EECS 349 Machine Learning Homework 5

Xinyi Chen

1)

Please run python spamfilter.py <TrainSpam> <TrainHam> <TestEmails>

e.g. python *spamfilter.py* spam/ easy_ham/ easy_ham/

2)

Please run python spamfilter.py <TrainSpam> <TrainHam> <TestEmails>

e.g. python *spamfilter.py* spam/ easy_ham/ easy_ham/

3)

If there were no issues with underflow or precision, equation (6) and (7) always return the same result, because we only care about the maximum point, and log is an increasing function, if A > B, then logA > logB, so the maximum will be at the same point.

$$V_{NB} = argmaxP(v_j) \prod_i P(a_i|v_j)$$

$$=> \quad V_{NB} = argmax(\log (P(v_j) \prod_i P(a_i|v_j))))$$

$$=> \quad V_{NB} = argmax(logP(v_j) + \log (\prod_i P(a_i|v_j)))$$

$$=> \quad V_{NB} = argmax(logP(v_j) + \sum_i \log (P(a_i|v_j)))$$

But if there's underflow, they may give different results. Because equation (6) is multiplication of many numbers which are all between 0 and 1, so the result of equation (6) is a very small number between 0 and 1. If the result is too small, it may become 0 because of underflow. So if there are two results, A and B (A > B) both are very small numbers, using equation (6) they will both be 0, so the result becomes A = B. However because equation (7) is an addition equation, the addition result won't beyond the precision, they won't become 0.

4)

A)

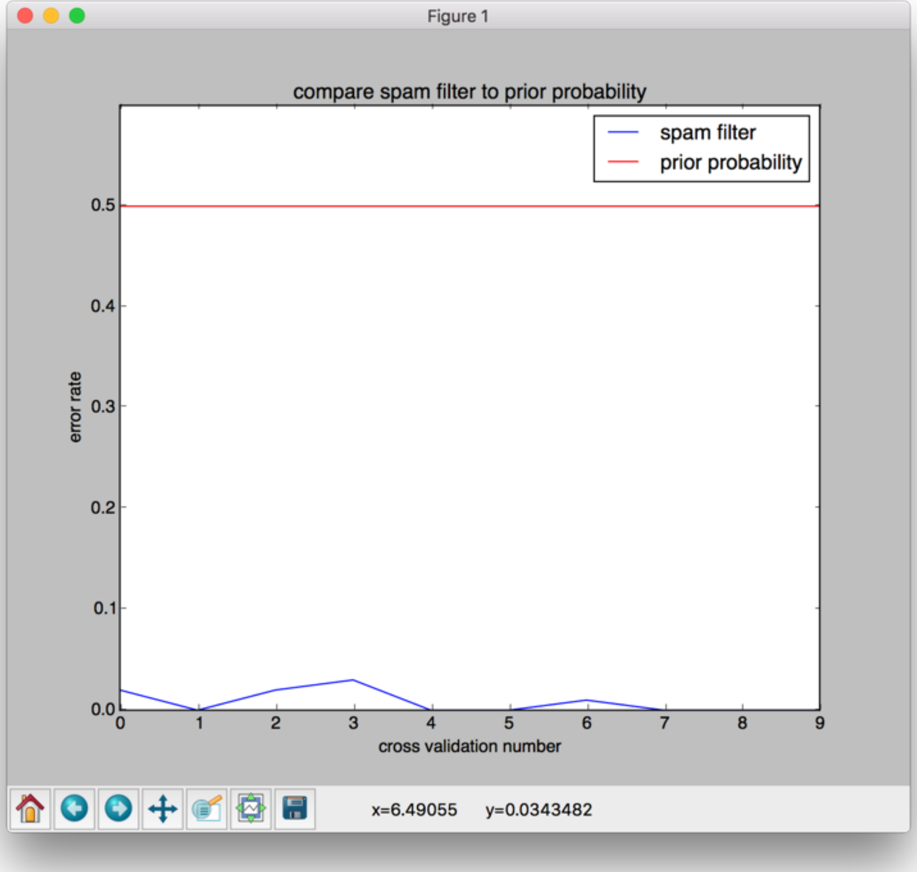
I used 9/10 of ham and spam emails to build my dictionary and the rest 1/10 of ham and spam emails to test everytime, and the total number of ham and spam emails I used are equal(both are 501, because spam has only 501 emails, so I picked 501 emails from ham randomly to make them have same data size). No they're not the same. I choose dataset this way is because the spam and ham emails are the only data I got and their data size are different, and to truly have a test on my filter, testing data should be different with training data. Also by doing this I can do cross validation to get a more accurate result. There are 1,000 emails to build dictionary, 500(50%) are spam, 500(50%) are ham. 102 emails to test, 51(50%) are spam, 51(50%) are ham. I don't think my data can represent the spam we might see in the real world, because the datasets are too small(only thousands of emails).

B)

I used 10 folds cross validation. Measure error: error rate = number of misclassified test emails/number of test emails, the smaller error rate, the better filter. E.g. There are 10 test spam emails, 2 of them are misclassified as ham, and 10 ham emails, 3 of them are misclassified as spam, so the error rate is (2+3)/(10+10) = 0.25.

C)

Yes, my spam filter works better than just using the prior probability in my dataset. As we can see in the graph, the 10 folds cross validation shows that in each experiment, my filter has lower error rate.



And I choose paired samples t-test as my statistical tests, because variants are independent and identically distributed(chose from the same data set independently and randomly), paired(change filter to generate paired response on same data) and variants have same variance(my filter error rate variance: 0.000116, prior probability error rate variance: 0.0) and also fit Gaussian distribution(scipy.stats.mstats.normaltest() shows that they both fit Gaussian distribution). The null hypothesis: the difference between two system variants are not statistically significant. I choose p = 0.05, because it is convenient to take this point as a limit in judging whether a deviation ought to be considered significant or not. The p value of this experiment is 2.98 * 10^(-16) which is smaller than 0.05, so the null hypothesis is rejected, the difference between two filters are statistically significant, my spam filter is better.