Topic Model

Ziheng Li

2024-11-07

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────────── tidy
verse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────────
──── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
me errors
```

```
library(tm)
```

```
## 载入需要的程序包：NLP
##
## 载入程序包：'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(topicmodels)
library(ldatuning)
library(tidytext)
library(Rtsne)
library(ggplot2)
library(wordcloud)
```

```
## 载入需要的程序包：RColorBrewer
```

```
library(RColorBrewer)
```

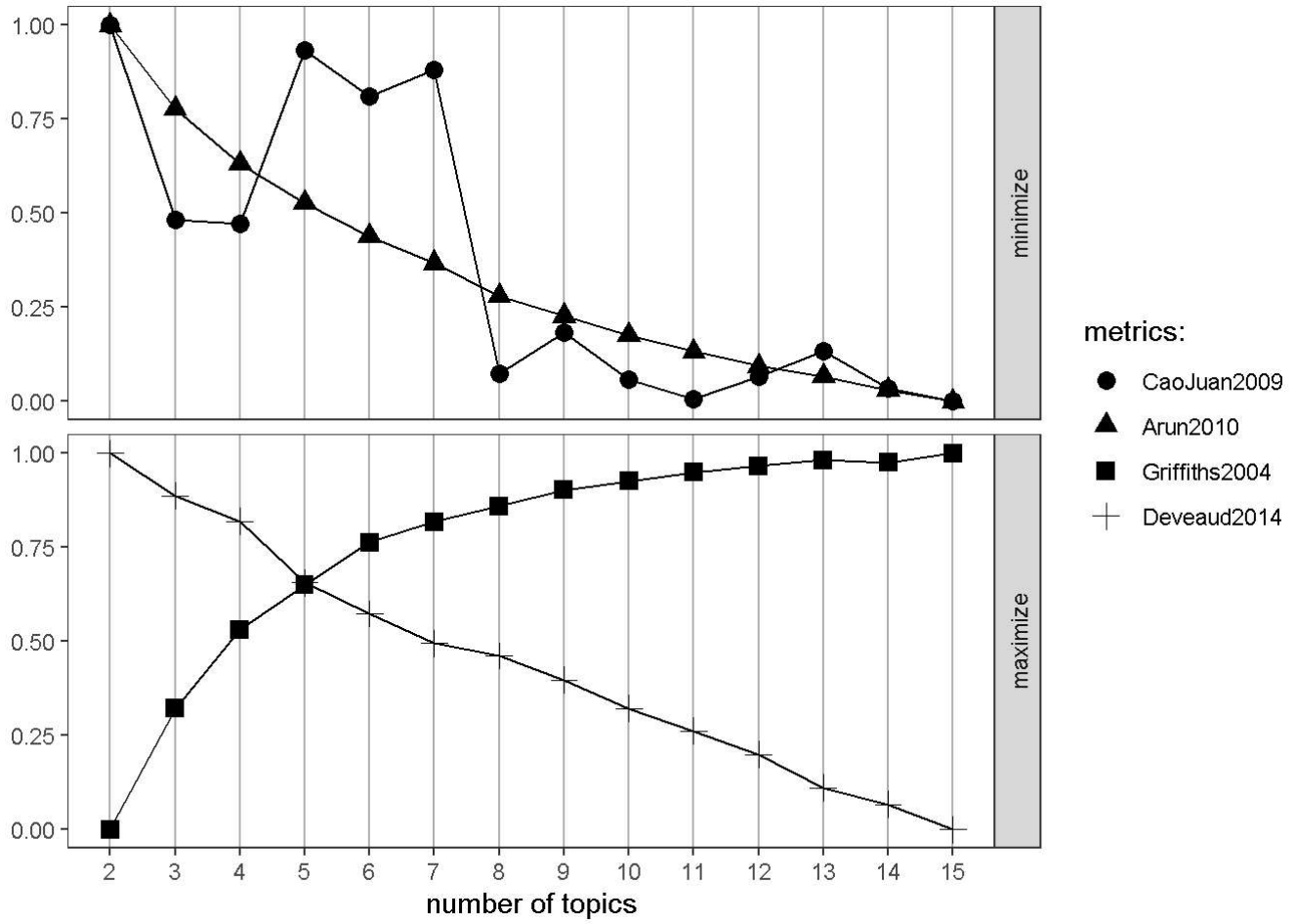
```
movie_data <- read_csv("movie_plots.csv", show_col_types = FALSE)

corpus <- VCorpus(VectorSource(movie_data$Plot))

corpus <- corpus %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removePunctuation) %>%
    tm_map(removeWords, stopwords("english")) %>%
    tm_map(stripWhitespace)

dtm <- DocumentTermMatrix(corpus)
```

```
result <- FindTopicsNumber(
    dtm,
    topics = seq(2, 15, by = 1),
    metrics = c("CaoJuan2009", "Arun2010", "Griffiths2004", "Deveaud2014"),
    method = "Gibbs",
    control = list(seed = 1234),
    mc.cores = 1L,
    verbose = TRUE
)
```

```
## fit models... done.
## calculate metrics:
##   CaoJuan2009... done.
##   Arun2010... done.
##   Griffiths2004... done.
##   Deveaud2014... done.
```

```
FindTopicsNumber_plot(result)
```

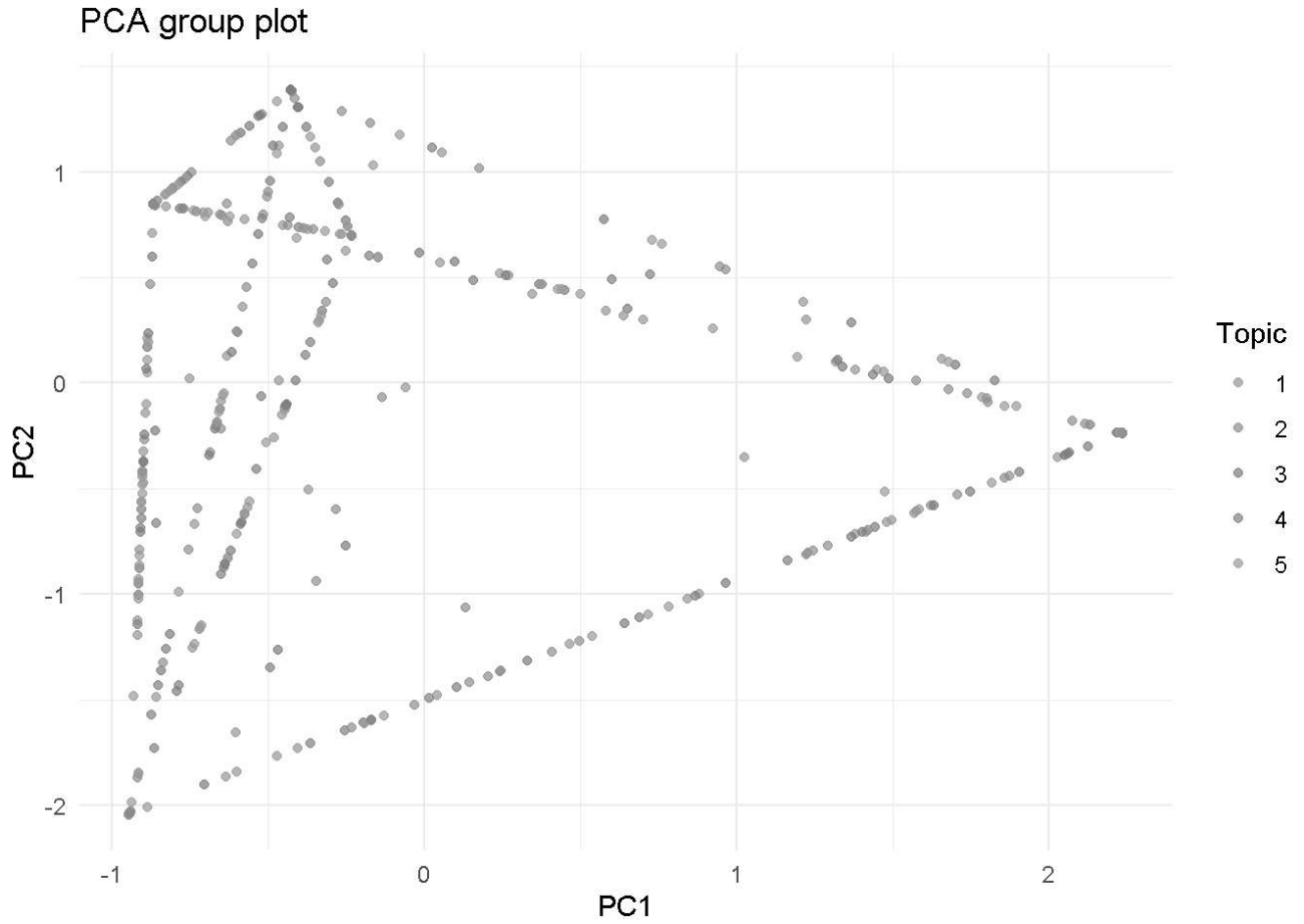
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the ldatuning package.
##   Please report the issue at <https://github.com/nikita-moor/ldatuning/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



#The scree plot helps identify the ideal number of topics k by highlighting where metric changes stabilize. For metrics to minimize (CaoJuan2009 and Arun2010), stability occurs around k=5 or k=6.indicating limited improvement beyond this point. For metrics to maximize (Griffiths2004 and Deveaud2014), Griffiths2004 stabilizes near k=5 and Deveaud2014 near k=6.

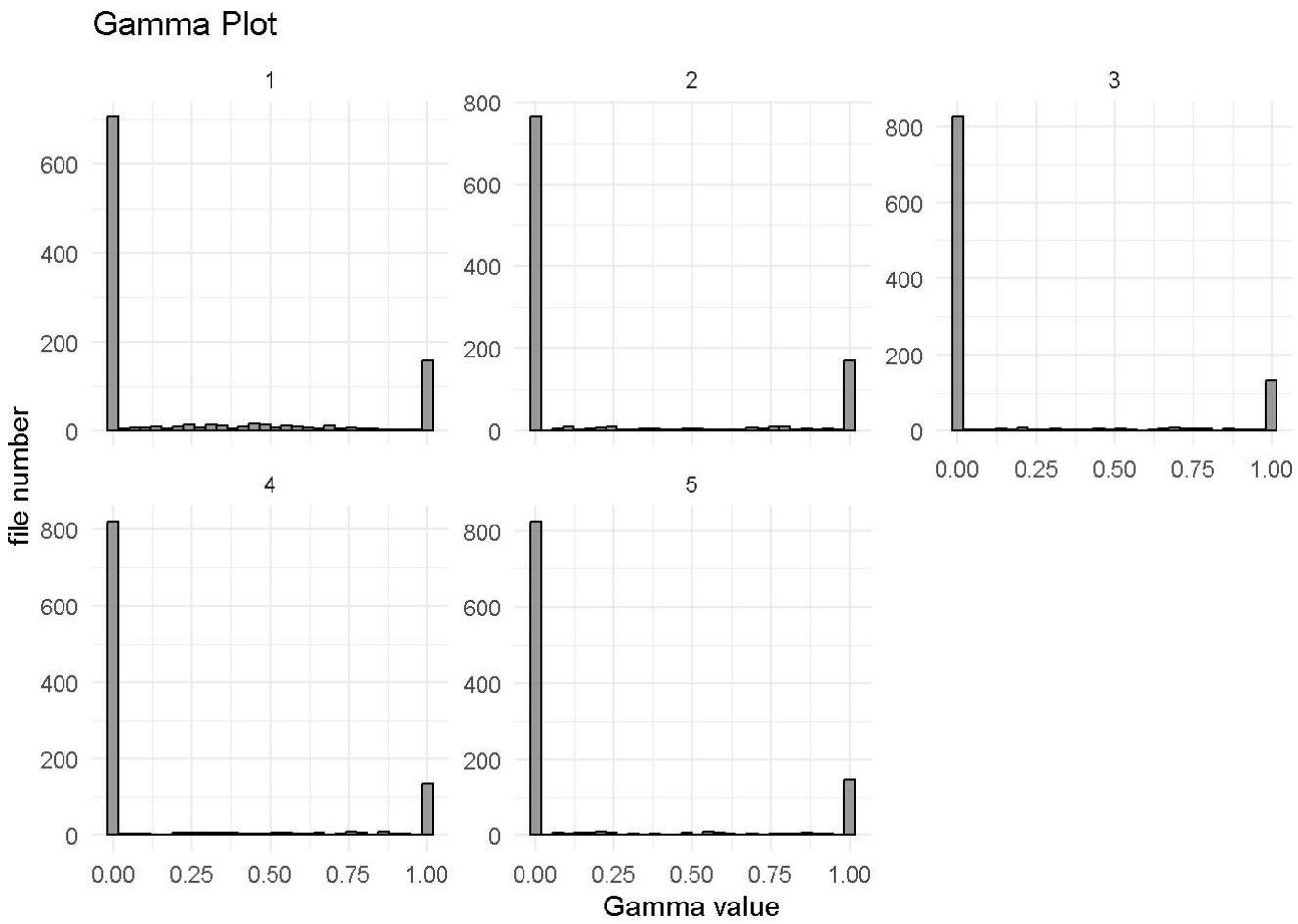
```
#set k to 6
k <- 5
lda_model <- LDA(dtm, k = k, control = list(seed = 1234))
```

```
gamma_matrix <- posterior(lda_model)$topics

pca_model <- prcomp(gamma_matrix, center = TRUE, scale. = TRUE)
pca_data <- as.data.frame(pca_model$x)

document_topics <- tidy(lda_model, matrix = "gamma")
doc_topic <- document_topics %>%
    group_by(document) %>%
    slice_max(gamma, n = 1) %>%
    ungroup()

pca_data$Topic <- factor(doc_topic$topic)

ggplot(pca_data, aes(x = PC1, y = PC2, color = Topic)) +
    geom_point(alpha = 0.7) +
    labs(title = "PCA group plot", x = "PC1", y = "PC2") +
    theme_minimal()
```



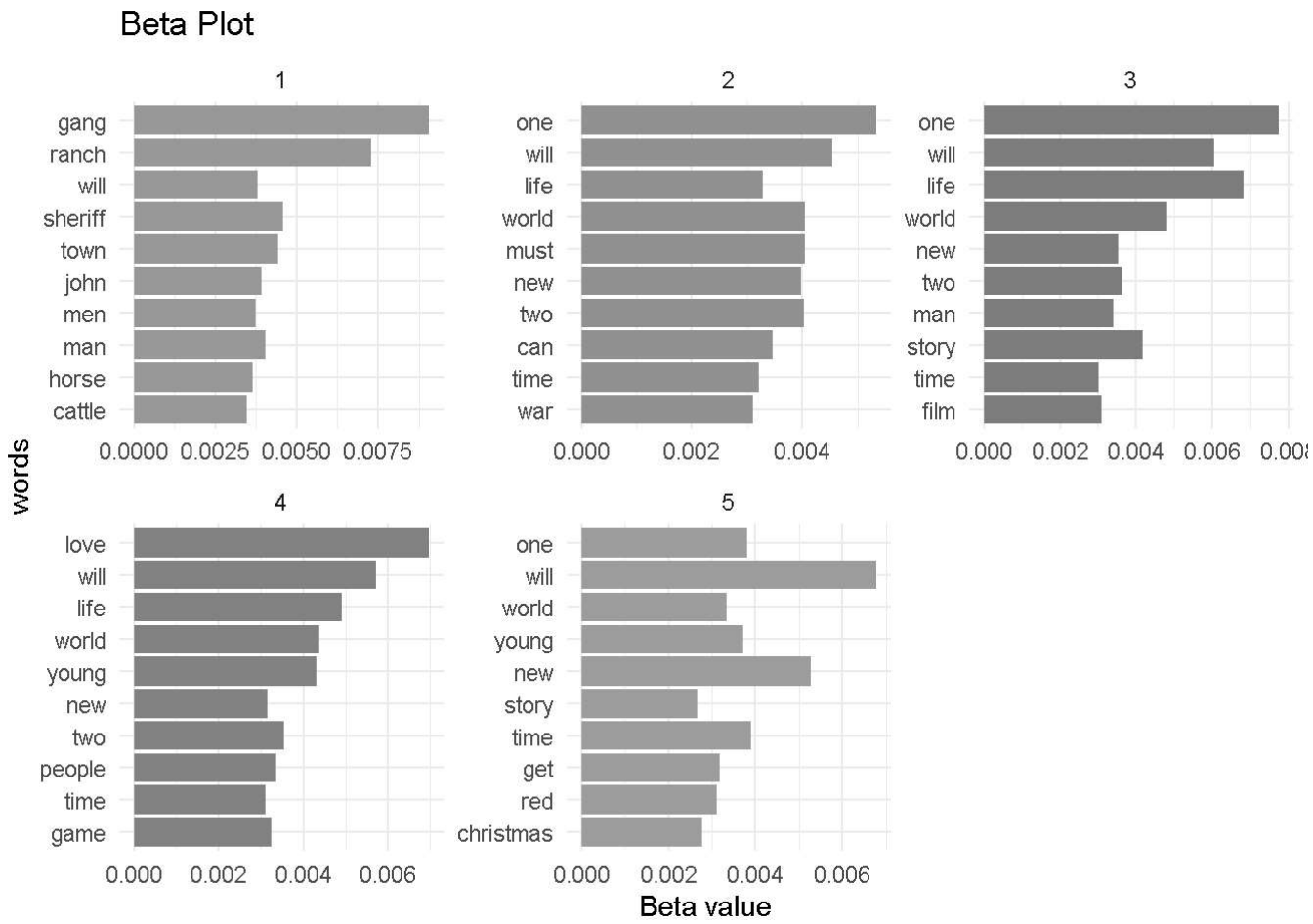
#This PCA clustering plot shows the distribution of the Gamma matrix of the topic model after downscaling it to a two-dimensional space. Each point represents a document and the colors indicate different topics (5 topics in total). Principal Component Distribution: The horizontal (PC1) and vertical (PC2) axes represent the first and second principal components, respectively, which capture the main variance of the data.Documents are distributed along the PC1 and PC2 axes, forming an extended triangular region that shows some structure.

```
document_topics <- tidy(lda_model, matrix = "gamma")

# Gamma plot
ggplot(document_topics, aes(x = gamma)) +
    geom_histogram(bins = 30, fill = "steelblue", color = "black", alpha = 0.7) +
    facet_wrap(~ topic, scales = "free_y") +
    labs(title = "Gamma Plot", x = "Gamma value", y = "file number") +
    theme_minimal()
```



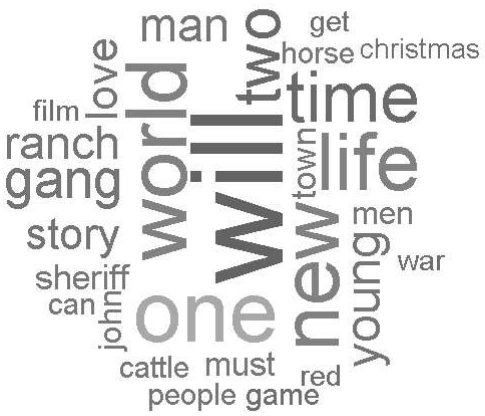
#Clear topic assignment: the distribution of Gamma values is centered on 0 and 1, indicating that the model is able to attribute most of the documents to a certain topic more clearly, and the documents have a high degree of attribution on the topic. Good differentiation between topics: there are almost no documents in the middle region, which indicates that there is less confusion in the attribution of documents to topics, which may be an indication of the high differentiation of the model. Direction for improvement: If you want to explore more detailed topics, you may be able to increase the number of topics and observe whether more documents with intermediate Gamma values appear, so that more refined topic content can be captured

```
topic_terms <- tidy(lda_model, matrix = "beta")

top_terms <- topic_terms %>%
    group_by(topic) %>%
    slice_max(beta, n = 10) %>%
    ungroup() %>%
    arrange(topic, -beta)

ggplot(top_terms, aes(x = reorder(term, beta), y = beta, fill = factor(topic))) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    facet_wrap(~ topic, scales = "free") +
    coord_flip() +
    labs(title = "Beta Plot", x = "words", y = "Beta value") +
    theme_minimal()
```



#Representative words for each theme: As we can see from the high-frequency words, there are some representative words for each theme, which help us to recognize the general content of each theme. Similarity between themes: Some words such as “life”, “world”, “time”, “will”, etc. appear in more than one theme, indicating that these themes may have similarities. ”, etc. appear in more than one theme, indicating that there may be some overlap between these themes. These words are more generic and therefore may appear in more than one theme. Improvement direction: If you want to get a clearer distinction of themes, you can try to increase the number of themes to capture more detailed theme features; or remove some common words in the text pre-processing to enhance the distinction of themes.

```
palette <- brewer.pal(8, "Dark2")

all_terms <- top_terms %>%
    group_by(term) %>%
    summarize(total_beta = sum(beta)) %>%
    arrange(desc(total_beta))
wordcloud(words = all_terms$term,
          freq = all_terms$total_beta,
          min.freq = 0.001,
          colors = palette,
          random.order = FALSE,
          rot.per = 0.35,
          scale = c(4, 0.5),
          main = "wordcloud")
```


man
two
get
horse
christmas
love
world
time
film
ranch
will
life
gang
town
men
story
new
young
war
sheriff
can
john
one
cattle
must
red
people
game