Midterm 2 write-up

Shane Stevens

In all honesty, I am not 100% sure what is causing the low accuracy, but I have a few theories, based on different things that I tried in the code. One thing I noticed was that there were quite a few ties between the two different probabilities in the code, and the way that the ties are dealt with can have a big impact on the accuracy. The most accurate way was marking all emails with tied probabilities as spam. I’m guessing this is the most accurate solution since ties only really occur when none of the words in the email are stored in the vocab, and this is more likely to happen in spam emails than ham emails. For the record, if emails are marked as ham during a tie, the accuracy drops from around 65-70% to closer to 40%.

I think that my more important discovery has to do with how Laplace smoothing is handled though. If my understanding Is correct, in Laplace smoothing, for each the probability for each word, you add 1 to the numerator and the size of the vocab to the denominator, where the numerator is the number of either spam or ham emails that contain the word, and the denominator is the total number of either spam or ham emails.

I believe that the way ties and Laplace smoothing are handled is where the low accuracy is being caused. Mainly the Laplace smoothing, since I already explained the best solution for the tie issue. I believe that the issue is that I used the size of the vocab in the denominator, when what I should have done instead was use the size of the vocab of words that are used in each class. For example, the ham probability would use the number of words that appeared in ham emails, and the spam probability would use the number of words that appeared in spam emails. This is because using the size of the entire vocab assumes that both spam and ham are equally likely to have an unseen word in them, which is untrue. My solution would better reflect this fact because the probability of unseen words being in each class is now related to the number of words in unique words in each class, which itself could reflect the likelihood of new words showing up in both classes.