Topic Modeling

Yibing Wang

2024-11-08

```
library(topicmodels)
library(tm)
```

```
## Loading required package: NLP
```

```
library(ldatuning)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
movie_data <- read.csv("~/Downloads/xid-113733278_1")

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)

#Document-Term Matrix
dtm <- DocumentTermMatrix(corpus)
```

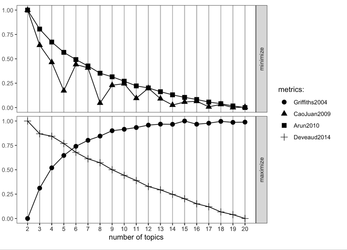
```
#class(dtm)
```

```
# optimal number of topics
result <- FindTopicsNumber(
  dtm,
  topics = seq(2, 20, by = 1),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 1234),
  mc.cores = 11L,
  verbose = TRUE
)
```

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

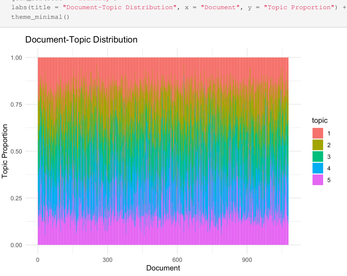



```
#Based on these metrics, k = 5 is likely to provide a good balance of coherent and distinct topics.it can capture the main themes in the movie plots dataset
```

```
#k=5 as an example
k <- 5
lda_model <- LDA(dtm, k = k, method = "Gibbs", control = list(seed = 1234))
terms(lda_model, 10)
```

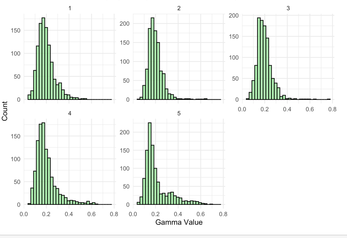
```
##       Topic 1     Topic 2 Topic 3   Topic 4   Topic 5
##  [1,] "war"     "will"  "life"    "film"    "gang"
##  [2,] "world"   "one"   "young"   "story"   "town"
##  [3,] "city"    "find"  "love"    "world"   "ranch"
##  [4,] "save"    "man"   "two"     "team"    "men"
##  [5,] "battle"  "new"   "girl"    "new"     "father"
##  [6,] "must"    "time"  "family"  "game"    "john"
##  [7,] "army"    "back"  "finds"   "series"  "bill"
##  [8,] "power"   "get"   "death"   "history" "killed"
##  [9,] "earth"   "now"   "years"   "set"     "money"
## [10,] "mysterious" "way"   "becomes" "great"   "gets"
```

```
#topic distribution per document
doc_topics <- posterior(lda_model)$topics

# df
doc_topics_df <- as.data.frame(doc_topics)
doc_topics_df$doc_id <- 1:nrow(doc_topics_df)
doc_topics_long <- reshape2::melt(doc_topics_df, id.vars = "doc_id", variable.name = "topic")

ggplot(doc_topics_long, aes(x = doc_id, y = value, fill = topic)) +
  geom_bar(stat = "identity") +
  labs(title = "Document-Topic Distribution", x = "Document", y = "Topic Proportion") +
  theme_minimal()
```




```
#Since each color represents a different topic, it suggests that each document contains a balanced mix of themes. #A lower k could produce more general topics, while a higher k might have more specific themes. The plot, also su ggests that k = 5 produces a even mix of topics across all documents
```

```
gamma <- posterior(lda_model)$topics
doc_topics_df <- as.data.frame(doc_topics)
doc_topics_df$document <- 1:nrow(doc_topics_df)
doc_topics_long <- melt(doc_topics_df, id.vars = "document", variable.name = "topic", value.name = "gamma")

ggplot(doc_topics_long, aes(x = gamma)) +
  geom_histogram(bins = 30, fill = "lightgreen", color = "black", alpha = 0.7) +
  facet_wrap(~ topic, scales = "free_y") +
  labs(title = "Gamma Distribution (Topic Proportions in Documents)", x = "Gamma Value", y = "Count") +
  theme_minimal()
```




```
beta <- posterior(lda_model)$terms

# df
beta_df <- as.data.frame(beta)
beta_df$topic <- factor(1:nrow(beta_df))
beta_long <- melt(beta_df, id.vars = "topic", variable.name = "term", value.name = "beta")

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

ggplot(top_terms, aes(x = reorder(term, beta), y = beta, fill = topic)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  labs(title = "Beta Plot (Top Terms in Topics)", x = "Term", y = "Beta Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```