Daniel Mané

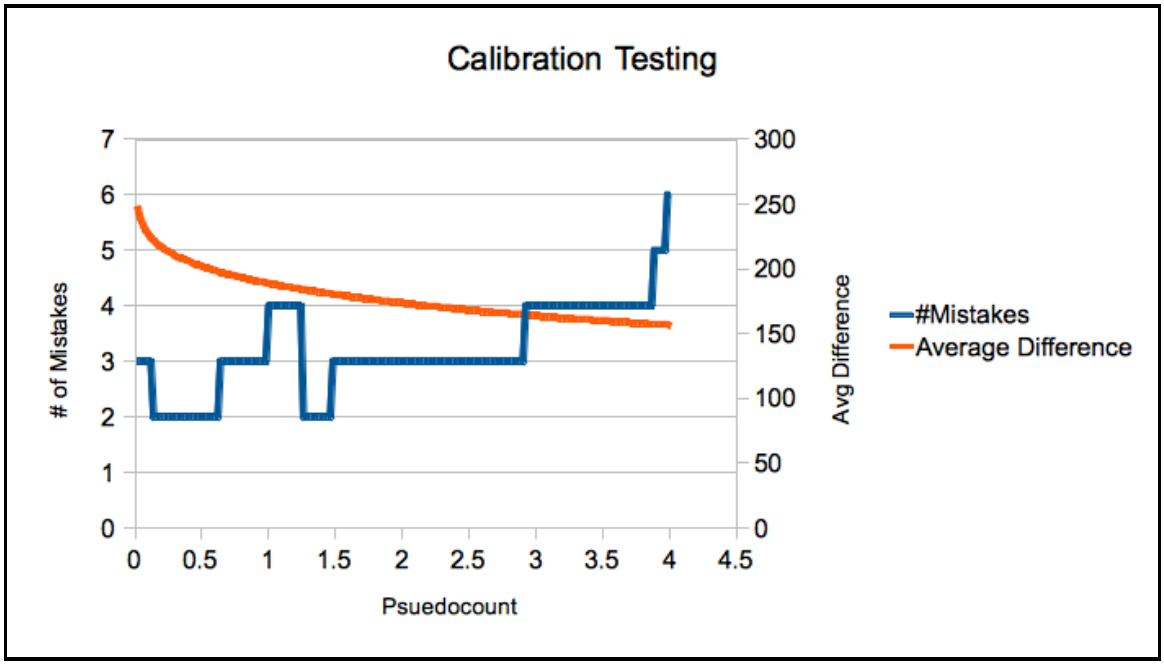
AI Homework 4

Methodology:

For each document *d* being tested, I first construct a dictionary for the test group *G*. This test group dictionary contains all words in all documents in *G*; it is equivalent to the concatenation of all documents in the group. Then I add a psuedocount for each word in the dictionary, and each word in the document being tested. I divide each value in the dictionary by the total number of words in the dictionary, producing a theta value for each word. I compute the log-likelihood value, which is a monotonic transformation of the likelihood that the document *d* came from the test group *g*.

Since the log-likelihood is a monotonic transformation of the likelihood value, we can use it as a basis for comparison between different test groups, to see which distribution is the best fit for the document.

One parameter needs to be chosen: the psuedocount *p*. To choose *p*, I did calibration testing; I ran the program on the known training documents for many different values of *p*. The results are below: the left axis shows the number of mistakes given a certain *p*, and the right axis shows the average difference between the log-likelihood values.



Note that the difference for each example was computed (LLV(Correct) – LLV(Incorrect)), so that when the program made a mistake, the LLV difference was negative for that datapoint. My intuition was that a higher average difference corresponds to a greater degree of differentiation between the correct and incorrect classification, and so suggests better performance by the program. This interpretation is not fully supported by the data. Average difference seems closely correlated with *p*, but not very closely correlated with the number of mistakes; for very low *p*, average difference is very high, but 3 mistakes are made rather than 2.

I was also surprised that the number of mistakes had two distinct minima, one near p=.4 and one near p=1.4. When p=.4, the program mistakenly classifies madison1 and madison2 as being written by Hamilton. However, when p=1.4, it mistakenly classifies hamilton13 and hamilton15 as being written by Madison.

Based on my calibration testing, I chose to use $p = .5$. This choice was in one of the lowest mistake regions, gave a high average difference, and was a nice round number.

Here are the results of validation testing with $p = .5$ (mistakes in bold):

Document:	Classification:	LLV:	Diff:
hamilton1.txt	Hamilton	-12632	256
hamilton10.txt	Hamilton	-9849	89
hamilton11.txt	Hamilton	-12454	186
hamilton12.txt	Hamilton	-21878	167
hamilton13.txt	Hamilton	-11533	4
hamilton14.txt	Hamilton	-13567	75
hamilton15.txt	Hamilton	-9241	13
hamilton2.txt	Hamilton	-14311	257
hamilton3.txt	Hamilton	-12750	211
hamilton4.txt	Hamilton	-12563	15
hamilton5.txt	Hamilton	-15930	410
hamilton6.txt	Hamilton	-13729	264
hamilton7.txt	Hamilton	-6008	90
hamilton8.txt	Hamilton	-19523	254
hamilton9.txt	Hamilton	-12580	224
madison1.txt	**Hamilton**	**-18913**	**-39**
madison10.txt	Madison	-17762	361
madison11.txt	Madison	-12545	193
madison12.txt	Madison	-15836	80
madison13.txt	Madison	-16778	793
madison14.txt	Madison	-11450	431
madison15.txt	Madison	-12761	142
madison2.txt	**Hamilton**	**-13589**	**-32**
madison3.txt	Madison	-17330	126
madison4.txt	Madison	-21043	260
madison5.txt	Madison	-15346	365
madison6.txt	Madison	-18435	427

madison7.txt	Madison	-22535	79
madison8.txt	Madison	-17454	150
madison9.txt	Madison	-21174	210

Here are the classifications of the unknown documents using p =.5:

Document:	Classification:	LLV:	Difference:
unknown1.txt	Madison	-10015	188
unknown10.txt	Madison	-14902	187
unknown11.txt	Madison	-18912	342
unknown2.txt	Madison	-7012	126
unknown3.txt	Madison	-11501	297
unknown4.txt	Madison	-11403	162
unknown5.txt	Madison	-13432	158
unknown6.txt	Madison	-12055	129
unknown7.txt	Madison	-12654	156
unknown8.txt	Madison	-9465	137
unknown9.txt	Madison	-13639	104