Boston University
Department of Electrical and Computer Engineering

ENG EC 503 (Ishwar) Learning from Data

Assignment 4

Issued: Mon 2 Oct 2017 **Due:** 5pm Fri 13 Oct 2017 in box outside PHO440 + Blackboard
Required reading: Your notes from lectures and additional notes on website on classification.

- **Advise**: This homework assignment requires a large amount of time and effort. We urge you to start right away. This assignment cannot be finished by staying up all night just before the deadline.
- This homework assignment requires programming in MATLAB. If you are new to MATLAB programming, please refer to the following link for a primer:
`http://www.math.ucsd.edu/~bdriver/21d-s99/matlab-primer.html`
- You will be making two submissions: (1) A paper submission in the box outside PHO440. (2) An electronic submission of all your MATLAB code (in a single zipped file appropriately named as described below) to blackboard.
- **Paper submission:** This must include all plots, figures, tables, numerical values, derivations, explanations (analysis of results and comments), and also printouts of all the MATLAB .m files that you either created anew or modified. Submit color printouts of figures and plots whenever appropriate. Color printers are available in PHO307 and PHO305. Be sure to annotate figures, plots, and tables appropriately: give them suitable *titles* to describe the content, label the *axes*, indicate *units* for each axis, and use a *legend* to indicate multiple curves in the plots. Please also explain each figure properly in your solution.
- **Blackboard submission:** All the MATLAB .m files (and only .m files) that you either create anew or modify must be appropriately named and placed into a **single** directory which should be zipped and uploaded into the course website. Your directory must be named as follows:
`<yourBUemailID>_assignment4`. For example, if your BU email address is `charles500@bu.edu` you would submit a single directory named: `charles500_assignment4.zip` which contains all the MATLAB code (and only the code).
- **File naming convention:** Instructions for file names to use are provided for each problem. As a general rule, each file name must begin with your BU email ID, e.g., `charles500_<filename>.m`. The file name will typically contain the problem number and subpart, e.g., for problem 4.1b, the file name would be `charles500_assignment4_1b.m`. Note that the dot . in 4.1 is replaced with an underscore (this is important).

Problem 4.1 Naive Bayes Text Document Classifiers: In this problem we classify text documents using Naive Bayes. We use the classic *20 newsgroup* dataset. Here, each document is labeled as one of 20 different classes and is provided in `data_20news.zip`. This directory contains the following files:

1. `vocabulary.txt`: a list of all the words that can possibly appear in the documents. The line number of a word is its ID (Word_ID).

2. `newsgrouplabels.txt`: the names of the 20 classes. The line number of a class (label) is its ID.

3. `train.data` and `test.data`: the words in each of the documents. Each line of each file is of the form: "Document_ID, Word_ID, Word_count". Word_count is the number of times word Word_ID appears in document Document_ID.

4. `train.label` and `test.label`: the labels of the training and test documents.

5. `stoplist.txt`: a list of so-called "stop words" such as "the", "is", "on", etc. This will be used only in part (f).

Let $\{1, \ldots, W\}$ denote the W distinct words in the vocabulary. We view each document $\mathbf{x}_j$, $j = 1, \ldots, n$ as a collection of words $\{w_{1,j}, \ldots, w_{d_j,j}\}$ where d_j is the number of words in document $\mathbf{x}_j$ and y_j is the label of the document. Each word $w_{i,j}$ takes values in $\{1, \ldots, W\}$. The Naive Bayes classifier assumes that:

$$P(Y_j = c|\mathbf{x}_j) \propto P(Y_j = c) \prod_{i=1}^{d_j} P(w_{i,j}|Y = c), \quad c = 1, \ldots, 20 \tag{1}$$

We denote $P(Y_j = c) = p_Y(c), c = 1, \ldots, 20$ and $P(w_{i,j} = w|Y = c) = \beta_{w,c}, w = 1, \ldots, W, c = 1, \ldots, 20$. Note that by definition, $\sum_{w=1}^{W} \beta_{w,c} = 1$, for each $c = 1 \ \ldots, 20$.

(a) Write a MATLAB script to load and parse the data. Name this file `<yourBUemailID>_assignment4_1a.m`. **Report** the following statistics

- The total number of **unique** (i.e., distinct) words that appear in the training set, the test set, and the entire dataset (training set + test set), respectively.
- The average and standard deviation of document length (in terms of number of words) in the training and test sets, respectively. Note: The standard deviation is the square root of the population variance.
- The total number of unique words that appear in the test set, but not in the training set.
- The 10 most frequent words and their number of appearances in the entire dataset (training set + test set).
- The smallest number of times that a word appears in the entire dataset (training set + test set), and the number of words that appear these many times. Among these words, list all the ones that start with "od".

Helpful MATLAB functions: `textread, sparse`.

We next train a Naive Bayes classifier. We first learn the parameters $\beta_{w,c}$'s using Maximum Likelihood Estimation (MLE):

$$\beta_{w,c} \propto n_{w,c} \tag{2}$$

where $n_{w,c}$ is the total number of times the word w appears across all training documents having label c.

(b) Train a Naive Bayes classifier using the MLE rule from Eq. 2 and the training set. Afterwards, evaluate this classifier on the test set. For any test document whose probabilities $P(Y = c|\mathbf{x})$ are equal for every class c, assign it to the class with the maximum observed prior probability $P(Y = c)$ in the training set. Do not use the built-in MATLAB function for the Naive Bayes classifier. **Submit** your script with name `<yourBUemailID>_assignment4_1b.m`. **Report** the following results:

- Attach a plot of the observed prior probabilities $P(Y = c)$ for $c = 1, \ldots, 20$, in the training set. Comment on what you obseve.
- Among the $W \times 20$ estimated parameters (the $\beta_{w,c}$'s), how many of them are zero?
- For some test documents, $P(Y = c|\mathbf{x}) = 0$ for all $c = 1, \ldots, 20$. What is the total number of such test documents? Can you explain why their probabilities are zero?
- The test CCR.
- The 20×20 confusion matrix. Make sure that you annotate the confusion matrix appropriately. Comment on what you observe.

(c) One strategy to overcome the issue encountered in part (b) is to remove words that only appear in the test documents (but not in the training documents). Remove these words. Repeat training and testing using the MLE rule in (b) and **report:**

- The total number of non-zero estimated $\beta_{w,c}$'s.
- The test CCR.
- The total number of test documents where $P(Y = c|\mathbf{x}) = 0$ for all $c = 1, \ldots, 20$. If there are still such test documents even after removing the words that only appear in the test set, explain why.

To learn the parameters $\beta_{w,c}$'s we can also use the Maximum Aposteriori Probability (MAP) rule. It is common practice to impose a Dirichlet prior on the $\beta_{w,c}$'s and as such, Eq. 2 becomes:

$$\beta_{w,c} \propto n_{w,c} + (\alpha - 1) \tag{3}$$

for some small $(\alpha - 1) > 0$.

(d) Train a Naive Bayes classifier using the MAP rule in Eq. 3 with $(\alpha - 1) = 1/W$, and evaluate it on the test set. Do not use the built-in MATLAB function for the Naive Bayes classifier. **Submit** your script with name <yourBUemailID>_assignment4_1d.m. **Report** the following results:

- The test CCR.
- The 20×20 confusion matrix. Make sure that you annotate the confusion matrix appropriately.
- The names of the 5 most confused class pairs and their degrees of confusion. The degree of confusion between class A and class B is defined as

$$d(A, B) = \frac{P(h(\mathbf{x}) = B|Y = A) + P(h(\mathbf{x}) = A|Y = B)}{2} \tag{4}$$

(e) Calculate the test error for different choices of α's in the range given by $(\alpha - 1) = 10^{-5}, 10^{-4.5}, 10^{-4}, 10^{-3.5}, ..., 10^{1}, 10^{1.5}$. Plot the test CCRs (y-axis) as functions of $(\alpha - 1)$ (x-axis). Use a log-scale for $(\alpha - 1)$.

(f) In classification tasks with text observation, the most common words (*stop words*) such as "the","is", etc., may be less informative. A common practice is to remove such stop words. The file `stopword.txt` contains a list of such stop words. Remove these words from all the documents and the vocabulary and then repeat the steps in part (d). **Report:**

- The total number of **unique** (i.e., distinct) words that appear in the training set, the test set, and the entire dataset (training set + test set), respectively, after removing the stop words.
- The average and standard deviation of document length (in terms of number of words) in the training and test sets respectively.

- The total number of unique words that appear in the test set, but not in the training set.
- The 10 most frequent words and their number of appearances in the entire dataset (training set + test set).
- The smallest number of times that a word appears in the entire dataset (training set + test set), and the number of words that appear these many times. Among these words, list all the ones that start with “aero”.
- The test CCR.