Text Mining for Economics and Finance
Homework 1

Instructions: Work in groups of up to three if you choose. Write up a report on the answers to the following questions. Turn in the report during class on 20 April. For the questions below, use the State-of-the-Union database at `https://github.com/sekhansen/text-mining-tutorial`.

1. Perform the following pre-processing steps on your data:

   - Tokenize the data (for example using `http://www.nltk.org/api/nltk.tokenize.html`).
   - Remove non-alphabetic characters (for string processing in Python the documentation at `https://docs.python.org/2/library/string.html` might be useful).
   - Remove stopwords using a list of your choice.
   - Stem the data using the Porter stemmer (for example using `http://www.nltk.org/_modules/nltk/stem/porter.html`).
   - Compute the corpus-level tf-idf score for every term, and choose a cutoff below which to remove words.

   From your pre-processed data, form the document-term matrix, and use this for the remaining questions.

2. Perform the following analysis:

   (a) Identify a dictionary of interest to measure heterogeneity across addresses. You can use an existing one, or invent one of your own. (It should not be difficult to find dictionaries online; for example see `http://www3.nd.edu/~mcdonald/Word_Lists.html`).

   (b) Use your dictionary to provide a quantitative representation of each address. Note that you will need to stem the terms in the dictionary as you did to form the document-term matrix.

   (c) Find some time series of interest over some portion of the addresses (for example, whether the US is in recession, engaged in a major war, the average inflation rate, etc.) that you think might correlate with your quantitative representation, and compute the correlation. Is it the sign you expected? Is it significant?

(d) Now use the same dictionary, but compute the content of each document using term weighting as discussed in class. Do your answers to the previous question change if you use this alternative representation?

3. Generate the tf-idf-weighted document-term matrix S. Perform a singular value decomposition on it using numpy (`http://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.svd.html`), and retain a "reasonable" number of singular values (no more than a few hundred) to form the approximate matrix $\hat{S}$.

   Now compare the cosine similarity of documents using both S and $\hat{S}$ in an example of your choosing. Does latent semantic analysis appear to outperform the standard analysis? For example, you could look at the average cosine similarity within and across speeches made by Republicans and Democrats, and assess whether LSA provided a sharper distinction between political parties.

4. Write a program to compute the parameter estimates for the multinomial mixture model using the EM algorithm, beginning from arbitrary initial values. The formulas for these are in the class lecture slides.

   Apply your code from above to the State-of-the-Union data. Verify that after each iteration the observed data log-likelihood function (i.e. the last expression in slide 15 of lecture 2) increases. Note this property holds for any initial values for the parameters, so you can select whichever you like. A good rule-of-thumb is to set $\rho_k = 1/K$ and randomly draw $\boldsymbol{\beta}_k$, for example from a Dirichlet distribution.