Topic_modeling

Dongkai Wu, Xu Luo, Xinpeng Hua, Yifeng Fan

2022-11-15

Import IMDB data

```
IMDB_Dataset <- read_csv("IMDB Dataset.csv")
```

```
## Rows: 50000 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): review, sentiment
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#Data Cleaning

```
#Delete the sentiment column.
imdb_1col <- IMDB_Dataset[,-(2)]

# Since the 50,000 dataset is too large to run, we decided to decrease the sample size and use only the
imdb_5000 <- imdb_1col[1:5000,]

#Give each review a number from 1 to 5000.
imdb_new <- imdb_5000 %>%
  mutate(review_num = row_number())

#Count the number of each words in each reviews.
imdb_finl <- imdb_new %>%
  unnest_tokens(word, review) %>%
   group_by(review_num) %>%
   count(word)

#Using the stop_words data to delete some common useless words.
word_counts <- imdb_finl %>%
  anti_join(stop_words) %>%
  count(review_num, word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
# Get rid of br and some high-frequency common words in each topics.
word_counts <- word_counts %>%
  filter(word != 'br' & word != 'film' & word != 'films')
fre_words <- c("1","2","3","4","5","6","7","8","9","10","film", "movie","character","time","watch",
               "story","people","plot","play","scene","director","actor","actress","make","life",
               "main","bad","worst","wrong","good","acting","series","cast","lot","real","top",
               "world","perform","feel","role","stuff","screen","tv","dvd","role","worth","take",
```

```
              "take","guy","excellent","fan","day","bit","script","set","hard","absolutely",
              "completely","true","john","job","minutes","audience","line","pretty","read","love")

word_counts <- word_counts[!grepl(paste(fre_words,collapse="|"),word_counts$`word`),]

#Transform the data set to a format that can be fitted in the LDA function.
imdb_dtm <- word_counts %>%
  cast_dtm(review_num, word, n)
```

```
#Create a 9-topic LDA model.
#We tried different numbers of topic from 5 to 15, and we find the top selected words seems most convin
imdb_lda <- LDA(imdb_dtm, k = 9, control = list(seed = 1234))
imdb_lda
```

```
## A LDA_VEM topic model with 9 topics.
```

```
#The beta shows the probability of that term being generated from that topic.
imdb_topics <- tidy(imdb_lda, matrix = "beta")
imdb_topics
```

```
## # A tibble: 356,454 x 3
##    topic term           beta
##    <int> <chr>         <dbl>
##  1     1 accustomed 5.93e- 5
##  2     2 accustomed 4.96e-15
##  3     3 accustomed 5.21e- 5
##  4     4 accustomed 1.05e-25
##  5     5 accustomed 1.44e- 4
##  6     6 accustomed 7.63e- 5
##  7     7 accustomed 3.16e- 5
##  8     8 accustomed 2.06e-24
##  9     9 accustomed 1.72e-23
## 10     1 agenda     3.66e- 5
## # ... with 356,444 more rows
## # i Use `print(n = ...)` to see more rows
```

```
#The same as the previous step, but arrange the data by topics. We picked the top 5 frequency words in
top_terms <- imdb_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

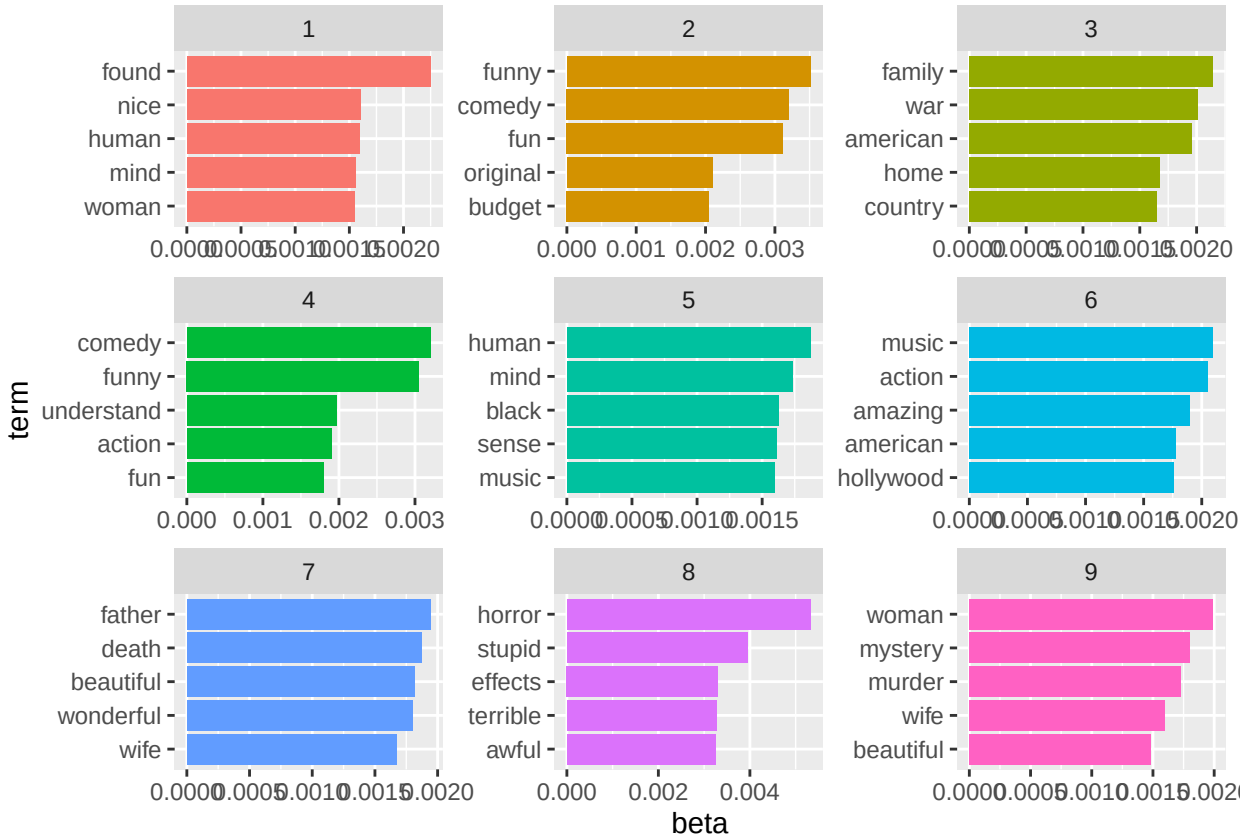
```
## # A tibble: 45 x 3
##    topic term        beta
##    <int> <chr>      <dbl>
##  1     1 found    0.00225
##  2     1 nice     0.00160
##  3     1 human    0.00159
##  4     1 mind     0.00155
##  5     1 woman    0.00155
##  6     2 funny    0.00352
##  7     2 comedy   0.00320
##  8     2 fun      0.00312
```

```
##  9     2 original 0.00210
## 10     2 budget   0.00205
## # ... with 35 more rows
## # i Use `print(n = ...)` to see more rows
```

#Data visualization

```
#Use slice_max() to find the 10 terms that are most common within each topic.
imdb_top_terms <- imdb_topics %>%
  group_by(topic) %>%
  slice_max(beta, n =5) %>%
  ungroup() %>%
  arrange(topic, -beta)

#Use ggplot to make a graph of words that extracted from each topic.
imdb_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```



#From the graph above, we can consider them as the rough features of each topic. It indicates that the most frequent words in each topic has a significance difference. However we are not yet able to define what each topic is about. For example, topic 2 and topic 4 both showed a high frequency of word comedy. And from some words like “found” and “budget” are hard to tell the movie genre.

```r
# have a look a some of the results (posterior distributions)
tmResult <- posterior(imdb_lda)
# format of the resulting object
theta <- tmResult$topics

beta <- tmResult$terms
```

```r
top3termsPerTopic <- terms(imdb_lda, 3)
topicNames <- apply(top3termsPerTopic, 2, paste, collapse=" ")
topicNames
```

```
##                   Topic 1                   Topic 2                   Topic 3
##        "found nice human"        "funny comedy fun"     "family war american"
##                   Topic 4                   Topic 5                   Topic 6
## "comedy funny understand"        "human mind black"    "music action amazing"
##                   Topic 7                   Topic 8                   Topic 9
##  "father death beautiful"   "horror stupid effects"    "woman mystery murder"
```

```r
#select top 3 words from each topic
```

```r
# re-rank top topic terms for topic names
topicNames <- apply(lda::top.topic.words(beta, 3, by.score = T), 2, paste, collapse = " ")

# What are the most probable topics in the entire collection?
topicProportions <- colSums(theta) / nDocs(imdb_dtm)
# mean probabilities over all paragraphs
names(topicProportions) <- topicNames
# assign the topic names we created before
sort(topicProportions, decreasing = TRUE)
```

```
##         horror waste gore           jokes fun funny      comedy funny episode
##                0.20535313                0.13153281                0.11955533
## amazing studio soundtrack detective mystery murders    billy father wonderful
##                0.09590189                0.09473411                0.09434081
##              alien sci fi interactions insane human           country war mel
##                0.09423333                0.08615559                0.07819300
```

```r
# show summed proportions in decreased order
```

```r
soP <- sort(topicProportions, decreasing = TRUE)
paste(round(soP, 5), ":", names(soP))
```

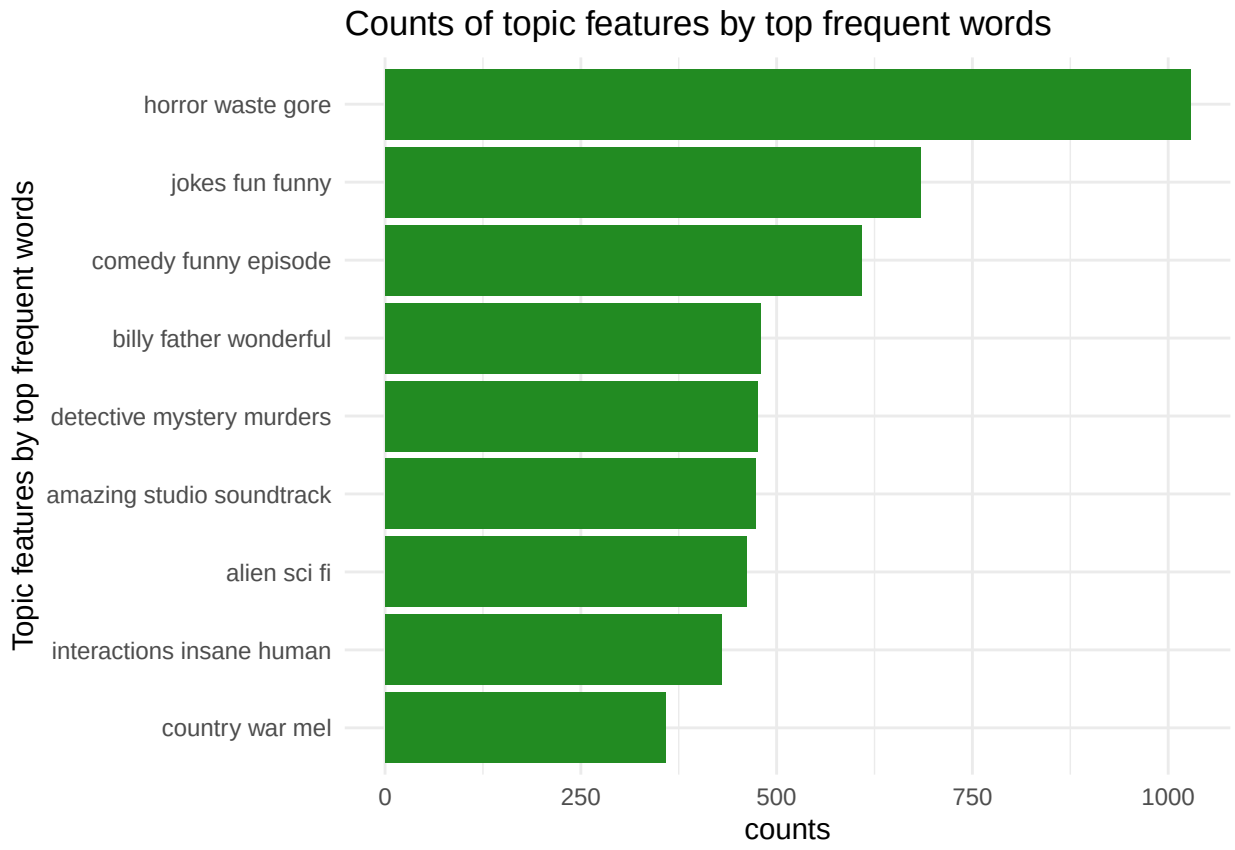
```
## [1] "0.20535 : horror waste gore"         "0.13153 : jokes fun funny"
## [3] "0.11956 : comedy funny episode"      "0.0959 : amazing studio soundtrack"
## [5] "0.09473 : detective mystery murders" "0.09434 : billy father wonderful"
## [7] "0.09423 : alien sci fi"              "0.08616 : interactions insane human"
## [9] "0.07819 : country war mel"
```

```r
#rank the proportions of the different topics
```

```r
countsOfPrimaryTopics <- rep(0, 9)
names(countsOfPrimaryTopics) <- topicNames
for (i in 1:nDocs(imdb_dtm)) {
  topicsPerDoc <- theta[i, ]
# select topic distribution for document i
# get first element position from ordered list
```

```
  primaryTopic <- order(topicsPerDoc, decreasing = TRUE)[1]
  countsOfPrimaryTopics[primaryTopic] <- countsOfPrimaryTopics[primaryTopic] + 1
}
```

```
counts <- data.frame(names(countsOfPrimaryTopics),countsOfPrimaryTopics)
counts <- counts %>% rename(class=names.countsOfPrimaryTopics.,
                            count=countsOfPrimaryTopics) %>%
  arrange(desc(count))
ggplot(counts) +
  aes(x = reorder(class,count), y = count) +
  geom_col(fill = "#228B22") +
  labs(x = "Topic features by top frequent words", y = "counts", title="Counts of topic features by top
  theme_minimal() +
  coord_flip()
```




```
#Creat a bar chart of the number of reviews of each topic which is explained by top 3 words
```

#The chart shows that horror themes were the most commented on some movies, followed by funny themes and popularity. Then there is the theme of education, the theme of detectives and robbers, the theme of science fiction and the theme of music. Least commented on the theme of war.Therefore, these data can be speculated that the feeling of horror and gore after watching the movie will make people more want to comment on the movie, followed by the joy and funny brought by the comedy.

```
# assign each document of topic with the top frequent words and assign film types to each review
imdb_genres <- imdb_5000
New_topicNames <- c('Sci-fi','Funny','War','Comedy','Documentary','musical','Family','Horror','Crime')
for (i in 1:nDocs(imdb_dtm)) {
```

```
  max_index <- which.max(theta[i, ])[[1]]
# select topic distribution for document i
# get first element position from ordered list

  imdb_genres$top_words[i] <- topicNames[max_index]
  imdb_genres$class[i] <- New_topicNames[max_index]
}
```

```
## Warning: Unknown or uninitialised column: `top_words`.
```

```
## Warning: Unknown or uninitialised column: `class`.
```

```
#"imdb_class" means that finally each comment was successfully classified into its own label category
```

```
write.csv(x = imdb_genres, file = "imdb_genres.csv")
```