8p-Vector Spaces

Ben Schmidt

3/19/2015

1. Basic re-productions on your own texts.

1. Cut and paste the term-document matrix function (`folderIntoTD`) from the handout: use it to read in a set of documents of your own. How big is your matrix?
2. Currently, the function is engineered to work with the SOTUs, and has a hand-coded cutoff of 1,000 words. (It appears in two places, and looks like this: `filter(sum(count)>1000,word!="")`. Change the function so that instead of being hard-coded, this an **argument** to the function called `wordCutoff`.
3. Then re-run `folderIntoTD` with an appropriate constraint so that you get about 100 columns in the matrix. Use the normalized version.

For the course blog:

4. Run your own Principal Components Analysis and plot them along the lines of the example or the Allison-Heuser pamphlet we read: as either a point chart using `shape` and `color` as aesthetics on geom_point or as a geom_label.
5. Look at the loadings for those results by examining your models `loadings`, and try to explain why PCA segregates in the way that it does.
6. (Optional) Before running PCA, try reducing the vocabulary down significantly. Drop out stopwords, for instance, or only look at capitalized words to capture names.
7. Choose a document you're particularly interested in: use the `dist` function to calculate what the documents it shares the closest profile with. Does this make sense? Are there signs of overfitting?

2. Expanding the domain

8. Principal components need not only be run on wordcounts: it can be done on anything with several different numeric dimensions. Try running it with a dataset whose columns are *not* word counts. **The easiest choice would be to go back and look at the populations in each census, treating rows as cities and columns as years.** A harder version would be population percent change, or something involving the library set. E-mail if you want some ideas. If you take the road less traveled, post it on the blog so we can talk about it in class next week.

If you choose cities, you will want to normalize. Instead of using the population in each year, use the *total percentage of the city's population that was there in year X. This shouldn't be too hard to do with `group_by` and `mutate`: if you're having trouble, e-mail the class.