Stat 133, Spr '08
Due Sat, Apr 18

Text mining the State of the Union Speeches

The goal of this homework is for you to analyze the "State of the Union" speeches from 1776 to 2008. To do this you will need to prepare the data in a form that is suitable for statistical analysis. In particular, you will be examining the words that each president used in his address and their frequency of use. With this information, you can compare the presidents and see how they differ across time and party.

The Data The data are available on the Web at

```
http://www.stat.berkeley.edu/users/nolan/stat133/data/
        stateoftheunion1790-2008.txt.zip
```

The first part of the file has "table of contents" information, and the speeches are delimited by lines of the form `***`. That is, the first speech starts with a `***` and the next speech starts with the next `***`.

Preparation Read the document into R: extract the name of the president who gave each speech and save it in a character vector, called Pres. Extract the date of the speech and save that as a POSIXct or POSIXlt value (hint use strptime) called speechDate. Finally, chop the speeches up into a list called speeches, where there is one element for each speech. Each speech should be a character vector, where each element of the vector is a character string corresponding to a sentence in the speech (not a line in the text file).

Explore Explore the speeches - compare the number of sentences in each speech, the average number of words in a sentence, the total number of words in the speech, and any other numerical summary of the speech. Does any president stand out? Do presidents from the same era give similar speeches? Produce one or two plots that show your findings.

Word Vectors For each president, create a word vector that counts the number of occurrences of each word that is used in any of the president's speeches. This vector should have the same format across presidents: there should an entry for every word that appeared in any state of the union speech, these entries should be in alphabetical order, and the value of the entry should be the number of times that the word appears in the president's speeches, i.e. the term frequency (or TF for short). It's a good idea to use the words as the names of the elements.

There are over 12,000 unique words, so think about "clever" ways to do this that use the fact that most of the counts will be 0. The table() function may prove helpful. Also, you will want to deal with capitalization, punctuation, etc. and stem the words to get their common form or base

(i.e. reduce running to run). To stem the words you will need the Rstem package, available on the `www.omegahat.org` site. The function wordStem() should do this for you.

Write a function called TF to create your term frequencies. The function should have two arguments: the character vector of all words for the given president's speeches and the collection of all the words from all speeches (allWords). The function should return a named integer vector that matches the length of allWords and has element names allWords. The vector contains the number of occurrences in the speech of each word in allWords. Also write a function called DF to create the document frequency. It should take in a matrix, where each column represents a set of Term Frequencies for a document./president (created by the TF function). The return value will be an integer vector that has the same length as allWords, where each element is the count of the number of documents that contain the corresponding words.

Distances between speeches We use the Shannon-Jensen (SJ) metric to compute the distance between the word vectors. SJ is defined in terms of Kullback-Leibler (KL) distance. I have provided you with a function to do this, called computeSJDistance(). It takes as input the Term Frequency matrix (check that you have the correct input format), the Document Frequency vector, and the allWords vector. The function returns a matrix of distances between each pair of speeches. Note also that there is a parameter to select which variant of the distance to use in the calculation.

Analysis At this point, you have what you need to analyze the speeches according to their word frequencies. Try using multi-dimensional scaling (the cmdscale function) and clustering (the hclust function). Produce a visualization of the results. Use labels, lines, color, etc. to make the plot as informative as possible. Do the presidents cluster by era more than by party or vice versa? Are there any anomalous presidents?

Turn In Write up at most one page on your findings in the exploratory and final analysis, including 2 exploratory plots and 2 analysis plots. Include all code in a plain text file. Your write up should contain six interesting observations about the presidents' speeches with historical background to help corroborate your findings.