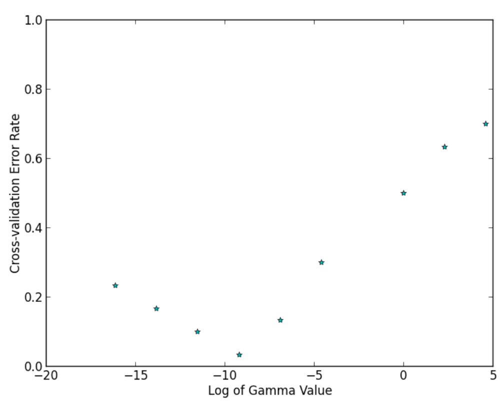
CMSC 25010 HW 5

Andrew Beinstein

May 20, 2013

1 Description

In this assignment, I used the Frequentist approach using an MLE and pseudocounts to predict the authors of the Federalist Papers. To do so, I fit two multinomial models (one for Hamilton and one for Madison), computed the log-likelihood of each paper with both models, and predicted the author whose model gave the largest log-likelihood. To determine the optimal pseudocount, I used cross-validation. Below is a plot describing the error rate as a function of the log of the gamma value chosen:



As you can see, the gamma value with the lowest error rate has a logarithm of around -10, corresponding to $\gamma = .0001$. So, I chose this gamma value to use for my predictions.

In constructing the word count dictionary, I decided to just remove punctuation. Before I settled on this method of preprocessing, I investigated the effect of also making everything lowercase, just making everything lowercase, and of doing no preprocessing whatsoever. To evaluate these options, I computed the cross-validation error and the average distance between the two log-likelihoods. I wanted to maximize the distance between the two log-likelihoods, since that would indicate more confidence in the chosen author. Here are my results:

Preprocessing	Cross-Validation Error	Average Distance
Remove punctuation and lowercase	0.1	172.548597143
No preprocessing	0.03	152.712089187
Remove punctuation	0.03	175.502080164
Lowercase	0.03	150.152509103

Except for the first result, all the other results had the same cross-validation error. I chose to just remove punctuation because that resulted in the lowest cross-validation error and the highest average distance. This suggests that each author starts his sentences with similar words, so by distinguishing capitalized words from uncapitalized words, we may have more information with which to predict the correct author. Conversely, the punctuation that the authors use may provide unhelpful noise.

2 Cross-Validation Results

Document	Log-Likelihood Hamilton	Log-Likelihood Madison	Prediction
hamilton1.txt	-2410.45023823	-2600.16943982	Hamilton
hamilton2.txt	-2501.8380319	-2831.10997394	Hamilton
hamilton3.txt	-2359.38661	-2546.828869	Hamilton
hamilton4.txt	-2306.98822675	-2371.40206415	Hamilton
hamilton5.txt	-2729.57937624	-3044.03282718	Hamilton
hamilton6.txt	-2420.67187514	-2635.89586421	Hamilton
hamilton7.txt	-1423.24416692	-1545.43289231	Hamilton
hamilton8.txt	-2898.11351065	-3190.73795197	Hamilton
hamilton9.txt	-2205.46641102	-2366.14910303	Hamilton
hamilton10.txt	-1948.46947151	-2017.64169849	Hamilton
hamilton11.txt	-2231.87752614	-2404.3758387	Hamilton
hamilton12.txt	-3027.90468051	-3241.16701214	Hamilton
hamilton13.txt	-2103.02678662	-2114.34135165	Hamilton
hamilton14.txt	-2374.4354125	-2502.11691288	Hamilton
hamilton15.txt	-1838.90542585	-1857.59921179	Hamilton
madison1.txt	-3123.61900145	-3168.5765047	Hamilton
madison2.txt	-2515.0087805	-2502.55981487	Madison
madison3.txt	-2872.03391807	-2803.87389212	Madison
madison4.txt	-3256.50303819	-3105.69992878	Madison
madison5.txt	-2900.70868731	-2626.50785282	Madison
madison6.txt	-3403.225883	-3084.36510039	Madison
madison7.txt	-3351.75912683	-3321.62550298	Madison
madison8.txt	-3022.125462	-2935.66779247	Madison
madison9.txt	-3249.89462441	-3041.54774475	Madison
madison10.txt	-3037.2639055	-2791.30753668	Madison
madison11.txt	-2314.74827586	-2144.00973411	Madison
madison12.txt	-2701.26162299	-2570.37464085	Madison
madison13.txt	-3907.09387761	-3235.72501105	Madison
madison14.txt	-2448.64340594	-2082.45502641	Madison
madison15.txt	-2355.93654768	-2269.11093075	Madison

3 Predicting the Documents of Unknown Authorship

Document	Log-Likelihood Hamilton	Log-Likelihood Madison	Prediction
unknown1.txt	-2045.56080554	-1910.79220363	Madison
unknown2.txt	-1751.67993928	-1666.29201826	Madison
unknown3.txt	-2275.52005672	-2087.68046752	Madison
unknown4.txt	-2280.97980074	-2120.99608225	Madison
unknown5.txt	-2500.27375744	-2409.72124037	Madison
unknown6.txt	-2340.97657527	-2261.37715919	Madison
unknown7.txt	-2390.79011838	-2315.69314349	Madison
unknown8.txt	-2050.38345067	-1930.2737154	Madison
unknown9.txt	-2522.97654903	-2473.02463587	Madison
unknown10.txt	-2613.09840101	-2466.71688714	Madison
unknown11.txt	-3107.07525267	-2941.92955858	Madison

4 Extra Credit

For extra credit, I also experimented with stemming every word. To do this, I used the Python NLTK toolkit's Snowball Stemming algorithm. This algorithm converts every word to its root form (i.e. talking → talk, countries → country). I also experimented with only using the most frequent words, and with removing the most frequent words. My results are below.

Preprocessing	Cross-Validation Error	Average Distance
Snowball Stem	0.06	159.549606857
20 most frequent words	0.13	96.5525539227
50 most frequent words	0.06	130.490581036
100 most frequent words	0.03	123.556328483
Remove 10 most frequent words	0.1	147.596044907
Remove 20 most frequent words	0.03	138.773622812
Remove 50 most frequent words	0.03	107.01465109
Remove 100 most frequent words	0.06	72.8395239884

None of these preprocessing steps beat simply removing punctuation. However, this analysis provides some interesting insights. First, it appears that removing the most frequent words is generally more helpful than just including the most frequent words. This means that the frequencies of the most common words ('the', 'of', 'and', etc) do not help predict the authors very well. Nevertheless, the fact that we can predict authorship using just the 20 most frequent words with an error rate of only 13% is quite remarkable. The most powerful of these new methods was removing the 20 most frequent words. Stemming, the most complicated of these methods, had a high average distance, but the error rate was larger. Yet, the sample size (30 cross-validations) may be too small to conclude for certain which of these preprocessing steps is the best one to use.