 | 11.865 |
| COHA\_2000 | 10.500 |

As we can see my results are almost twice as good with respect to MAE.