STA2201H Methods of Applied Statistics II

Monica Alexander

Week 11: Text analysis

Notes

- Presentations
- Assignment
- Research proposal

Text as data

- Increasing amount of digital text being recorded
- Increasing share of human interaction, communication and culture
- Information encoded in text is a complement to more traditional forms of data
- 'Content analysis'; nonreactive research

Examples

- Studying impact of new information on markets
- Social networks and information spread
- Changing political narratives over time, e.g. partisianship
- Changing themes in research

Text as data

- Inherently high dimensional
- Sample of documents, w words long, p possible words: unique representation is p^w
- Statistical methods relate to those used to study other high-dimensional data, sometimes adapted specifically to deal with text.

In general, we have raw text $\mathcal{D}$, and text analysis involved

1. Representing $\mathcal{D}$ as a numerical array $\boldsymbol{C}$
2. Map $\boldsymbol{C}$ to predicted values $\hat{\boldsymbol{V}}$ of unknown outcomes $\boldsymbol{V}$
3. Use $\hat{\boldsymbol{V}}$ in subsequent analysis (causal or otherwise)

Steps

1. Representing $\mathcal{D}$ as a numerical array $\boldsymbol{C}$

The elements of $\boldsymbol{C}$ are usually counts of *tokens*: words, phrases, etc.

2. Map $\boldsymbol{C}$ to predicted values $\hat{\boldsymbol{V}}$ of unknown outcomes $\boldsymbol{V}$

$\boldsymbol{V}$ could be, for example

- sentiment
- whether or not an email is spam

3. Use $\hat{\boldsymbol{V}}$ in subsequent analysis (causal or otherwise)

- Most commonly prediction
- But more and more in social science we are interested in causal interpretations

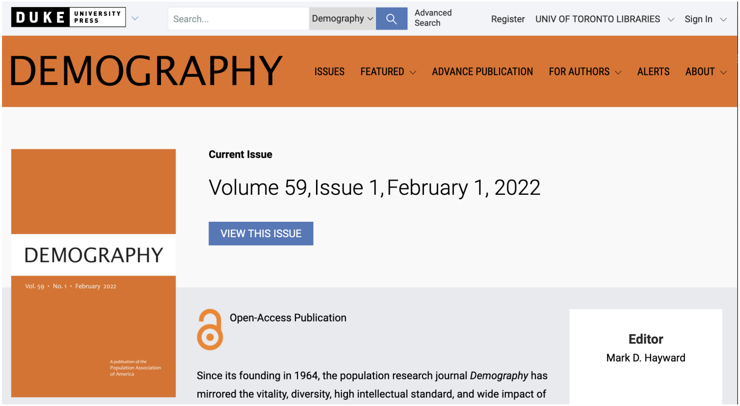
Today

- Representing text as data
- Feature selection
  - n-grams
  - tf-idf
- Dictionary-based methods (sentiment analysis)
- Topic models (LDA)

Examples


RADIOHEAD
THE KING OF LIMBS
IN/ RAINBOWS
IN RAIN/BOWS
IN RAINBOWS/
IN RAIN_BOWS
RADIOHEAD
AMNESIAC
OK COMPUTER
RADIOHEAD
radiohead
PABLO HONEY
RADIOHEAD
KID A
RADIOHEAD
the bends

Examples


DUKE UNIVERSITY PRESS
Search...
Demography
Advanced Search
Register
UNIV OF TORONTO LIBRARIES
Sign In
DEMOGRAPHY
ISSUES
FEATURED
ADVANCE PUBLICATION
FOR AUTHORS
ALERTS
ABOUT
DEMOGRAPHY
Vol. 59 • No. 1 • February 2022
A publication of the Population Association of America
Current Issue
Volume 59, Issue 1, February 1, 2022
VIEW THIS ISSUE
Open-Access Publication
Since its founding in 1964, the population research journal Demography has
mirrored the vitality, diversity, high intellectual standard, and wide impact of
Editor
Mark D. Hayward

Representing text as data

- Text is incredibly complex
- Structure, interpretation, grammar
- Much like any other model, we need to simplify to make the problem tractable
- Three main simplifications
  - dividing the text into individual documents
  - reducing the number of language elements we consider
  - limiting the extent to which we encode dependence among elements within documents

The result is mapping raw text $\mathcal{D}$ as a numerical array $\boldsymbol{C}$. A row $\boldsymbol{c}_i$ of $\boldsymbol{C}$ is a numerical vector with each element indicating the presence or count of a particular language token in document i.

What is a document?

- The first step is to divide the raw text into individual documents
- In many applications, this is governed by the level at which attributes of interest $\boldsymbol{V}$ are defined
- Examples:
  - Radiohead: songs
  - Demography: articles
  - Hansard: ?

Example

```
## # A tibble: 2,482 x 4
##    line                                           song_name album_name   year
##    <chr>                                          <chr>     <fct>       <dbl>
##  1 How come I end up where I started?             15 Step   In Rainbows  2007
##  2 How come I end up where I went wrong?          15 Step   In Rainbows  2007
##  3 Won't take my eyes off the ball again          15 Step   In Rainbows  2007
##  4 You reel me out, then you cut the string       15 Step   In Rainbows  2007
##  5 How come I end up where I started?             15 Step   In Rainbows  2007
##  6 How come I end up where I went wrong?          15 Step   In Rainbows  2007
##  7 Won't take my eyes off the ball again          15 Step   In Rainbows  2007
##  8 First you reel me out and then you cut the string 15 Step   In Rainbows  2007
##  9 You used to be alright                         15 Step   In Rainbows  2007
## 10 What happened?                                 15 Step   In Rainbows  2007
## # ... with 2,472 more rows
```

Tokens and unnesting

- What should a token be? Words, phrases or sentences are common
- For example, let's consider words as our tokens
- We need to unnest the tokens from raw text
- (Easy to do in R with `tidytext` package)

```
lyrics_tidy <- lyrics %>%
  unnest_tokens(word, line)
lyrics_tidy
```

```
## # A tibble: 13,562 x 4
##    song_name album_name   year word
##    <chr>     <fct>       <dbl> <chr>
##  1 15 Step   In Rainbows  2007 how
##  2 15 Step   In Rainbows  2007 come
##  3 15 Step   In Rainbows  2007 i
##  4 15 Step   In Rainbows  2007 end
##  5 15 Step   In Rainbows  2007 up
##  6 15 Step   In Rainbows  2007 where
##  7 15 Step   In Rainbows  2007 i
##  8 15 Step   In Rainbows  2007 started
##  9 15 Step   In Rainbows  2007 how
## 10 15 Step   In Rainbows  2007 come
## # ... with 13,552 more rows
```

Removing stop-words

- common to remove a subset of words that are very common.
- Very common words, often called 'stop words', include articles ('the,' 'a'), conjunctions ('and,' 'or'), forms of the verb 'to be,' and so on.
- These words are important to the grammatical structure of sentences, but they typically convey relatively little meaning on their own.

```
data("stop_words")
stop_words
```

```
## # A tibble: 1,149 x 2
##    word        lexicon
##    <chr>       <chr>
##  1 a           SMART
##  2 a's         SMART
##  3 able        SMART
##  4 about       SMART
##  5 above       SMART
##  6 according   SMART
##  7 accordingly SMART
##  8 across      SMART
##  9 actually    SMART
## 10 after       SMART
## # ... with 1,139 more rows
```

Removing stop-words

```
lyrics_tidy <- lyrics_tidy %>%
  anti_join(stop_words %>% filter(lexicon=="snowball"))
lyrics_tidy
```

```
## # A tibble: 6,421 x 4
##    song_name album_name   year word
##    <chr>     <fct>       <dbl> <chr>
##  1 15 Step   In Rainbows  2007 come
##  2 15 Step   In Rainbows  2007 end
##  3 15 Step   In Rainbows  2007 started
##  4 15 Step   In Rainbows  2007 come
##  5 15 Step   In Rainbows  2007 end
##  6 15 Step   In Rainbows  2007 went
##  7 15 Step   In Rainbows  2007 wrong
##  8 15 Step   In Rainbows  2007 take
##  9 15 Step   In Rainbows  2007 eyes
## 10 15 Step   In Rainbows  2007 ball
## # ... with 6,411 more rows
```

Stemming

Another step that is commonly used to reduce the feature space is stemming: replacing words with their root such that,
e.g. "economic," "economics," "economically" are all replaced by the stem "economic."

tf-idf

For every word within a document, we can calculate the term frequency-inverse document frequency (tf-idf)

- For a word or other feature j in document i, term frequency tf_{ij} is the count $c_i j$ of occurrences of j in i.
- Inverse document frequency (idf_j) is

$$\log(n/d_j)$$

  where $d_j = \sum_i \mathbf{1}_{[c_{ij}>0]}$ and n is the total number of documents.
- tf-idf if the product of these two quantities
- Common to also remove words that are below a tf-idf threshold

Frequencies

```
song_words <- lyrics_tidy %>%
  group_by(song_name, album_name, year, word) %>%
  tally() %>%
  arrange(song_name, -n) %>%
  group_by(song_name, album_name, year) %>%
  mutate(total_words = sum(n))

song_words
```

```
## # A tibble: 3,348 x 6
## # Groups:   song_name, album_name, year [96]
##    song_name    album_name  year word            n total_words
##    <chr>        <fct>      <dbl> <chr>       <int>       <int>
##  1 (Nice Dream) The Bends   1995 dream          17          69
##  2 (Nice Dream) The Bends   1995 nice           17          69
##  3 (Nice Dream) The Bends   1995 enough          4          69
##  4 (Nice Dream) The Bends   1995 think           4          69
##  5 (Nice Dream) The Bends   1995 belong          2          69
##  6 (Nice Dream) The Bends   1995 love            2          69
##  7 (Nice Dream) The Bends   1995 strong          2          69
##  8 (Nice Dream) The Bends   1995 angel           1          69
##  9 (Nice Dream) The Bends   1995 answerphone     1          69
## 10 (Nice Dream) The Bends   1995 brother         1          69
## # ... with 3,338 more rows
```

tf-idf

```r
song_words %>%
  bind_tf_idf(word, song_name, n) %>%
  select(song_name:word, tf:tf_idf) %>%
  arrange(-tf_idf)
```

```
## # A tibble: 3,348 x 7
## # Groups:   song_name, album_name, year [96]
##    song_name                album_name          year word        tf   idf tf_idf
##    <chr>                    <fct>              <dbl> <chr>    <dbl> <dbl>  <dbl>
##  1 Sit Down. Stand Up       Hail to the Thief   2003 raindro~ 0.608  4.56   2.77
##  2 Feral                    The King of Limbs   2011 judge    0.429  4.56   1.96
##  3 Give Up the Ghost        The King of Limbs   2011 hurt     0.507  3.47   1.76
##  4 Pulk/Pull Revolving Doors Amnesiac           2001 doors    0.385  3.87   1.49
##  5 The National Anthem      Kid A               2000 holding  0.368  3.87   1.43
##  6 Ripcord                  Pablo Honey         1993 ripcord  0.273  4.56   1.24
##  7 The National Anthem      Kid A               2000 everyone 0.316  3.87   1.22
##  8 The Tourist              OK Computer         1997 slow     0.261  4.56   1.19
##  9 Prove Yourself           Pablo Honey         1993 prove    0.255  4.56   1.16
## 10 Give Up the Ghost        The King of Limbs   2011 arms     0.333  3.47   1.16
## # ... with 3,338 more rows
```

n-grams

- Producing a tractable representation also requires that we limit dependence among language elements.
- The simplest and most common way to represent a document is called **bag-of-words**. The order of words is ignored altogether, and $\boldsymbol{c}_i$ is a vector whose length is equal to the number of words in the vocabulary and whose elements c_{ij} are the number of times word j occurs in document i
- This scheme can be extended to encode a limited amount of dependence by counting unique phrases rather than unique words
- These are called *n*-grams
- For example, bi-grams are two word phrases

bigrams

```r
bigrams <- lyrics %>% unnest_tokens(bigram, line, token = "ngrams", n = 2)
bigrams
```

```
## # A tibble: 11,126 x 4
##    song_name album_name   year bigram
##    <chr>     <fct>       <dbl> <chr>
##  1 15 Step   In Rainbows  2007 how come
##  2 15 Step   In Rainbows  2007 come i
##  3 15 Step   In Rainbows  2007 i end
##  4 15 Step   In Rainbows  2007 end up
##  5 15 Step   In Rainbows  2007 up where
##  6 15 Step   In Rainbows  2007 where i
##  7 15 Step   In Rainbows  2007 i started
##  8 15 Step   In Rainbows  2007 how come
##  9 15 Step   In Rainbows  2007 come i
## 10 15 Step   In Rainbows  2007 i end
## # ... with 11,116 more rows
```

bigrams

```r
bigrams %>%
  count(bigram, sort = TRUE) %>%
  drop_na()
```

```
## # A tibble: 4,923 x 2
##    bigram             n
##    <chr>          <int>
##  1 no no            198
##  2 in the            64
##  3 the raindrops     48
##  4 you can           47
##  5 in a              39
##  6 don't hurt        38
##  7 hurt me           38
##  8 all the           35
##  9 and the           32
## 10 on the            31
## # ... with 4,913 more rows
```

bigrams

Removing any with stop words

```
bigrams_separated <- bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ") %>%
  filter(bigram!="NA NA")
```

Investigating bigrams

We can calculate the tf-idf for bigrams

```
bigram_tf_idf <- bigrams_united %>%
  count(song_name, bigram) %>%
  bind_tf_idf(bigram, song_name, n) %>%
  filter(n>1, tf<1) %>%
  arrange(desc(tf_idf))
bigram_tf_idf %>%
  select(song_name,bigram, tf_idf)
```

```
## # A tibble: 142 x 3
##    song_name            bigram            tf_idf
##    <chr>                <chr>              <dbl>
##  1 House of Cards       denial denial       3.73
##  2 Weird Fishes/Arpeggi weird fishes        3.58
##  3 2 + 2 = 5            paying attention    3.41
##  4 The Tourist          idiot slow          3.36
##  5 Morning Bell         walking walking     2.98
##  6 Reckoner             blank shore         2.98
##  7 Nude                 gonna happen        2.52
##  8 Identikit            broken hearts       2.44
##  9 Ful Stop             foul tasting        2.24
## 10 Ful Stop             tasting medicine    2.24
## # ... with 132 more rows
```

Statistical methods

Statistical methods

- We are interested in mapping the document-token matrix $\boldsymbol{C}$ to predictions of an attribute $\boldsymbol{V}$.
- In some cases, the observed data is partitioned into submatrices $\boldsymbol{C}_{\text{train}}$ and $\boldsymbol{C}_{\text{test}}$, where the matrix $\boldsymbol{C}_{\text{train}}$ collects rows for which we have observations $\boldsymbol{V}_{\text{train}}$ of $\boldsymbol{V}$ and the matrix $\boldsymbol{C}_{\text{test}}$ collects rows for which $\boldsymbol{V}$ is unobserved.
- Attributes in $\boldsymbol{V}$ can include observable quantities such as the frequency of flu cases, the positive or negative rating of album or song reviews, or the unemployment rate, for which the documents are informative.
- There can also be latent attributes of interest, such as the topics being discussed in a debate or in news articles.

Method can be divided into different categories: dictionary methods, regression methods, and generative models.

Dictionary methods

- Dictionary-based methods, do not involve statistical inference at all: they simply specify $\hat{\mathbf{v}}_i = f(\mathbf{c}_i)$ for some known function $f(\cdot)$.
- This is by far the most common method in the social science literature.
- In some cases, researchers define $f(\cdot)$ based on a prespecified dictionary of terms capturing particular categories of text

Sentiment analysis

- Outcome of interest $\mathbf{v}_i$ is latent sentiment
- Function $f(\cdot)$ defined using a pre-specified dictionary that relates words to particular categories of sentiment

```r
get_sentiments("afinn")
```

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # ... with 2,467 more rows
```

```r
unique(get_sentiments("afinn")$value)
```

```
##  [1] -2 -3  2  1 -1  3  4 -4 -5  5  0
```

An alternative dictionary

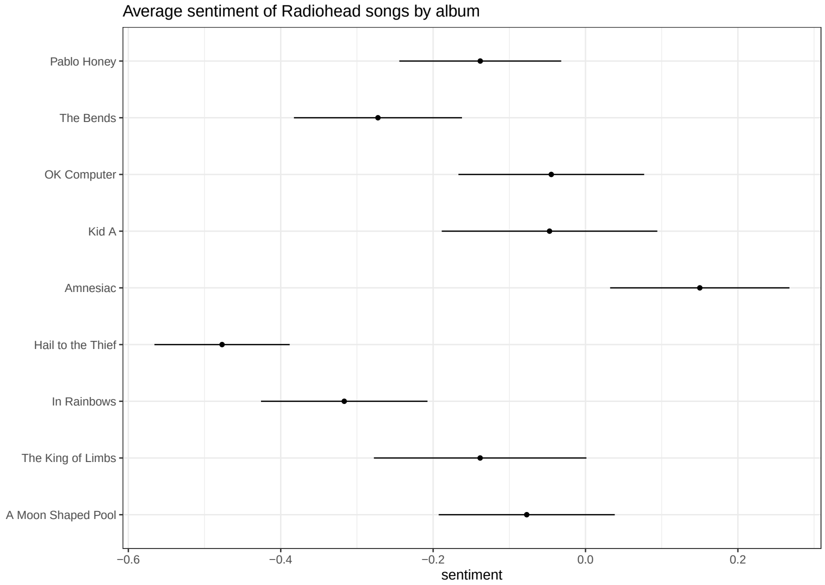
```r
get_sentiments("bing")
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # ... with 6,776 more rows
```

```r
unique(get_sentiments("nrc")$sentiment)
```

```
##  [1] "trust"        "fear"         "negative"     "sadness"      "anger"
##  [6] "surprise"     "positive"     "disgust"      "joy"          "anticipation"
```

Sentiment analysis example



Topic models

Generative models

- Generative models: we begin with a model of $p\left(\mathbf{c}_i \mid \mathbf{v}_i\right)$
- Useful when thinking about causal chain of language and outcomes
- e.g. Congresspeople's ideology is not determined by their use of partisan language; rather, people who are more conservative or liberal to begin with are more likely to use such language.

Topic models

- Unsupervised generative models: we do not observe the true value of $\mathbf{v}_i$, but impose sufficient structure to allow $\mathbf{v}_i$ to be inferred

Topic models assume that a document is a realization of a mixture of latent topics. The topics themselves are represented by a set of words that are selected from that topic.

- E.g. a politician first chooses the topics they want to speak about. After choosing the topics, the politician then chooses appropriate words to use for each of those topics.
- We are trying to model this generative process, and estimate the underlying topics
- Each document as a mixture of topics, and each topic as a mixture of words.
- Statistically, topic models consider each document as having been generated by some probability distribution over topics. Similarly, each topic is considered a probability distribution over words/terms

Latent Dirichlet Allocation

Blei, Ng, and Jordan (2003). Probably the most commonly used topic model. Assumes every document is generated independently based on fixed hyperparameters. For document m, the topic distribution θ_m is assumed to be

$$\theta_m \sim \text{ Dirichlet } (\alpha)$$

For topic k the distribution of terms is also dirichlet

$$\beta_k \sim \text{ Dirichlet } (\eta)$$

Then we assume a topic for a particular term/word n in document d are categorical

$$z_{m,n} \sim \text{Categorical}(\theta_m)$$

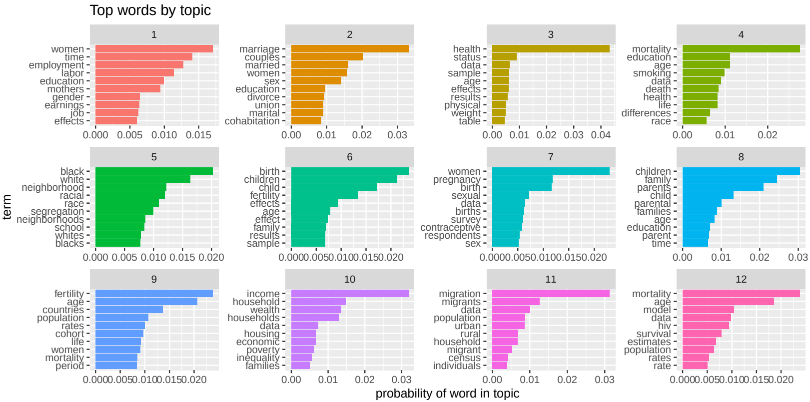
and then a term/word for a particular topic in document d is

$$w_{m,n} \sim \text{Categorical}(\beta_{z_{m,n}})$$

Fit in Bayesian framework

- Can fit using MCMC in Stan as a mixture model
- See here: https://mc-stan.org/docs/2_18/stan-users-guide/latent-dirichlet-allocation.html
- In practice, hard to do, especially with a small corpus

Demography topics



Demography topics

1. women and work
2. marriage, cohabitation, divorce
3. health
4. differentials in mortality
5. neighbourhood effects and segregation
6. fertility and family
7. family planning
8. children and time use
9. fertility (formal demog)
10. economic demography
11. migration
12. mortality (formal demog)

Top papers in each topic

where “top” is highest probability that the document contains that topic

1. “The Motherhood Penalty in Context: Assessing Discrimination in a Polarized Labor Market.” (keywords: Motherhood, Employment, Discrimination, Inequality)
2. " Same-Sex and Different-Sex Cohabiting Couple Relationship Stability." (Union stability, LGBT, Cohabitation, Marriage)
3. “Health Measurement in Population Surveys: Combining Information from Self-reported and Observer-Measured Health Indicators.” (Health measurement, Self-rated health, Biomarkers, Measurement error, Socioeconomic position)
4. “Population Composition, Public Policy, and the Genetics of Smoking.” (Smoking, Genetics, Gene-environment interaction, Policy)
5. “The Determinants of Neighborhood Satisfaction: Racial Proxy Revisited.” (Segregation, Residential preferences, Neighborhood satisfaction)

Top papers continued

6. "Preference for Boys, Family Size, and Educational Attainment in India" (Quantity-quality trade-off, Education, Family size, India)
7. "The Relationship History Calendar: Improving the Scope and Quality of Data on Youth Sexual Behavior." (Survey methodology, Sexual behavior, Condom use, Life history calendar, Data collection)
8. "Family Structure Experiences and Child Socioemotional Development During the First Nine Years of Life: Examining Heterogeneity by Family Structure at Birth" (Family structure, Family instability, Fragile Families and Child Wellbeing Study, Repartnering, Child well-being)

Continued

9. “On Nonstable and Stable Population Momentum.” (Population momentum, Age distribution, Decomposition)
10. “Decomposing the Decline of Cash Assistance in the United States, 1993 to 2016.” (Poverty, Social policy, Welfare state, Cash assistance, TANF)
11. “Recovery Migration After Hurricanes Katrina and Rita: Spatial Concentration and Intensification in the Migration System.” (Recovery migration, Migration system, Environment, Disasters, Hurricanes Katrina and Rita)
12. “Estimating Adult Death Rates From Sibling Histories: A Network Approach.” (Networks, Mortality, Demographic and health surveys, Sampling)

Topic models

- Lots of assumptions
- Can relax independence assumption (CTM)
- Can include covariates (STM)
- Not always useful
- Maybe some interesting science of science stuff?
- Propagation of uncertainty hard