Stat.653 Project

Nurahmet Maimaitiyiming

May 1st,2020

### I have collected a set of documents that definitely relate to four separate topics, then perform topic modeling to see whether the algorithm can correctly distinguish the four groups.

## Books selected

* *Birds Every Child Should Know* by Neltje Blanchan (Animal Category)
* *Pride and Prejudice* by by Jane Austen
* *Houses and House-Life of the American Aborigines* by Lewis Henry Morgan (History Category)
* *Musicians of To-Day*  by Romain Rolland (Music Category)

## Retrieving the text of these four books using the gutenbergr package

titles <- c("Birds Every Child Should Know", "Pride and Prejudice",  
 "Houses and House-Life of the American Aborigines", "Musicians of To-Day")

library(gutenbergr)  
library(ggplot2)  
library(tidyverse)

## -- Attaching packages ------------------------------------------------- tidyverse 1.3.0 --

## v tibble 3.0.0 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0  
## v purrr 0.3.3

## -- Conflicts ---------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

books<- gutenberg\_works(title %in% titles) %>%  
 gutenberg\_download(meta\_fields = "title")

## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest

## Using mirror http://aleph.gutenberg.org

As pre-processing, divided these into chapters, use tidytext’s unnest\_tokens() to separate them into words, then remove stop\_words. We’re treating every chapter as a separate “document”, each with a name like Birds Every Child Should Know\_1 or Musicians of To-Day\_11.

library(stringr)  
library(dplyr)  
library(tidyr)  
library(tidytext)  
  
# divide into documents, each representing one chapter  
by\_chapter <-books %>%  
 group\_by(title) %>%  
 mutate(chapter = cumsum(str\_detect(text, regex("^chapter ", ignore\_case = TRUE)))) %>%  
 ungroup() %>%  
 filter(chapter > 0) %>%  
 unite(document, title, chapter)  
  
# split into words  
by\_chapter\_word <- by\_chapter %>%  
 unnest\_tokens(word, text)  
  
# find document-word counts  
word\_counts <- by\_chapter\_word %>%  
 anti\_join(stop\_words) %>%  
 count(document, word, sort = TRUE) %>%  
 ungroup()

## Joining, by = "word"

word\_counts

## # A tibble: 69,284 x 3  
## document word n  
## <chr> <chr> <int>  
## 1 Musicians of To-Day\_1 music 519  
## 2 Musicians of To-Day\_1 art 242  
## 3 Musicians of To-Day\_1 footnote 238  
## 4 Musicians of To-Day\_1 musical 196  
## 5 Musicians of To-Day\_1 french 176  
## 6 Musicians of To-Day\_1 time 167  
## 7 Musicians of To-Day\_1 life 162  
## 8 Musicians of To-Day\_1 de 153  
## 9 Musicians of To-Day\_1 wagner 152  
## 10 Houses and House-Life of the American Aborigines\_12 tribes 146  
## # ... with 69,274 more rows

### LDA on chapters

Right now data frame word\_counts is in a tidy form, with one-term-per-document-per-row, but the topicmodels package requires a DocumentTermMatrix.We can cast a one-token-per-row table into a DocumentTermMatrix with tidytext’s cast\_dtm().

library(topicmodels)  
chapters\_dtm <- word\_counts %>%  
 cast\_dtm(document, word, n)  
  
chapters\_dtm

## <<DocumentTermMatrix (documents: 104, terms: 18831)>>  
## Non-/sparse entries: 69284/1889140  
## Sparsity : 96%  
## Maximal term length: 23  
## Weighting : term frequency (tf)

Useing the LDA() function to create a four-topic model. In this case we know we’re looking for four topics because there are four books; in other problems we may need to try a few different values of k.

chapters\_lda <- LDA(chapters\_dtm, k = 4, control = list(seed = 1234))  
chapters\_lda

## A LDA\_VEM topic model with 4 topics.

We can examine per-topic-per-word probabilities.

chapter\_topics <- tidy(chapters\_lda, matrix = "beta")  
chapter\_topics

## # A tibble: 75,324 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 music 1.61e- 2  
## 2 2 music 5.88e- 4  
## 3 3 music 3.22e- 4  
## 4 4 music 6.47e- 5  
## 5 1 art 7.52e- 3  
## 6 2 art 3.72e-44  
## 7 3 art 8.06e- 5  
## 8 4 art 5.09e- 4  
## 9 1 footnote 7.39e- 3  
## 10 2 footnote 1.88e-86  
## # ... with 75,314 more rows

Notice that this has turned the model into a one-topic-per-term-per-row format. For each combination, the model computes the probability of that term being generated from that topic.

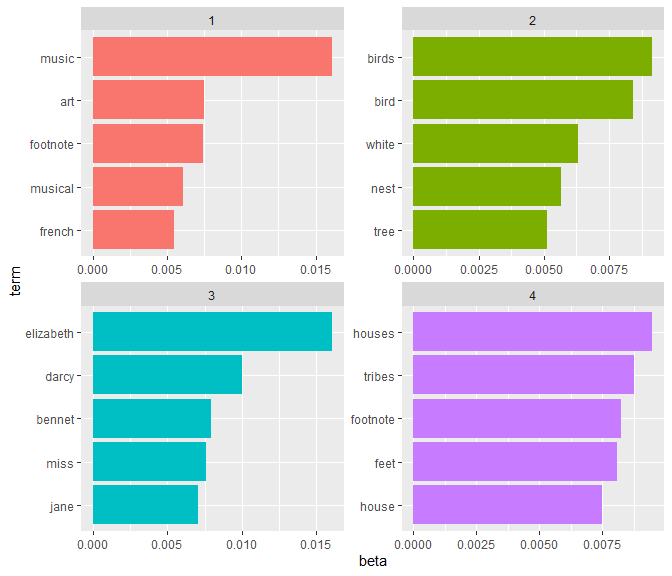
We could use dplyr’s top\_n() to find the top 5 terms within each topic.

top\_terms <- chapter\_topics %>%  
 group\_by(topic) %>%  
 top\_n(5, beta) %>%  
 ungroup() %>%  
 arrange(topic, -beta)  
  
top\_terms

## # A tibble: 20 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 music 0.0161   
## 2 1 art 0.00752  
## 3 1 footnote 0.00739  
## 4 1 musical 0.00608  
## 5 1 french 0.00546  
## 6 2 birds 0.00916  
## 7 2 bird 0.00840  
## 8 2 white 0.00630  
## 9 2 nest 0.00567  
## 10 2 tree 0.00513  
## 11 3 elizabeth 0.0160   
## 12 3 darcy 0.0100   
## 13 3 bennet 0.00789  
## 14 3 miss 0.00760  
## 15 3 jane 0.00706  
## 16 4 houses 0.00950  
## 17 4 tribes 0.00876  
## 18 4 footnote 0.00824  
## 19 4 feet 0.00809  
## 20 4 house 0.00751

## visualizing the top 5 terms with ggplot2

library(ggplot2)  
  
top\_terms %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(term, beta, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free") +  
 coord\_flip() +  
 scale\_x\_reordered()



The terms that are most common within each topic

### Per-document classification

Each document in this analysis represented a single chapter. Thus, we may want to know which topics are associated with each document. Can we put the chapters back together in the correct books? We can find this by examining the per-document-per-topic probabilities, (“gamma”).

chapters\_gamma <- tidy(chapters\_lda, matrix = "gamma")  
chapters\_gamma

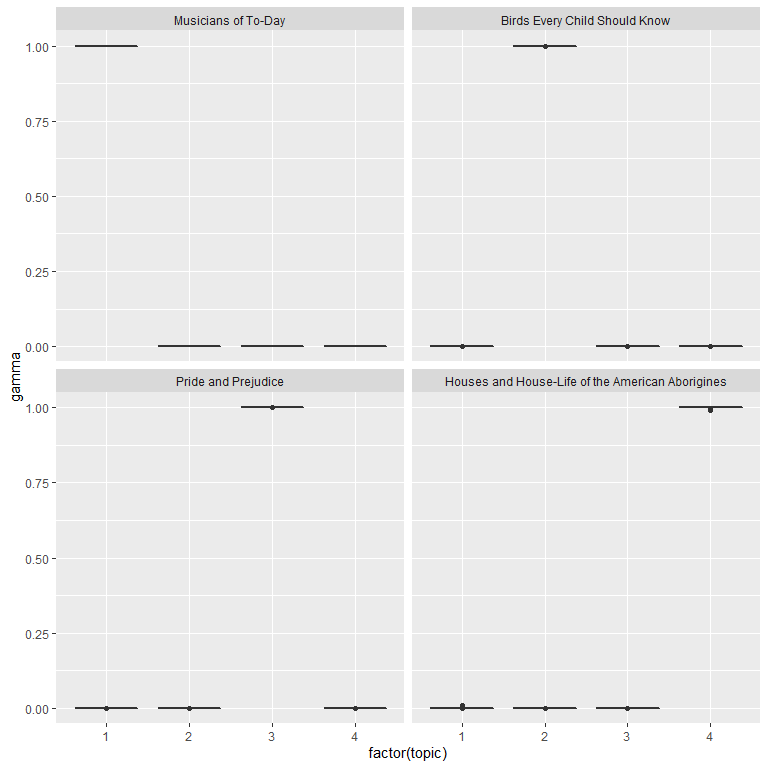
## # A tibble: 416 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 Musicians of To-Day\_1 1 1.00   
## 2 Houses and House-Life of the American Aborigines\_12 1 0.00000167  
## 3 Houses and House-Life of the American Aborigines\_19 1 0.00307   
## 4 Houses and House-Life of the American Aborigines\_15 1 0.00000202  
## 5 Houses and House-Life of the American Aborigines\_20 1 0.00000286  
## 6 Birds Every Child Should Know\_9 1 0.00000390  
## 7 Houses and House-Life of the American Aborigines\_18 1 0.00862   
## 8 Houses and House-Life of the American Aborigines\_16 1 0.00000260  
## 9 Houses and House-Life of the American Aborigines\_23 1 0.00000208  
## 10 Houses and House-Life of the American Aborigines\_14 1 0.00000421  
## # ... with 406 more rows

First we re-separate the document name into title and chapter, after which we can visualize the per-document-per-topic probability for each

chapters\_gamma <- chapters\_gamma %>%  
 separate(document, c("title", "chapter"), sep = "\_", convert = TRUE)  
  
chapters\_gamma

## # A tibble: 416 x 4  
## title chapter topic gamma  
## <chr> <int> <int> <dbl>  
## 1 Musicians of To-Day 1 1 1.00   
## 2 Houses and House-Life of the American Aborigines 12 1 0.00000167  
## 3 Houses and House-Life of the American Aborigines 19 1 0.00307   
## 4 Houses and House-Life of the American Aborigines 15 1 0.00000202  
## 5 Houses and House-Life of the American Aborigines 20 1 0.00000286  
## 6 Birds Every Child Should Know 9 1 0.00000390  
## 7 Houses and House-Life of the American Aborigines 18 1 0.00862   
## 8 Houses and House-Life of the American Aborigines 16 1 0.00000260  
## 9 Houses and House-Life of the American Aborigines 23 1 0.00000208  
## 10 Houses and House-Life of the American Aborigines 14 1 0.00000421  
## # ... with 406 more rows

# reorder titles in order of topic 1, topic 2, etc before plotting  
chapters\_gamma %>%  
 mutate(title = reorder(title, gamma \* topic)) %>%  
 ggplot(aes(factor(topic), gamma)) +  
 geom\_boxplot() +  
 facet\_wrap(~ title)



The gamma probabilities for each chapter within each book

chapter\_classifications <- chapters\_gamma %>%  
 group\_by(title, chapter) %>%  
 top\_n(1, gamma) %>%  
 ungroup()  
  
chapter\_classifications

## # A tibble: 104 x 4  
## title chapter topic gamma  
## <chr> <int> <int> <dbl>  
## 1 Musicians of To-Day 1 1 1.00  
## 2 Birds Every Child Should Know 9 2 1.00  
## 3 Birds Every Child Should Know 16 2 1.00  
## 4 Birds Every Child Should Know 1 2 1.00  
## 5 Birds Every Child Should Know 4 2 1.00  
## 6 Birds Every Child Should Know 8 2 1.00  
## 7 Birds Every Child Should Know 14 2 1.00  
## 8 Birds Every Child Should Know 2 2 1.00  
## 9 Birds Every Child Should Know 5 2 1.00  
## 10 Birds Every Child Should Know 7 2 1.00  
## # ... with 94 more rows

We can then compare each to the “consensus” topic for each book (the most common topic among its chapters) and see which were most often misidentified.

book\_topics <- chapter\_classifications %>%  
 count(title, topic) %>%  
 group\_by(title) %>%  
 top\_n(1, n) %>%  
 ungroup() %>%  
 transmute(consensus = title, topic)  
  
chapter\_classifications %>%  
 inner\_join(book\_topics, by = "topic") %>%  
 filter(title != consensus)

## # A tibble: 0 x 5  
## # ... with 5 variables: title <chr>, chapter <int>, topic <int>, gamma <dbl>,  
## # consensus <chr>

### By word assignments: augment

assignments <- augment(chapters\_lda, data = chapters\_dtm)  
assignments

## # A tibble: 69,284 x 4  
## document term count .topic  
## <chr> <chr> <dbl> <dbl>  
## 1 Musicians of To-Day\_1 music 519 1  
## 2 Birds Every Child Should Know\_9 music 2 2  
## 3 Houses and House-Life of the American Aborigines\_22 music 3 4  
## 4 Pride and Prejudice\_18 music 2 3  
## 5 Birds Every Child Should Know\_4 music 5 2  
## 6 Birds Every Child Should Know\_2 music 2 2  
## 7 Pride and Prejudice\_10 music 2 3  
## 8 Pride and Prejudice\_8 music 1 3  
## 9 Birds Every Child Should Know\_10 music 4 2  
## 10 Birds Every Child Should Know\_13 music 1 2  
## # ... with 69,274 more rows

assignments <- assignments %>%  
 separate(document, c("title", "chapter"), sep = "\_", convert = TRUE) %>%  
 inner\_join(book\_topics, by = c(".topic" = "topic"))  
  
assignments

## # A tibble: 69,284 x 6  
## title chapter term count .topic consensus   
## <chr> <int> <chr> <dbl> <dbl> <chr>   
## 1 Musicians of To-Day 1 music 519 1 Musicians of To-Day   
## 2 Birds Every Child Shoul~ 9 music 2 2 Birds Every Child Should~  
## 3 Houses and House-Life o~ 22 music 3 4 Houses and House-Life of~  
## 4 Pride and Prejudice 18 music 2 3 Pride and Prejudice   
## 5 Birds Every Child Shoul~ 4 music 5 2 Birds Every Child Should~  
## 6 Birds Every Child Shoul~ 2 music 2 2 Birds Every Child Should~  
## 7 Pride and Prejudice 10 music 2 3 Pride and Prejudice   
## 8 Pride and Prejudice 8 music 1 3 Pride and Prejudice   
## 9 Birds Every Child Shoul~ 10 music 4 2 Birds Every Child Should~  
## 10 Birds Every Child Shoul~ 13 music 1 2 Birds Every Child Should~  
## # ... with 69,274 more rows

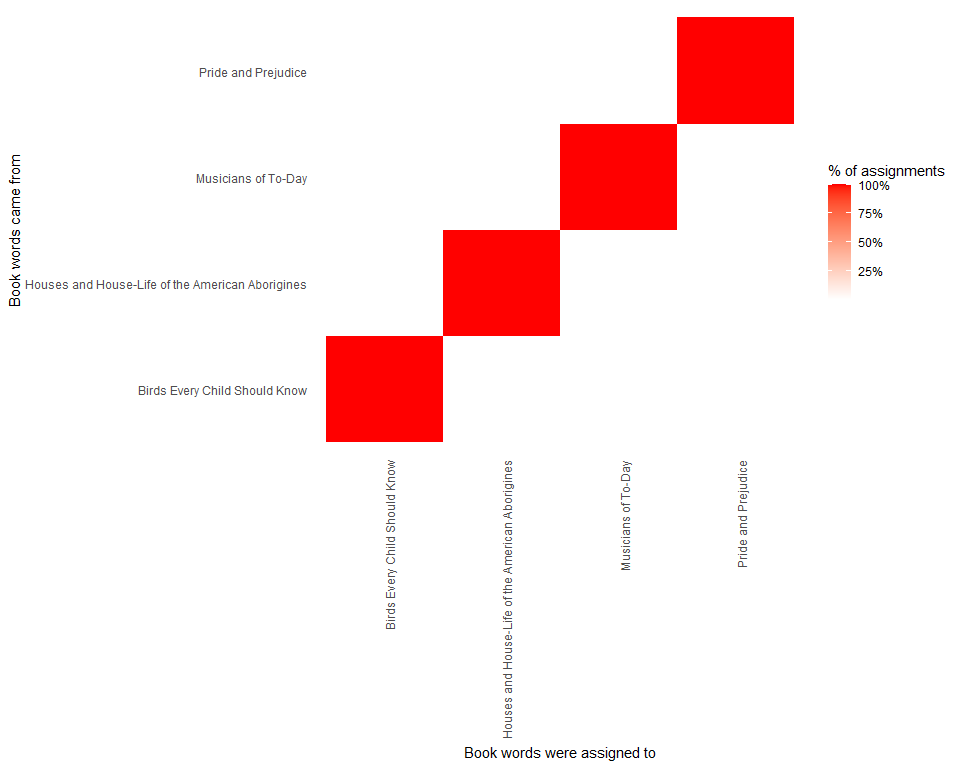
library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

assignments %>%  
 count(title, consensus, wt = count) %>%  
 group\_by(title) %>%  
 mutate(percent = n / sum(n)) %>%  
 ggplot(aes(consensus, title, fill = percent)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(high = "red", label = percent\_format()) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1),  
 panel.grid = element\_blank()) +  
 labs(x = "Book words were assigned to",  
 y = "Book words came from",  
 fill = "% of assignments")



Confusion matrix showing where LDA assigned the words from each book.

What were the most commonly mistaken words?

wrong\_words <- assignments %>%  
 filter(title != consensus)  
  
wrong\_words

## # A tibble: 29 x 6  
## title chapter term count .topic consensus   
## <chr> <int> <chr> <dbl> <dbl> <chr>   
## 1 Houses and House-Life of the A~ 19 paris 1 1 Musicians of ~  
## 2 Houses and House-Life of the A~ 19 performa~ 1 1 Musicians of ~  
## 3 Houses and House-Life of the A~ 17 performa~ 1 1 Musicians of ~  
## 4 Houses and House-Life of the A~ 17 scene 1 1 Musicians of ~  
## 5 Houses and House-Life of the A~ 17 february 1 1 Musicians of ~  
## 6 Houses and House-Life of the A~ 18 phrases 1 1 Musicians of ~  
## 7 Houses and House-Life of the A~ 19 september 1 1 Musicians of ~  
## 8 Houses and House-Life of the A~ 18 atmosphe~ 1 1 Musicians of ~  
## 9 Houses and House-Life of the A~ 17 admirable 1 1 Musicians of ~  
## 10 Houses and House-Life of the A~ 17 clearness 1 1 Musicians of ~  
## # ... with 19 more rows

wrong\_words %>%  
 count(title, consensus, term, wt = count) %>%  
 ungroup() %>%  
 arrange(desc(n))

## # A tibble: 28 x 4  
## title consensus term n  
## <chr> <chr> <chr> <dbl>  
## 1 Houses and House-Life of the American Abo~ Musicians of To-~ performan~ 2  
## 2 Houses and House-Life of the American Abo~ Musicians of To-~ 1848 1  
## 3 Houses and House-Life of the American Abo~ Musicians of To-~ admirable 1  
## 4 Houses and House-Life of the American Abo~ Musicians of To-~ atmosphere 1  
## 5 Houses and House-Life of the American Abo~ Musicians of To-~ born 1  
## 6 Houses and House-Life of the American Abo~ Musicians of To-~ clearness 1  
## 7 Houses and House-Life of the American Abo~ Musicians of To-~ confines 1  
## 8 Houses and House-Life of the American Abo~ Musicians of To-~ conscious 1  
## 9 Houses and House-Life of the American Abo~ Musicians of To-~ discovers 1  
## 10 Houses and House-Life of the American Abo~ Musicians of To-~ faint 1  
## # ... with 18 more rows