Machine Learning
Rainbow Introduction (by Andrew Ragone)

Introduction

This homework assignment explains the basics of how to operate rainbow. It is intended to get your feet wet, so you have some background on how rainbow functions when doing subsequent rainbow assignments. This document assumes you are working on queen. Rainbow is located in the following path "/misc/itcsl/ml/rainbow/". You should put this directory in your path. The text we are going to use can be found at "/misc/itcsl/ml/20_newsgroups/". If you do not wish to use queen, you can obtain rainbow and the 20_newsgroups from the software directory on the course website and install it on a Unix like OS (e.g. Linux).

Rainbow basics

Rainbow is a command line program that performs statistical text classification. It is based on the *Bow* library (http://www.cs.cmu.edu/~mccallum/bow). The general pattern of rainbow usage is a two steps process:

1. Have rainbow read your documents and write to disk a "model" containing their statistics.
2. Using the model, rainbow performs classification or diagnostics.

Before discussing this process further, you should take a look at the on-line documentation that can be obtained by typing:

```
rainbow --help
```

You should output this information to a text document with the redirect operator ("> somefile.txt") for further study. Rainbow also has information about how the switches should be utilized with the following command:

```
rainbow --usage
```

Modeling Newsgroup Text Documents

Before performing classification or diagnostics with rainbow, you must first have rainbow index your data--that is, read your documents and archive a "model" containing their statistics. The text indexed for the model must contain all the training data. The testing data may also be read as part of the model, or it can be left out and read later. The model is placed in the file system location indicated by the `-data-dir` option. If no `-data-dir` option is given, the name `~/.rainbow` is used by default. (The model name is actually a file system directory containing separate files for different aspects of the model. If the model directory location does not exist when rainbow is invoked, rainbow will create it automatically.)

In the most basic setting, the text data should be in plain text files, one file per document. No special tags are needed at the beginning or end of documents. Thus, for example, you should be able to index a directory of UseNet articles (as we will do here) or MH mailboxes without any preprocessing. The files should be organized in directories, such that all documents with the same class label are contained within a directory. (Rainbow does not directly support classification tasks in which individual documents have multiple class labels.)

To build a model, call rainbow with the `--index` switch, followed by one directory name for each class. For this assignment we are going to only use 3 newsgroups for classification, comp.graphics, sci.med, and talk.politics.misc. One of the problems with newsgroup documents is that they have extraneous header information that we do not want to include in the model. Rainbow has a lexing option `--skip-header`, which makes rainbow only include the text that comes after two new line characters.

```
rainbow --data-dir ~/model/ --skip-header --index
        /misc/itcsl/ml/20_newsgroups/comp.graphics/
        /misc/itcsl/ml/20_newsgroups/sci.med/
        /misc/itcsl/ml/20_newsgroups/talk.politics.misc/
```

Diagnostics of a Model

Rainbow just made a model of the documents in three newsgroups, but the million-dollar questions are "What is in the model?" and "How does it help me classify text from these newsgroups?" In this section you will see the answer to the first question and partial answer to the second.

The model simply stores various statistics on each document such as its corresponding newsgroup (the document's classifier) and the number of times words appear in a document in a matrix format. Use the `--print-matrix` option to print out the model's matrix.

```
rainbow --data-dir model --print-matrix
```

It would be wise to send the output of this data to a file for further analysis, as there is a great deal of data. As for the content of the output, it is as follows:

- Each line consist of one document statistics
- The first entree on the line is the location of the document on the file system
- The second entree on the line is the document's classification
- The following data are the words in the document
- Each word is numerated and has the number of times it shows up in the document

We just had rainbow give the full model with `--print-matrix`. However rainbow is also capable of limiting it's output, for your perl parsing needs. The following is a list of options you can put after --print-matrix

Print entries for all words in the vocabulary, or just print the words that actually occur in the document.	
`a`	all
`s`	sparse, (default)
Print word counts as integers or as binary presence/absence indicators.	
`b`	binary
`i`	integer, (default)
How to indicate the word itself.	
`n`	integer word index
`w`	word string

c	combination of integer word index and word string, (default)
e	empty, don't print anything to indicate the identity of the word

Most of the other commands that are used to extract data from the model usually start with "--print" prefix, and are documented in the one line help. Some of the more useful commands are the switches that allow you to access the probability, entropy, and information gain of words in the classifiers. Also helpful are the switches that output the names of the documents. This is helpful when classifying documents, because you can split the documents into an explicit training and testing set.

Classifying Documents

Now that you understand how rainbow index's the documents in it's model, you will better grasp how rainbow utilizes this model for classification of documents. Statistics from a set of *training* documents will determine the parameters of the classifier; classification of a set of *testing* documents will be output. Note that rainbow does not look at the actual documents, but the tokenized model of the documents.

The `--test` switch performs a specified number of trials and prints the classifications of the documents in each trial's test-set to standard output. For example,

```
rainbow -d model --test-set=0.4 --test=3
```

will output the results of three trials. The --test-set switch will randomly select 40 percent of the documents for testing, and 60 percent of the documents are used for training for each of the three trials.

Classification results are printed as a series of text lines that consisting of the following fields:

```
directory/filename TrueClass TopPredictedClass:score1 2ndPredictedClass:score2 ...
```

The Perl script `rainbow-stats`, reads lines like this and outputs average accuracy, standard error, and a confusion matrix. Enter the following command:

```
rainbow -d model --test-set=0.4 --test=3 | rainbow-stats
```

Rainbow has different methods for classifying documents (e.g. K-nearest neighbor and Naive Bayes). The default is naïve bayes, however you can change this option with the --method switch. For example, the option for k-nearest neighbor is knn. Try this command:

```
rainbow -d model –method=knn --test-set=0.4 --test=3 | perl rainbow-stats
```

Reading Comprehension Questions

Modeling Newsgroup Text Documents

1. How does rainbow distinguish between two classes of data when constructing a model?
2. If not explicitly given, what is the default directory rainbow stores the classifier model indexing data?
3. What is the advantage/disadvantage of storing the model in a user defined directory?

4. When indexing a file, rainbow turns the file's stream of characters into tokens by a process called tokenization or "lexing". We used the --skip-header switch that avoids lexing news/mail headers by scanning forward until two newlines. Name and describe two other lexing options that one can utilize when constructing a model with rainbow.

Diagnostics of a Model

1. In the first document entry of print matrix output, all the words are numerated in order from zero to N, where N is the number of words in that document. However, in subsequent document entries the words seem to not follow that pattern. Explain. Give proof, from the output, for your explanation.
2. What is the command so that rainbow prints a sparse matrix, in which the word counts is represent as an integer, and the word itself is represented in integer format only?
3. What is the command to print the probabilities of all the words in the comp.graphics?

Classifying Documents

1. When using the --test-set switch, we utilized the percentage option. That is to say we gave it a decimal number that represented percentage of test-sets out of all the documents for each episode. Name two other options you can utilize for the --test-set switch.
2. What other classification methods has rainbow implemented?