Homework #1 (Due 9/25)

Statistics for Social Data

A. Initial Data Processing

For the exercises in this section, turn in your code, but do not report any output.

1. Process the Raw Data in Python. Write a Python script to extract the metadata and compute the word frequencies for all 85 *Federalist* papers from `federalist.json`. The script should create a directory called `paper`. Within the `paper` directory, the script should create a separate directory for each paper, named `01`, `02`, ..., `85`. The directory for each paper should contain the following files with metadata:

- `TEXT`: the text of the paper
- `ADDRESSEE`, `AUTHOR`, `SIGNATURE`, `TITLE`, and `VENUE`: one-line files with plain text containing the paper's metadata
- `words.csv`: a CSV file with the nonzero word counts in the paper

In addition, create one file in the `paper` directory called `words.txt` listing all of the unique words that appear in the corpus, with one word per line, in alphabetical order.

The directory structure should look like:

```
paper/
  01/
    ADDRESSEE
    AUTHOR
    SIGNATURE
    TEXT
    TITLE
    words.csv
  .
  .
  .
  85/
    ADDRESSEE
    AUTHOR
    SIGNATURE
    TEXT
    TITLE
    words.csv
  words.txt
```

The file `paper/01/AUTHOR` should just contain one line:

```
HAMILTON
```

The other metadata files should be similar (`TEXT` will have multiple lines).

The first few lines of `paper/01/words.csv` should look something like this:

```
word,freq
a,25
able,1
absurd,1
accident,1
accordingly,1
acknowledge,1
```

The first few lines of `paper/words.txt` should look something like this:

```
’s
’t
1
1648
1683
1685
1688
1706
1726
1774
```

Hint: Go through the “Computing Word Frequencies in Python” tutorial before attempting this exercise. You can re-use some of the code from that tutorial.

2. Read the Word Frequency Data into R. Use the following code on the next page to read the word frequencies into R. This code defines a function, `read.words`. When called from the parent directory of `paper`, this returns a sparse matrix with one row for each word and one column for each paper. The entries of the matrix store the word frequencies in each paper. Try to understand every line of the code.

```
require(Matrix)

read.words <- function() {
    # read in the word list
    words <- readLines(file.path(’paper’, ’words.txt’))

    # read in the word frequencies
    files <- list.dirs(’paper’, recursive=FALSE)
    freqs <- as.list(rep(NA, length(files)))
    for (j in seq_along(files)) {
      freqs[[j]] <- read.csv(file.path(files[j], ”words.csv”), row.names = 1)
    }

    # put the information in a sparse matrix
    is <- integer()
    js <- integer()
    xs <- numeric()

    for (j in seq_along(files)) {
        f <- freqs[[j]]
        is <- c(is, match(rownames(f), words))
        js <- c(js, rep(j, nrow(f)))
        xs <- c(xs, f$freq)
    }

    x <- spMatrix(length(words), length(files), is, js, xs)
    rownames(x) <- words
    colnames(x) <- sprintf(”P%02d”, seq_along(files))
    attr(x, ”paper”) <- seq_along(files)
    x
}
```

3. Read the Author Metadata into R. Write a function called `read.author` that returns a vector of length 85 with the authors of the federalist papers. It may be helpful to study the `read.words` function to accomplish this task.

Reproduce *one* of the authorship analyses from Mosteller and Wallace (1963). You will need to modify the analysis slightly, because you do not have any external writings available. **You can either do the exercises in Part B, or you can to the exercises in Part C.** You do not need to do both sets of analysis.

B. Weight-Rate Authorship Analysis

1. Training Set. Split the papers of known authorship into a "screening" and a "calibration set." Describe how you performed the split.

2. Words as Features. Either use the screening set to select a list of non-contextual words or use Mosteller and Wallace's list. Describe how you chose your list of words.

3. Discriminant Function. Fit a discriminant function via Fisher's Linear Discriminant Analysis or another method of your choosing.

4. Evaluation on Training Set. Evaluate the discriminant function on the screening set to verify that it separates Hamliton and Madison papers. Report the results.

5. Evaluate on Test Set. Use the discriminant function and the calibration set to assess how consistent the disputed papers are with the two authors' writing styles. Report the results.

6. Summarize your findings. Assess the strength of the evidence for the two authors on the disputed papers.

C. Bayesian Authorship Analysis

1. Words as Features. Either use the papers of known authorship to find a list of non-contextual words or use Mosteller and Wallace's list. Describe how you chose your list of words.

2. Posterior Estimates. Use the Poisson model with one choice of the Mosteller-Wallace prior to estimate the word rates for both authors, using the papers of known authorship (the training set).

3. Feature Selection. Possibly discard words whose estimated rates are similar between the two authors. If you discord words, indicate how you chose what words to discard, and how many words you discarded.

4. Evaluation on Training Set. Use the estimated word rates to estimate the mean posterior log likelihood ratios between the authors on the papers in the training set. Report the results.

5. Evaluation on Test Set. Do the same for the disputed papers.

6. Varying the Prior. Repeat the analysis for a few different choices of the hyperparameters (the "underlying constants" in Mosteller and Wallace's terminology).

7. Alternative Model (Optional). Repeat the analysis for the negative binomial model.

8. Summarize Your Findings. Assess the strength of the evidence for the two authors on the disputed papers.

D. Power-Law Behavior in Word Usage Rates

1. Power-Law Fit. Ignoring authorship information, compute the frequencies of occurrences of all unique words in *The Federalist* papers. Either use Cosma Shalizi's R code at `http://tuvalu.santafe.edu/~aaronc/powerlaws/` or write your own code to fit a discrete power law to the word frequencies. Report the estimates of α and x_{min} along with standard errors for these quantities.

2. Alternatives. Perform likelihood ratio tests of the discrete power law against the discretized exponential, the discretized log-normal, and the discrete stretch exponential distributions. Again, either write your own code or use Cosma Shalizi's R code.

3. Summarize Your Findings. Assess the strength of the evidence for power-law behavior in *Federalist* word occurrence frequencies.