CSCI 1950-F Homework 1:
Naive Bayes Spam Classification

Brown University, Spring 2012

Homework due at 12:00pm on February 10, 2012

Hello, I am a prince in desperate need of a personal favor. I've been receiving a lot of unwanted emails lately and would really appreciate it if you would build me a spam filter. It shouldn't be too tough with some simple but effective machine learning. This is a classification problem in which you will predict one of two labels: *Spam* or *Ham*. Our dataset is drawn from emails released during your country's regrettable Enron investigation.

To help you out, I have given each of the W words found in the dataset an index. A $W \times 1$ Matlab cell array lists the character string corresponding to each vocabulary index. I've also included train, validation, and test datasets, defined in terms of Data and ID matrices. The sparse Data matrix (e.g., trainData) is $D \times W$, where D is the number of documents. Each row of this matrix represents one email, with column j containing the number of times word j appears in that email. The ID matrix (e.g., trainID) is $D \times 1$, where each row contains a 1 if the corresponding document is Spam, and a 0 if it is Ham.

Please build me a naive Bayes classifier. Assume that the Spam and Ham classes are equally likely *a priori*. Model each word instance (token) from an email of class c as being independently drawn from a class-specific, W-dimensional categorical distribution θ_c. Document word counts then follow class-specific multinomial distributions.

Question 1:

a) Specify formulas for the maximum likelihood (ML) estimates of the class-specific word distributions θ_c. A detailed derivation is not necessary. Calculate these ML estimates, and evaluate your model by predicting the most likely labels for the validation dataset. Limit the words used in this experiment to those that occur at least once in the training dataset; any validation or test instances of these words are then also ignored. If you don't do this, you will get near random (50%) accuracy, while you should get over 80% accuracy with this fix. Write a sentence or two explaining why this preprocessing is necessary, and report your validation accuracy.

b) You can potentially improve your model by reducing overfitting to the limited training data. Assume that infrequent words represent noise that leads to such overfitting. Retrain your model using only words w that appear more than ρ times in the training data, for candidate thresholds $\rho \in \{0, 10, 20, \ldots, 200\}$. Plot validation accuracy versus ρ.

c) Alternatively, we can retain all of the data but use Bayesian estimates of the class-specific multinomial distributions. To do this, we assign symmetric Dirichlet priors $\theta_c \sim Dir(\alpha, \alpha, \ldots, \alpha)$. *Estimate parameters by their posterior mean under this prior, for each* $\alpha \in \{.0001, .001, .01, .1, 1, 10\}$. *Plot validation accuracy versus* α, *using a logarithmic scale for* α *so that the points are evenly spaced.*

d) Consider all of the models from parts (b) and (c), and pick the one with the highest validation accuracy. Which regularization approach was most effective? What is the test accuracy of the chosen model?

e) Using the best model from part (d), consider the following score function:

$$g(w) = \frac{p(word = w \mid class = Spam)}{p(word = w)}$$

Compute this score for each word w, *and report the 10 words with the highest value of this ratio as being highly indicative of the Spam class. For this ranking, why didn't we ask you to simply report the words for which* $p(word = w \mid class = Spam)$ *is highest? Also report the 10 words which are most indicative of the Ham class.*

f) Think about the structure of this naive Bayes generative model for emails. Because each word is treated as an independant draw from a multinomial distribution, this is commonly called a "bag of words" model. Describe how you would compose a spam email that a bag of words model would have trouble identifying as spam. This is something that people who lack your good judgment get paid lots of money for.

Hints: If you directly calculate class-specific likelihood functions, you will encounter numerical underflow problems. Since you will only be interested in which likelihood (Ham or Spam) is greater, it is better to instead calculate log-likelihoods. Also, make sure that when you limit the word indices used in your experiments using the threshold ρ, you do it before you estimate parameters.