TopicModeling

Yiming Chen

Packages

```r
library(tidyverse)
```

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

```r
library(tm)
```

```
Loading required package: NLP

Attaching package: 'NLP'

The following object is masked from 'package:ggplot2':

    annotate
```

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

```
Loading required package: RColorBrewer
```

```r
library(RColorBrewer)
```

Data Cleaning

```r
movie_data = read_csv("movie_plots.csv")
view(movie_data)
```

```r
plots_by_word <- movie_data %>% unnest_tokens(word, Plot)
plot_word_counts <- plots_by_word %>%
  anti_join(stop_words,by = join_by(word)) %>%
  count("Movie Name", word, sort = TRUE)
```

```r
corpus <- VCorpus(VectorSource(movie_data$Plot))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)
dtm <- DocumentTermMatrix(corpus)
```

1. Tokenization: split the "Plot" column into individual words.
2. Removing Stop Words: remove common stop words (like "the," "and," "is") which do not contribute to meaningful topics.
3. Counting Words: count the occurrences of each word, sorted by frequency.

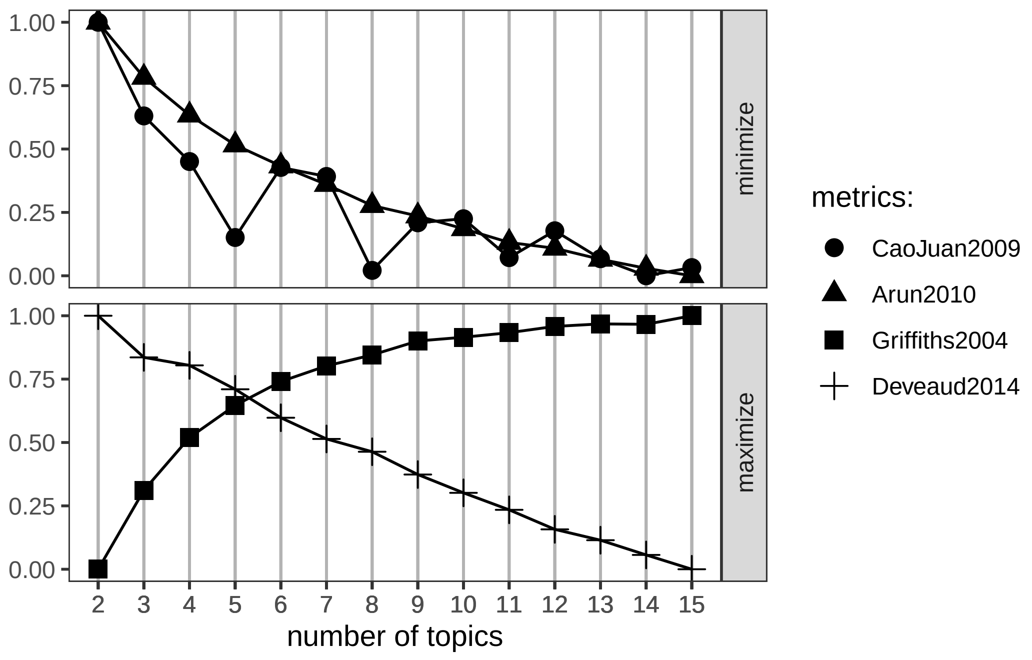
Visualizations

```r
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.
```

```r
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>.
```

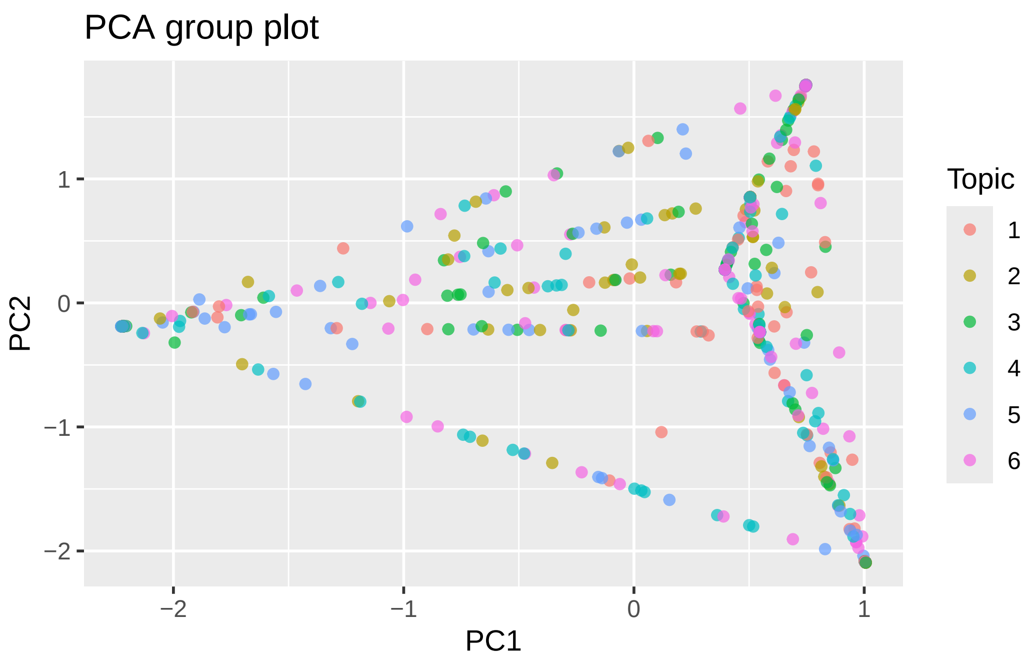


For each metric, identify the point at which the metric either stabilizes or reaches an optimal value. The intersection of these indicators provides a recommendation for the best number of topics. Around k=5 to k=8, most metrics start to stabilize or reach optimal values, we could choose one of these as the ideal number of topics. Let's try k=6.

```
#set k to 6
k <- 6
lda_model <- LDA(dtm, k = k, control = list(seed = 1234))
terms(lda_model, 10)
```

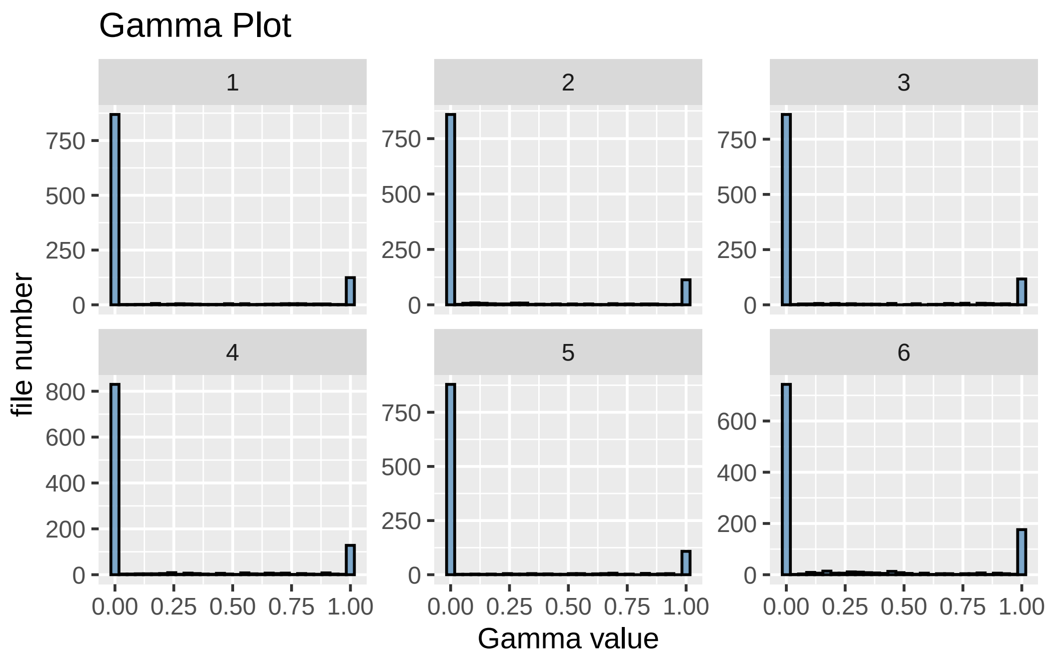
```
      Topic 1 Topic 2  Topic 3 Topic 4 Topic 5 Topic 6
 [1,] "man"   "new"    "one"   "will"  "town"  "will"
 [2,] "gang"  "one"    "young" "gang"  "film"  "world"
 [3,] "ranch" "will"   "town"  "bill"  "story" "one"
 [4,] "war"   "story"  "world" "ranch" "world" "life"
 [5,] "one"   "can"    "gold"  "get"   "life"  "new"
 [6,] "two"   "life"   "find"  "jim"   "two"   "man"
 [7,] "back"  "two"    "get"   "tom"   "new"   "war"
 [8,] "will"  "action" "will"  "life"  "one"   "time"
 [9,] "town"  "game"   "two"   "love"  "steve" "earth"
[10,] "new"   "world"  "can"   "young" "time"  "must"
```

```
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")
```



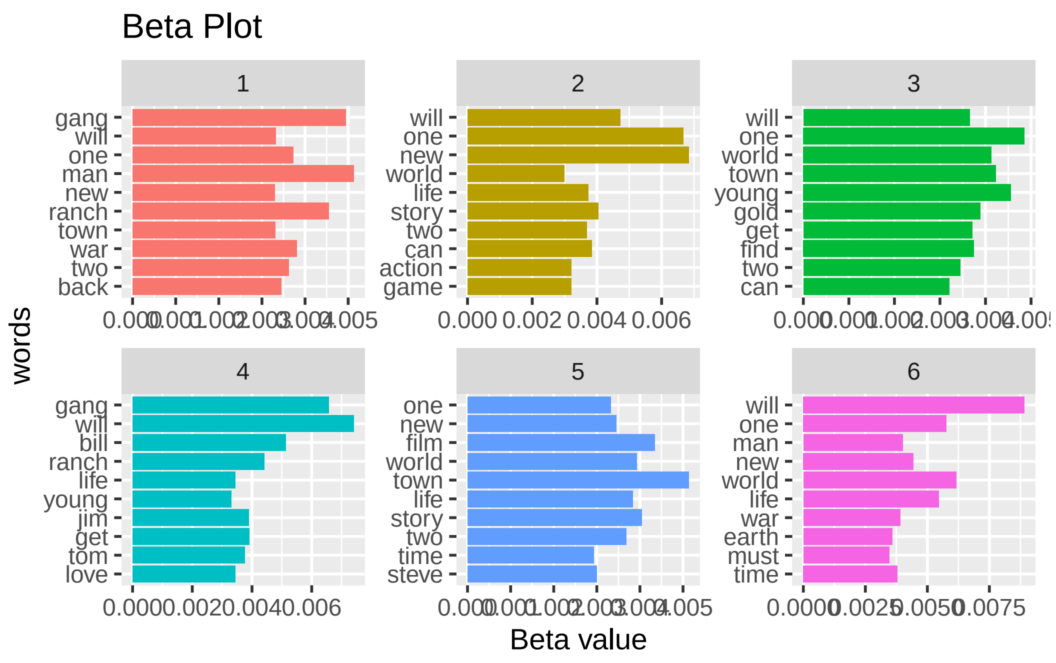
The code fits an LDA model with k=6 topics on a document-term matrix derived from movie plots, extracts the document-topic probabilities (gamma matrix), and uses PCA to reduce the dimensionality for visualization. Each document is assigned to its dominant topic based on the highest probability, and a scatter plot is generated to display the documents in a 2D space with colors representing different topics. The resulting PCA plot reveals how documents cluster by topic, illustrating the distinctiveness and relationships between topics by showing how closely documents with similar themes group together.

```r
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")
```



The Gamma Plot illustrates the distribution of topic probabilities (gamma values) for each document across the six topics in the LDA model. Each subplot represents a specific topic and displays the frequency of gamma values for that topic. The plots reveal that most documents have gamma values close to 0 for most topics, while a smaller number of documents exhibit gamma values close to 1 for a single topic. This indicates a clear pattern of topic dominance, where the model assigns documents strongly to one primary topic rather than spreading their probabilities across multiple topics. Such a distribution aligns with the expectations of a well-performing topic model, effectively distinguishing documents into distinct topics.

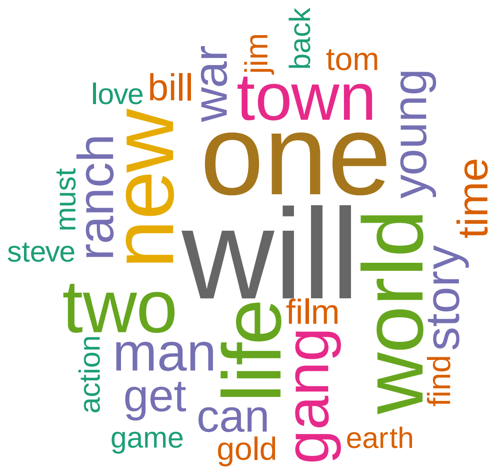
```r
topic_terms <- tidy(lda_model, matrix = "beta")

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")
```



The Beta Plot showcases the top 10 terms for each of the six topics in the LDA model, with each term's importance quantified by its beta value. Beta values indicate the probability of a word being associated with a particular topic, where higher beta values correspond to words that are more strongly representative of the topic. Each subplot represents one topic and displays the most significant words in descending order of their beta values. This visualization provides a clear understanding of the defining terms for each topic, making it easier to interpret and label topics based on the prominent words associated with them. It offers a comprehensive view of the thematic structure within the LDA model.

```r
library(RColorBrewer)
```

```r
all_terms <- top_terms %>%
    group_by(term) %>%
    summarize(total_beta = sum(beta)) %>%
    arrange(desc(total_beta))

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



The provided code generates a word cloud to visualize the most significant terms across all topics in an LDA model. Using the top_terms dataset, terms are aggregated by summing their beta values across topics, reflecting their overall importance. Larger beta values indicate a stronger association with the topics, and the terms are sorted in descending order of importance. The wordcloud function uses these beta values to size the words proportionally, with visually distinct colors assigned from the "Dark2" palette. The resulting word cloud highlights key terms such as *"will"*, *"one"*, *"new"*, and *"world"*, which dominate across topics. This visualization effectively conveys the most influential words in the corpus, providing an intuitive summary of the thematic structure.