Discovery, 1

Frank Edwards

10/23/2019

Text data: setup

```r
# Run me first install.packages(c('tm', 'SnowballC, wordcloud, modelr'))
library(tm)
library(SnowballC)
library(wordcloud)
library(modelr)
```

Load the federalist papers

```r
DIR_SOURCE <- system.file("extdata/federalist", package = "qss")
corpus.raw <- Corpus(DirSource(directory = DIR_SOURCE, pattern = "fp"))
corpus.raw
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 85
```

Federalist 1

```r
content(corpus.raw[[1]])
```

```
## [1] "AFTER an unequivocal experience of the inefficiency of the subsisting \n        federal government
```

Tidy it up

```r
corpus.prep <- tm_map(corpus.raw, content_transformer(tolower))
corpus.prep <- tm_map(corpus.prep, stripWhitespace)
corpus.prep <- tm_map(corpus.prep, removePunctuation)
corpus.prep <- tm_map(corpus.prep, removeNumbers)
```

Federalist 1 (post processing)

```r
content(corpus.prep[[1]])
```

```
## [1] "after an unequivocal experience of the inefficiency of the subsisting federal government you are c
```

Remove stopwords

```r
head(stopwords("english"))
```

```
## [1] "i"      "me"     "my"     "myself" "we"     "our"
```

```r
corpus <- tm_map(corpus.prep, removeWords, stopwords("english"))
```

Federalist 1 (post processing)

```r
content(corpus[[1]])
```

```
## [1] "  unequivocal experience   inefficiency   subsisting federal government   called upon  deliberate
```

Stem words into comparable terms

```r
corpus <- tm_map(corpus, stemDocument)
content(corpus[[1]])
```

```
## [1] "unequivoc experi ineffici subsist feder govern call upon deliber new constitut unit state america
```

Format into document-term matrix

Obtain measures of term-frequency, how often a given stem-term appears in a document or corpus.

```
dtm <- DocumentTermMatrix(corpus)
dtm.mat <- as.matrix(dtm)
dtm.mat[1:5, 1:5]
```

```
##           Terms
## Docs       abl absurd accid accord acknowledg
##   fp01.txt   1      1     1      1          1
##   fp02.txt   0      0     0      0          0
##   fp03.txt   2      0     0      1          2
##   fp04.txt   1      0     0      1          0
##   fp05.txt   0      0     0      0          0
```

Topic discovery

Topic discovery is a form of exploratory data analysis that looks at the frequency of terms used in a text or corpus to identify topics discussed in documents.

- Unsupervised learning approaches are agnostic, and look for patterns in the data without identified outcome or predictor variables. Examples: clustering, topic discovery
- Supervised learning uses theory or hypotheses to explore data looking for particular patterns across outcomes and predictors. Examples: regression

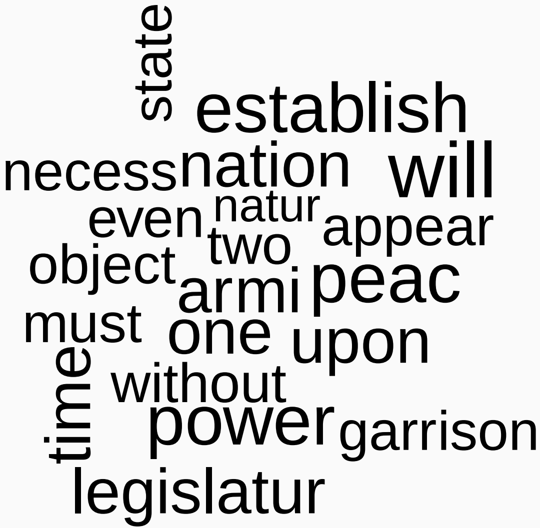
Assumptions: *texts are bags of words*

Topic discovery: wordcloud for Federalist 12

```
wordcloud(colnames(dtm.mat), dtm.mat[12, ], max.words = 20)
```

Topic discovery: wordcloud for Federalist 24

```r
wordcloud(colnames(dtm.mat), dtm.mat[24, ], max.words = 20)
```



Adjust for common words in corpus

The term frequency-inverse document frequency penalizes terms that occur frequently across documents in the corpus, highlighting novel terms in particular texts.

$$\mathrm{tf} - \mathrm{idf}(w, d) = \mathrm{tf}(w, d) \times \mathrm{idf}(w)$$

$$\mathrm{idf}(w) = log \left(\frac{n}{\mathrm{df}\,(w)} \right)$$

Weighted term frequency

Federalist 12

```r
dtm.tfidf <- as.matrix(weightTfIdf(dtm))
head(sort(dtm.tfidf[12, ], decreasing = TRUE), n = 10)
```

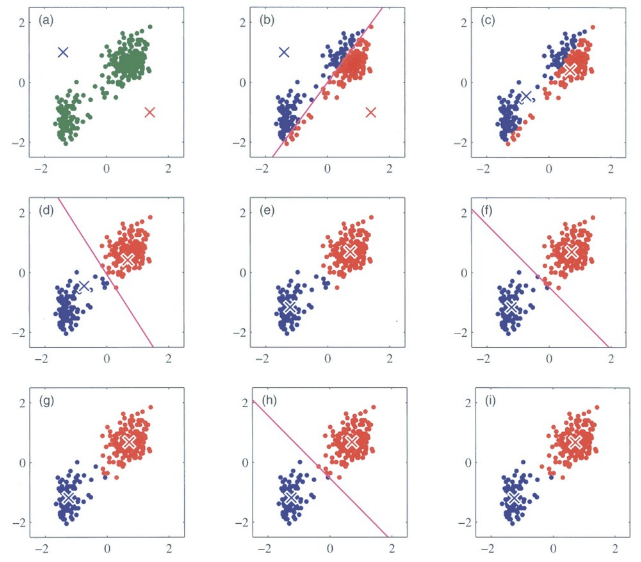
```
##     revenu contraband     patrol      excis      coast      trade
## 0.01905877 0.01886965 0.01886965 0.01876560 0.01592559 0.01473504
##        per        tax       cent     gallon
## 0.01420342 0.01295466 0.01257977 0.01257977
```

Federalist 24

```r
dtm.tfidf <- as.matrix(weightTfIdf(dtm))
head(sort(dtm.tfidf[24, ], decreasing = TRUE), n = 10)
```

```
##   garrison settlement   dockyard      spain       armi   frontier
## 0.02965511 0.01962294 0.01962294 0.01649040 0.01544256 0.01482756
##    arsenal    western       post     nearer
## 0.01308196 0.01306664 0.01236780 0.01166730
```

Remember the k-means algorithm?



We can use k-means to group texts by theme

K-means is an unsupervised approach to uncovering groupings in the data.

We can use it to learn and group texts by topic across the corpus.

```r
k <- 4  # clusters
# hamilton authored papers (known)
hamilton <- c(1, 6:9, 11:13, 15:17, 21:36, 59:61, 65:85)
dtm.tfidf.hammy <- dtm.tfidf[hamilton, ]
# k-means
km.out <- kmeans(dtm.tfidf.hammy, centers = k)
```

Evaluating clusters

```r
for (i in 1:k) {
    print("top words")
    print(head(sort(km.out$centers[i, ], decreasing = TRUE), n = 10))
    print(c("included texts in cluster", i))
    print(rownames(dtm.tfidf.hammy)[km.out$cluster == i])
}
```

```
## [1] "top words"
##     pardon    treason      guilt    clemenc     conniv      crime
## 0.04472060 0.02894567 0.02510566 0.02367348 0.02367348 0.01929712
##      impun      plead      sedit       weak
## 0.01788824 0.01673710 0.01492075 0.01470109
## [1] "included texts in cluster" "1"
## [1] "fp74.txt"
## [1] "top words"
##       senat      presid    governor     appoint       nomin      vacanc
## 0.019382349 0.015789668 0.009857989 0.009838966 0.009551661 0.009328505
##       offic      impeach        fill      treati
## 0.007941282 0.006589793 0.006552566 0.006460916
## [1] "included texts in cluster" "2"
## [1] "fp66.txt" "fp67.txt" "fp68.txt" "fp69.txt" "fp75.txt" "fp76.txt"
## [7] "fp77.txt" "fp79.txt"
## [1] "top words"
##        armi     militia    militari       navig   disciplin         war
## 0.011624485 0.011450433 0.008761049 0.005321748 0.004948897 0.004854514
##        peac    northern    frontier confederaci
## 0.004668017 0.004661314 0.004559462 0.004540867
## [1] "included texts in cluster" "3"
## [1] "fp06.txt" "fp08.txt" "fp11.txt" "fp13.txt" "fp24.txt" "fp25.txt"
```

Predicting authorship of federalist papers

66 essays have a known author (Hamilton or Madison). Authorship of the remaining 11 is unknown.

We will use writing style to predict who authored these 11 essays, based on word counts from the known-authorship essays.

```
dtm1 <- as.matrix(DocumentTermMatrix(corpus.prep))
### count words per 1,000
tfm <- dtm1/rowSums(dtm1) * 1000
## subset to words of interest
words <- c("although", "always", "commonly", "consequently", "considerable",
    "enough", "there", "upon", "while", "whilst")
tfm <- tfm[, words]
```

Calculate word usage by Hamilton, Madison

```
madison <- c(10, 14, 37:48, 58)
tfm.ave <- rbind(colSums(tfm[hamilton, ])/length(hamilton), colSums(tfm[madison,
    ])/length(madison))
tfm.ave
```

```
##        although    always  commonly consequently considerable    enough
## [1,] 0.01756975 0.7527744 0.2630876   0.02600857    0.5435127 0.3955031
## [2,] 0.27058809 0.2006710 0.0000000   0.44878468    0.1601669 0.0000000
##         there      upon     while      whilst
## [1,] 4.417750 4.3986828 0.3700484 0.007055719
## [2,] 1.113252 0.2000269 0.0000000 0.380113114
```

Hamilton likes there, upon. Madison likes whilst, consequently. We'll use these style differences to predict who wrote the unknown papers

Predict authorship for unknown papers

Fit a regression model with authorship as the outcome for known papers using distinctive words for each author as predictor. Then predict authorship for unknown papers.

```
author <- rep(NA, nrow(dtm1))
author[hamilton] <- TRUE
author[madison] <- FALSE
author_dat <- data.frame(author = author, tfm) %>% filter(!(is.na(author)))
```

Fit the model

Fit a model to known authorship papers

```r
m0 <- lm(author ~ upon + there + consequently + whilst, data = author_dat)
coef(m0)
```

```
##  (Intercept)         upon        there consequently       whilst
##   0.36855800   0.08338945   0.04746947  -0.22006171  -0.32937544
```

Recall that upon, there are Hamilton's words, and consequently, whilst are Madison's. Regression coefficients line up there.

Model fit

How well does the model fit the data?

- Evaluate total error of the model
  - Root Mean Square Error: $\sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$
  - Coefficient of determination: $R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
- Evaluate prediction performance

	Positive, obs.	Negative, obs.
Positive, pred.	**True positive**	False positive
Negative, pred.	False negative	**True negative**

How well did the model predict authors?

```
author_dat %>% mutate(yhat = fitted(m0) > 0.5) %>% group_by(author) %>% summarise(prop.true.positive = mea
    author))
```

```
## # A tibble: 2 x 2
##   author prop.true.positive
##   <lgl>               <dbl>
## 1 FALSE                   1
## 2 TRUE                    1
```

Perfection!

However, we may have *overfit* the data. Fitting your model to a particular dataset is no guarantee that it will predict well *out-of-sample*

Cross validation

We can hold out subsets of the data, re-fit the model, and check classification performance. This gives us a sense of how well our model performs at predicting new cases.

Leave-one-out cross validation: The algorithm

For each row of the data $i \cdots n$ 1. Remove ith row of the data 2. Fit the model to the remaining $n - 1$ observations 3. Predict the outcome for the held-out ith observation, calculate prediction error

Then, calculate the mean prediction error across all n observations.

Cross validation using loops

See Arnold for a tidyverse approach using modelr

```
n <- nrow(author_dat)
classify_out <- list()
for (i in 1:n) {
    sub.fit <- lm(author ~ upon + there + consequently + whilst, data = author_dat[-i,
        ])
    classify_out[[i]] <- predict(sub.fit, newdata = author_dat[i, ]) > 0.5
    classify_out[[i]] <- classify_out[[i]] == author_dat$author[[i]]
}
classify_out <- unlist(classify_out)
mean(classify_out)
```

```
## [1] 1
```

We still predict perfectly - this is a good model

How about those unidentified cases?

```
author <- rep(NA, nrow(dtm1))
author[hamilton] <- TRUE
author[madison] <- FALSE
author_dat <- data.frame(author = author, tfm) %>% filter((is.na(author)))

yhat <- predict(m0, newdata = author_dat)
## Hamilton is >0.5
yhat <- yhat > 0.5
table(yhat)
```

```
## yhat
## FALSE  TRUE
##    18     1
```

All but one attributed to Madison.

Lab, HW, next week

- Next week: network and spatial data
- Homework (two weeks on this one: due 10/30): Choose either text data, network data, or spatial data.
- Text data: 5.5.1, Network data 5.5.2, Spatial data 5.5.3
- Lab: more regression, review HW 4 and prediction from model