Quantitative Text Analysis
Exercise 8: Unsupervised Scaling

30th July 2014, Essex Summer School

Kenneth Benoit and Paul Nulty

This exercise involves using automatic unsupervised document scaling using correspondence analysis and the Poisson scaling model in R. For texts we will use the 2010 Irish budget speech corpus that you have worked with on previous days. Note that this is also the set of speeches analyzed using Poisson scaling (and other methods) in Will Lowe and Kenneth Benoit (2013), "Validating Estimates of Latent Traits From Textual Data Using Human Judgment as a Benchmark," *Political Analysis* 21: 298313.

Instructions

1. Create a `dfm` from the 2010 Irish budget speeches, for working the next two questions. We will also tidy up the document names from their built-in filename settings.

   (a) Load the Irish budget speech corpus and take the subset for the year 2010.

   ```
   library(quanteda)
   data(iebudgets)
   iebudgets2010 <- subset(iebudgets, year=="2010")
   ```

   (b) Create a dfm from this called `ieDfm`, where you apply stemming.

   (c) To make the output nicer, we will rename the rows of `ieDfm`. To change them you should reassign `rownames(ieDfm)` to the format "`[speaker] ([party])`". You can get `speaker` and `party` from the data.frame returned by `getData(iebudgets2010)`, and use the `paste` command to put this together:

   ```
   with(getData(iebudgets2010), paste(speaker, " (", party, ")", sep=""))
   ```

   (d) Verify the new document names of your dfm using the `docs(ieDfm)` command.

2. Correspondence analysis in R

   (a) For the correspondence analysis, we will use an R package called `ca`. This package may not be installed on your machine — try to load it with `library(ca)`. If this doesn't work, install the package using `install.packages("ca")`, and then load it using the `library` command.

   (b) Perform a correspondence analysis on `ieDfm` using the `ca()` function, and assign this to a new object `ieCA`.

   (c) Plot the correspondence analysis in two dimensions. Look at the functionality of the `what=` argument to `plot.ca` and set the value so that you only plot the document positions.

   (d) Plot the document positions in one dimension. A useful plot for this is `dotchart`.

   ```
   # plot the speakers in one dimension, ordered
   dotchart(ieCA$rowcoord[order(ieCA$rowcoord[,1]),1],
            labels = ieCA$rownames[order(ieCA$rowcoord[,1])])
   ```

(e) The function `summary` can be called on a wide range of R objects. For the object returned by the `ca`, which we have named `ieCA`, it will return details of document and word weights used by the model. Execute a summary of this object, and inspect the output. Why is it so long? Is this useful?

3. Poisson scaling in R

   (a) To run the Poisson scaling ("wordfish"), we will use the implementation from the `austin` package. Install this package on your machine with this command:

   ```
   install.packages("austin", repos="http://R-Forge.R-project.org",
                    type="source", dependencies=TRUE)
   ```

   (b) To set the orientation of the estimation you will need to note which column is Joan Burton (opposition anchor from Labour) and which is Brian Lenihan (Finance Minister).

   (c) Estimate the wordfish model using the following command. (The column indices in the "`dir`" vector refer to 3 for Burton and 1 for Lenihan.)

   ```
   ieWF <- wordfish(ieDfm, dir=c(11,4))
   ```

   (d) Summarize the results, by document, using the `summary` method for the wordfish object `ieWF`.

   (e) Plot the results for the documents, using the `plot` method for the wordfish object `ieWF`.

   (f) To recreate a version of the "Eiffel tower" plot from Slapin and Proksch (2008) Figure 2, we will plot the $\hat{\psi}_j$ values against the $\hat{\beta}_j$ values.

   ```
   plot(ieWF$beta, ieWF$psi, type="n", xlab="Word weights", ylab="Word Fixed Effect")
   text(ieWF$beta, ieWF$psi, ieWF$words, col="grey50", cex=.6)
   ```

   (g) (Extra credit) On the previous plot, highlight some words in red, for instance "Christmas", "Fianna", and "bailout". Hint: You can use `grep` on `ieWF$words` to get the indexes of these words. Remember that they will have been stemmed and converted to lowercase.