CS534 — Implementation Assignment 2 — Due Oct 24th 11:59PM, 2015

General instructions.

1. The following languages are acceptable: Java, C/C++, Matlab, Python and R.
2. You can work in team of up to 3 people. Each team will only need to submit one copy of the source code and report.
3. Your source code and report will be submitted through the TEACH site

`https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth`

Please clearly indicate your team members' information.
4. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.
5. In your report, the results should always be accompanied by discussions of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Naive Bayes (total 70pts)

In this assignment you will implement the Naive Bayes classifier for document classification with both the Bernoulli model and the Multinomial model. For Bernoulli model, a document is described by a set of binary variables, and each variable corresponds to a word in the vocabulary V and represents its presence/absence. The probability of observing a document $\mathbf{x}$ given its class label y is then defined as:

$$p(\mathbf{x}|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}(1 - p_{i|y})^{(1-x_i)}$$

where $p_{i|y}$ denotes the probability that the word i will be present for a document of class y. If $x_i = 1$, the contribution of this word to the product will be $p_{i|y}$, otherwise it will be $1 - p_{i|y}$.

For the Multinomial model, a document is represented by a set of integer-valued variables, and each variable x_i also corresponds to the i-th word in the vocabulary and represents the number of times it appears in the document. The probability of observing a document $\mathbf{x}$ given its class label y is defined as:

$$p(\mathbf{x}|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}$$

Here we assume that each word in the document follows a multinomial distribution of $|V|$ outcomes and $p_{i|y}$ is the probability that a randomly selected word is word i for a document of class y. Note that $\sum_{i=1}^{|V|} p_{i|y} = 1$ for $y = 0$ and $y = 1$.

Your implementation need to estimate $p(y)$, and $p_{i|y}$ for $i = 1, \cdots, |V|$, and $y = 1, 0$ for both models. For $p(y)$, you can use MLE estimation. For $p_{i|y}$, you MUST use Laplace smoothing for both types of models.

Apply your Naive Bayes classifiers to the provided 20newsgroup data. By that I mean learn your models using the training data and apply the learned model to perform prediction for the test data and report the performance.

One useful thing to note is that when calculating the probability of observing a document given its class label, i.e., $p(\mathbf{x}|y)$, it can and will become overly small because it is the product of many probabilities. As a result, you will run into underflow issues. To avoid this problem, you should operate with log of the probabilities.

Data set information: The data set is the classic 20-newsgroup data set. There are six files.

- vocabulary.txt is a list of the words that may appear in documents. The line number is word's id in other files. That is, the first word ('archive') has wordId 1, the second ('name') has wordId 2, etc.

- newsgrouplabels.txt is a list of newsgroups from which a document may have come. Again, the line number corresponds to the label's id, which is used in the .label files. The first line (class 'alt.atheism') has id 1, etc.
- train.label contains one line for each training document specifying its label. The document's id (docId) is the line number.
- test.label specifies the labels for the testing documents.
- train.data describes the counts for each of the words used in each of the documents. It uses a sparse format that contains a collection of tuples "docId wordId count". The first number in the tuple species the document ID, and the second number is the word ID, and the final number species the number of times the word with id wordId in the training document with id docId. For example "5 3 10" species that the 3rd word in the vocabulary appeared 10 times in document 5.
- test.data is exactly the same as train.data, but contains the counts for the test documents.

  Note that you don't have to use the vocabulary.txt file in your learning and testing. It is only provided to help possibly interpret the models. For example, if you find that removing the first feature can help the performance, you might want to know what word that first feature actually corresponds to in the vocabulary.

Basic implementation:

1. (5 pts) Please explain how you use the log of probability to perform classification.
2. (10 pts) Report the overall testing accuracy (number of correctly classified documents over the total number of documents) for both models.
3. (5 pts) Are there any news groups that are confused more often than others? Why do you think this is? To answer this question, you might want to produce a K by K confusion matrix, where K is the number of classes, and the i, j-th entry of the matrix shows the number of class i documents being predicted to belong to class j. A perfect prediction will have only diagonal elements in this confusion matrix.

Priors and overfitting:
(20 pts) In this part, we will focus on the multinomial model and experiment with different priors. In particular, your last set of experiments use' Laplace smoothing for MAP estimation, which corresponding to using $Dirichlet(1+\alpha, ..., 1+\alpha)$ with $\alpha = 1$ as the prior. In this part, you will retrain your classifier with different values of α between 10^{-5} and 1 and report the accuracy on the test set for different α values. Create a plot with value of α on the x-axis and test set accuracy on the y-axis. Use a logarithmic scale for the x-axis. Comment on how the test set accuracy change as α changes and provide a short explanation for your observation.

Identifying important features:
(20 pts) For this part, design and test a heuristic to reduce the vocabulary size and improve the classification performance. This is intended to be open-ended exploration. Please describe clearly what is your strategy for reducing the vocabulary size and the results of your exploration. A basic pointer to seed your exploration is that we would like to remove words of no discriminative power. How can we measure the discriminative power of a word?