|  |
| --- |
| title: “Topic modeling project1” |
| author: “Haochen Pan Xiaoyanbin Cai Shengbo Wang” |
| date: “2022-11-14” |
| output: |
| word\_document: default |
| html\_document: default |

#### This is the r markdown file with codes and graphs, with no explanation. The pdf file we submitted is the file with explain and code.

library(topicmodels)

library(tidytext)  
library(ggplot2)

library(dplyr)

##

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

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

library(tidyr)  
library(jsonlite)

library("tm")

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

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ stringr 1.4.1  
## ✔ readr 2.1.3 ✔ forcats 0.5.2  
## ✔ purrr 0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ NLP::annotate() masks ggplot2::annotate()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ purrr::flatten() masks jsonlite::flatten()  
## ✖ dplyr::lag() masks stats::lag()

library(corrplot)

## corrplot 0.92 loaded

library(PerformanceAnalytics)

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
##   
##   
## The following objects are masked from 'package:dplyr':  
##   
## first, last  
##   
##   
##   
## The following object is masked from 'package:graphics':  
##   
## legend

library(stringi)

library(readr)  
IMDB\_Dataset <- read\_csv("D:/MSSP/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.

IMDB\_df <- tibble(IMDB\_Dataset)  
glimpse(IMDB\_df)

## Rows: 50,000  
## Columns: 2  
## $ review <chr> "One of the other reviewers has mentioned that after watchin…  
## $ sentiment <chr> "positive", "positive", "positive", "negative", "positive", …

head(IMDB\_df)

## # A tibble: 6 × 2  
## review senti…¹  
## <chr> <chr>   
## 1 "One of the other reviewers has mentioned that after watching just 1 … positi…  
## 2 "A wonderful little production. <br /><br />The filming technique is … positi…  
## 3 "I thought this was a wonderful way to spend time on a too hot summer… positi…  
## 4 "Basically there's a family where a little boy (Jake) thinks there's … negati…  
## 5 "Petter Mattei's \"Love in the Time of Money\" is a visually stunning… positi…  
## 6 "Probably my all-time favorite movie, a story of selflessness, sacrif… positi…  
## # … with abbreviated variable name ¹​sentiment

IMDB\_df %>%   
 mutate(review\_number = row\_number()) ->IMDB\_df   
IMDB\_df <- IMDB\_df %>% select(-sentiment)

library(tm)  
myCorpus <- Corpus(VectorSource(IMDB\_df$review))  
data\_clean <- tm\_map(myCorpus, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(myCorpus, removeWords, stopwords("english")):  
## transformation drops documents

data\_clean <- tm\_map(data\_clean, removeNumbers)

## Warning in tm\_map.SimpleCorpus(data\_clean, removeNumbers): transformation drops  
## documents

data\_clean <- tm\_map(data\_clean, removePunctuation)

## Warning in tm\_map.SimpleCorpus(data\_clean, removePunctuation): transformation  
## drops documents

data\_clean <- tm\_map(data\_clean, removeWords, c("this","the","one","can","also","but","br","moive","film","time"))

## Warning in tm\_map.SimpleCorpus(data\_clean, removeWords, c("this", "the", :  
## transformation drops documents

tdm <- TermDocumentMatrix(data\_clean)  
inspect(tdm)

## <<TermDocumentMatrix (terms: 162475, documents: 50000)>>  
## Non-/sparse entries: 4924101/8118825899  
## Sparsity : 100%  
## Maximal term length: 72  
## Weighting : term frequency (tf)  
## Sample :  
## Docs  
## Terms 12648 3025 31241 31437 31482 3655 40522 42947 43822 5709  
## even 1 4 4 5 2 3 2 1 6 2  
## good 0 2 0 7 1 2 2 4 1 2  
## just 1 5 1 4 2 2 6 4 1 3  
## like 5 4 7 15 3 4 3 7 2 8  
## movie 12 2 1 0 0 14 22 2 4 0  
## really 0 3 2 6 0 1 2 2 0 1  
## see 1 3 3 0 1 1 7 6 1 2  
## story 6 0 1 5 0 3 2 3 2 0  
## the 22 9 9 3 31 18 15 18 21 7  
## this 2 1 1 1 0 2 9 2 3 2

library(stm)

## stm v1.3.6 successfully loaded. See ?stm for help.   
## Papers, resources, and other materials at structuraltopicmodel.com

tdm\_dfm <- tidy(tdm) %>%   
 cast\_dfm(document = document, term = term, value = count)  
  
tdm\_lda <- stm(tdm\_dfm,   
 K = 10,   
 verbose = FALSE,   
 init.type = "LDA")  
summary(tdm\_lda)

## A topic model with 10 topics, 50000 documents and a 162475 word dictionary.

## Topic 1 Top Words:  
## Highest Prob: love, life, the, she, family, story, young   
## FREX: scarlett, streep, meryl, austen, natalie, rochester, eyre   
## Lift: abernethie, abuzz, achilless, acquart, adele, adriennes, ager   
## Score: mother, she, husband, love, father, woman, family   
## Topic 2 Top Words:  
## Highest Prob: movie, just, like, see, really, movies, this   
## FREX: please, movie, rented, maybe, sucked, want, moviebr   
## Lift: beethovan, cognaci, deke, exitingfunnydramatic, haarman, jabez, ¨director¨spent   
## Score: movie, bad, worst, stupid, movies, watch, waste   
## Topic 3 Top Words:  
## Highest Prob: the, story, great, well, good, characters, many   
## FREX: anime, animation, vhs, enjoyed, dvd, series, amazing   
## Lift: £amazing, «battlestar, «bazar», «blakes», «blindpassasjer», «farscape», «lexx»   
## Score: book, great, series, original, dvd, animation, story   
## Topic 4 Top Words:  
## Highest Prob: the, role, john, cast, performance, best, good   
## FREX: batman, stewart, holmes, columbo, reynolds, caine, flynn   
## Lift: “the, ackroyds, adolphe, aiello, ardolino, atkinson, barkley   
## Score: batman, stewart, supporting, robert, matthau, comedy, columbo   
## Topic 5 Top Words:  
## Highest Prob: show, the, funny, like, kids, comedy, school   
## FREX: football, carrey, barney, simpsons, marty, wwe, snl   
## Lift: aardman, aardmans, aardvark, abirrrd, activision, addictivebr, adv   
## Score: show, kids, funny, jokes, season, comedy, episode   
## Topic 6 Top Words:  
## Highest Prob: the, man, get, scene, gets, back, killed   
## FREX: truck, police, cops, chan, sniper, bullet, gun   
## Lift: abducts, acromegaly, adorf, adp, aguilar, alabamas, alejandra   
## Score: killer, police, car, kill, killed, cop, guy   
## Topic 7 Top Words:  
## Highest Prob: the, war, world, american, people, life, this   
## FREX: documentary, historical, propaganda, hitler, countries, soviet, germans   
## Lift: communist,     film,   with,  a,  as,  astounding,  journey   
## Score: war, documentary, political, historical, che, soldiers, soviet   
## Topic 8 Top Words:  
## Highest Prob: the, song, music, musical, songs, star, space   
## FREX: ship, trek, tarzan, concert, sinatra, musicals, laurel   
## Lift: aamir, abishag, abrahambr, adama, adelaides, admiralty, admirersbr   
## Score: songs, tarzan, musical, ship, planet, dance, song   
## Topic 9 Top Words:  
## Highest Prob: the, bad, horror, acting, plot, even, effects   
## FREX: zombies, freddy, scarecrow, abysmal, boll, troma, uwe   
## Lift: ¿actingthor, ¿special, £m, «average», «bluff», «hot», «i   
## Score: horror, bad, gore, zombie, budget, worst, effects   
## Topic 10 Top Words:  
## Highest Prob: the, characters, character, films, seems, even, much   
## FREX: bourne, bettie, pornography, lynch, polanski, verhoeven, pornographic   
## Lift: ‘collaborations’, ‘depraved’, ‘enjoyable’, ‘humanization’, ‘showdown’, “big, “extended   
## Score: sex, sexual, characters, violence, bettie, films, disturbing

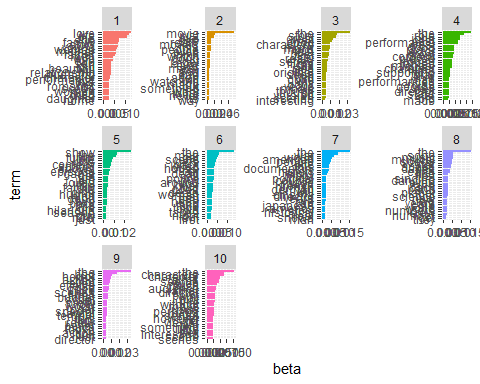
tidy(tdm\_lda)

## # A tibble: 1,624,750 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 "\b\b\b\ba" 3.11e- 79  
## 2 2 "\b\b\b\ba" 4.26e-153  
## 3 3 "\b\b\b\ba" 1.09e-197  
## 4 4 "\b\b\b\ba" 1.14e-204  
## 5 5 "\b\b\b\ba" 2.23e- 6  
## 6 6 "\b\b\b\ba" 1.41e- 57  
## 7 7 "\b\b\b\ba" 1.10e-169  
## 8 8 "\b\b\b\ba" 4.60e-216  
## 9 9 "\b\b\b\ba" 5.35e-238  
## 10 10 "\b\b\b\ba" 8.95e- 85  
## # … with 1,624,740 more rows

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

## # A tibble: 1,624,750 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 "\b\b\b\ba" 3.11e- 79  
## 2 2 "\b\b\b\ba" 4.26e-153  
## 3 3 "\b\b\b\ba" 1.09e-197  
## 4 4 "\b\b\b\ba" 1.14e-204  
## 5 5 "\b\b\b\ba" 2.23e- 6  
## 6 6 "\b\b\b\ba" 1.41e- 57  
## 7 7 "\b\b\b\ba" 1.10e-169  
## 8 8 "\b\b\b\ba" 4.60e-216  
## 9 9 "\b\b\b\ba" 5.35e-238  
## 10 10 "\b\b\b\ba" 8.95e- 85  
## # … with 1,624,740 more rows

library(ggplot2)  
library(dplyr)  
  
tdm\_top\_terms <- tdm\_topics %>%  
 group\_by(topic) %>%  
 slice\_max(beta, n = 30) %>%   
 ungroup() %>%  
 arrange(topic, -beta)  
  
tdm\_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()



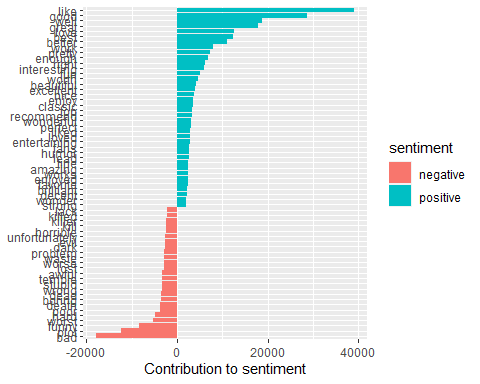
library(dplyr)  
library(tidytext)  
  
ap\_td <- tidy(tdm)  
ap\_td

## # A tibble: 4,924,101 × 3  
## term document count  
## <chr> <chr> <dbl>  
## 1 accustomed 1 1  
## 2 agenda 1 1  
## 3 agreements 1 1  
## 4 appeal 1 1  
## 5 around 1 1  
## 6 audiences 1 1  
## 7 away 1 1  
## 8 awaybr 1 1  
## 9 become 1 1  
## 10 bitches 1 1  
## # … with 4,924,091 more rows

ap\_sentiments <- ap\_td %>%  
 inner\_join(get\_sentiments("bing"), by = c(term = "word"))  
  
ap\_sentiments

## # A tibble: 813,339 × 4  
## term document count sentiment  
## <chr> <chr> <dbl> <chr>   
## 1 appeal 1 1 positive   
## 2 brutality 1 1 negative   
## 3 charm 1 1 positive   
## 4 classic 1 1 positive   
## 5 comfortable 1 1 positive   
## 6 crooked 1 1 negative   
## 7 darker 1 1 negative   
## 8 death 1 1 negative   
## 9 faint 1 1 negative   
## 10 injustice 1 1 negative   
## # … with 813,329 more rows

library(ggplot2)  
  
ap\_sentiments %>%  
 count(sentiment, term, wt = count) %>%  
 ungroup() %>%  
 filter(n >= 2000) %>%  
 mutate(n = ifelse(sentiment == "negative", -n, n)) %>%  
 mutate(term = reorder(term, n)) %>%  
 ggplot(aes(n, term, fill = sentiment)) +  
 geom\_col() +  
 labs(x = "Contribution to sentiment", y = NULL)



library(ggplot2)  
library(igraph)

##

## The following objects are masked from 'package:purrr':  
##   
## compose, simplify

## The following object is masked from 'package:tibble':  
##   
## as\_data\_frame

## The following object is masked from 'package:tidyr':  
##   
## crossing

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

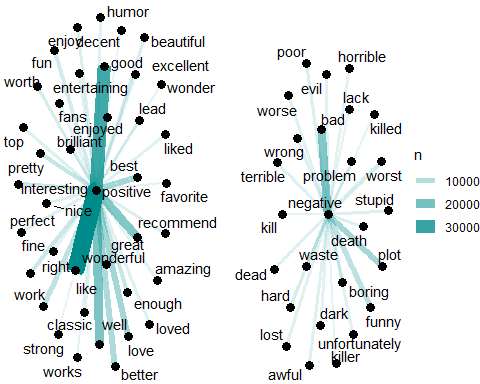
## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

library(ggraph)

set.seed(1234)  
ap\_sentiments %>%  
 count(sentiment, term, wt = count) %>%  
 ungroup() %>%  
 filter(n >= 2000) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n, edge\_width = n), edge\_colour = "cyan4") +  
 geom\_node\_point(size = 3) +  
 geom\_node\_text(aes(label = name), repel = TRUE,   
 point.padding = unit(0.7, "lines")) +  
 theme\_void()

## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` in the `default\_aes` field and elsewhere instead.



library(tidyr)  
beta\_wide <- tdm\_topics %>%  
 mutate(topic = paste0("topic", topic)) %>%  
 pivot\_wider(names\_from = topic, values\_from = beta) %>%   
 filter(topic1 > .001 | topic2 > .001) %>%  
 mutate(log\_ratio = log2(topic2 / topic1))  
beta\_wide

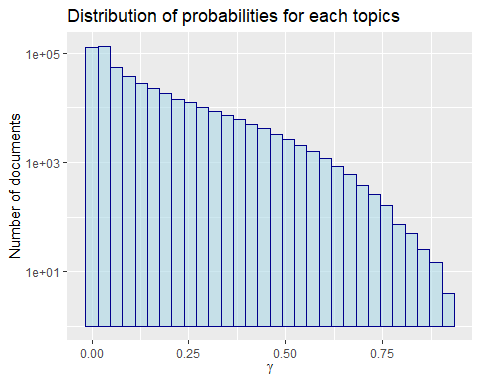
## # A tibble: 298 × 12  
## term topic1 topic2 topic3 topic4 topic5 topic6 topic7 topic8  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 absolut… 1.50e-4 1.62e- 3 7.73e-4 1.95e-4 1.16e-11 4.80e-19 1.58e- 8 1.24e-10  
## 2 acting 1.04e-3 2.62e- 3 3.90e-3 5.46e-4 5.01e-23 1.92e-26 6.78e-10 3.25e-24  
## 3 actor 9.65e-4 1.08e- 3 4.52e-6 4.61e-3 1.10e-13 1.67e-14 4.55e- 5 1.02e-14  
## 4 actors 4.49e-4 2.12e- 3 4.01e-3 2.17e-3 4.27e-18 3.81e-25 3.04e- 4 8.31e-22  
## 5 actress 2.38e-3 2.05e- 4 8.61e-9 1.14e-3 1.80e- 7 4.84e-15 5.08e-28 5.32e-21  
## 6 actually 3.66e-7 4.18e- 3 5.18e-4 6.77e-8 8.31e- 4 4.60e- 4 3.84e- 9 7.02e- 4  
## 7 after 7.38e-4 1.02e- 3 8.03e-5 2.76e-4 4.44e- 4 1.37e- 3 2.71e- 4 9.76e- 4  
## 8 age 1.03e-3 2.22e-12 6.43e-4 1.16e-4 1.01e- 3 2.79e-14 2.26e- 4 6.74e- 4  
## 9 all 2.30e-4 1.43e- 3 1.35e-3 3.56e-4 9.95e- 4 1.78e- 4 3.69e- 4 6.99e- 4  
## 10 almost 7.21e-4 1.14e- 3 1.77e-3 4.21e-4 3.40e- 4 3.49e- 4 5.90e- 4 8.03e- 4  
## # … with 288 more rows, and 3 more variables: topic9 <dbl>, topic10 <dbl>,  
## # log\_ratio <dbl>

tdm\_documents <- tidy(tdm\_lda, matrix = "gamma")  
tdm\_documents

## # A tibble: 500,000 × 3  
## document topic gamma  
## <int> <int> <dbl>  
## 1 1 1 0.0132   
## 2 2 1 0.0258   
## 3 3 1 0.203   
## 4 4 1 0.148   
## 5 5 1 0.248   
## 6 6 1 0.231   
## 7 7 1 0.0144   
## 8 8 1 0.00843  
## 9 9 1 0.00976  
## 10 10 1 0.110   
## # … with 499,990 more rows

ggplot(tdm\_documents, aes(gamma)) +  
 geom\_histogram(alpha = 0.6, color="darkblue", fill="lightblue") +  
 scale\_y\_log10() +  
 labs(title = "Distribution of probabilities for each topics",  
 y = "Number of documents", x = expression(gamma))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

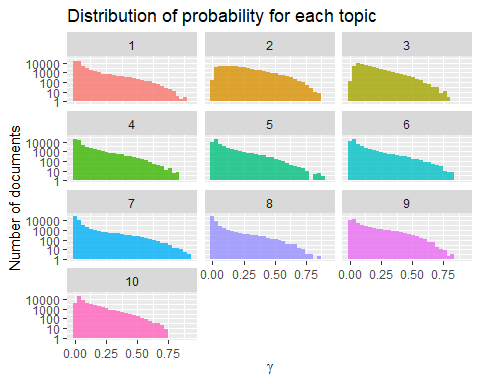


ggplot(tdm\_documents, aes(gamma, fill = as.factor(topic))) +  
 geom\_histogram(alpha = 0.8, show.legend = FALSE) +  
 facet\_wrap(~ topic, ncol = 3) +  
 scale\_y\_log10() +  
 labs(title = "Distribution of probability for each topic",  
 y = "Number of documents", x = expression(gamma))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 17 rows containing missing values (`geom\_bar()`).



tidy(tdm) %>%  
 filter(document == 6) %>%  
 arrange(desc(count))

## # A tibble: 54 × 3  
## term document count  
## <chr> <chr> <dbl>  
## 1 movie 6 2  
## 2 and 6 1  
## 3 awakening 6 1  
## 4 believable 6 1  
## 5 bette 6 1  
## 6 boring 6 1  
## 7 brings 6 1  
## 8 cause 6 1  
## 9 children 6 1  
## 10 davis 6 1  
## # … with 44 more rows