Fundamentals of Computing and Data Display

Term paper template

Tim Raxworthy & Carlos Cristiano

2022-12-06

Contents

Introduction

There are many factors that impact people's decision making processes. Much is unknown as to how people formulate their decisions when choosing support for different conservation organizations that portray themselves to be following a mission statement agreeable to their own beliefs. One major question is, "Why do people select certain organizations over others to donate money to, participate in events, subscribe to newsletters, and read their online articles? Primates other than humans are a particularly important species for measuring concepts normally overlooked as having impact in how people formulate their support for organizations. Many conservation, biological and evolutionary scientists are keen on the acute impact that primates have on human society, their importance and their role in our society's understanding of evolutionary concepts but what is not known is how current human interests surrounding primates could be influencing (directly or indirectly) our own way of designing conservation organizations.

This research aims at gaining some insight into how people view primates on social media. To do this we will be exploring and analyzing twitter data to create different topics related to our search terms (monkey, ape, chimp, primate etc.). This information will then be interpreted within the framework of what aspects of organizations that interact with primate species are "valued" or "normally expected" over others, although we are not comparing different organizations but rather different aspects surrounding a respondents current interest towards primate species. We will also compare if any of the "primate values" outlined in (**Marshall2016?**) overlap with the topics that our LDR model generates.

The fate of primate conservation has much do with human intervention whether that be positive or negative. Examining human interpretation of primates will be vital for those designing conservation projects currently and into the future. Primate conservation is valuable to humans for many different reasons, but a significant overarching reason is that primates are more similar to us than other orders of organisms on Earth. This similarity gives insight into our own species that no other animal can. It has been found that primates are excellent model animals for understanding physical and psychological illnesses that ail humans (**Estrada2017?**). Primates also possess similar cognitive abilities to humans and some captive chimpanzees have displayed a working memory that rivals that of humans (Inoue, 2007). Chimpanzees use of tools could imply that they have an understanding of causation and posses exceptional problem solving skills (Whiter, 2011). These features demonstrate some of the similarities that other primate species share with humans. This study is an exploration into what kinds of public support for primate conservation are being discussed on internet forums such as twitter. To determine this, we are building a topic model that can help distinguish and categorize these different discussions, quantifying which ones are happening at the highest frequency across tweets.

Data

The data set is extracted from the social media Twitter through the function search_tweets from the package rtweet and then saved as a data frame object. The sample size for this study corresponds to four thousand tweets; the hashtag monkey was chosen as the query to be searched and used to filter and select tweets for this study. The last parameter considered for this sample is the language of the tweets, for this study we just consider tweets in english.

```
 Data.science <- search_tweets(
   q = "monkey", # search for Tweets with "data" AND "science",
   n = 1000 ,
   lang = "en"
 )
# data = Data.science %>%
#   select(full_text) %>%
#   mutate(doc_id=seq(n())) %>%
#   data.frame()
data <- read.csv(file = "tweet.csv")
summary(data)
```

```
##        X          full_text             doc_id
##  Min.   :   1   Length:4000        Min.   :   1
##  1st Qu.:1001   Class :character   1st Qu.:1001
##  Median :2000   Mode  :character   Median :2000
##  Mean   :2000                      Mean   :2000
##  3rd Qu.:3000                      3rd Qu.:3000
##  Max.   :4000                      Max.   :4000
```

Results

This section presents the main results.

Data exploration

For this study, we generated a corpus to make the Document Term Matrix needed for the model analysis. For this, we make a Pre-processing of the dataset in the following way:

1. We converted the text to lowercase as a Word replacement and dropped Punctuation and a non-alphanumeric character. This cleaning is made with the function tokens, where we use the parameters: remove_punct, remove_numbers, and remove_symbols equal TRUE.
2. We drop commonly used words, such as 'the', 'is', 'that', 'a', etc., that would completely dominate our analysis but don't offer much insight into the text of the tweets. The cleaning is made wiht the function tokens_remove, as
3. We split the tweeter text up into individual words as tokens. This is also done with the function tokens.

4. We grouped together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. To do the lematization we use baseform_en.tsv file, which provides the lemma of the words.

```
corpus_sotu_orig <- corpus(data,
                           docid_field = "doc_id",
                           text_field = "full_text")
corpus_sotu_proc <- tokens(corpus_sotu_orig,
                           remove_punct = TRUE,
                           remove_numbers = TRUE,
                           remove_symbols = TRUE) %>%
  tokens_tolower()
lemmaData <- read.csv2("baseform_en.tsv",
                       sep="\t",
                       header=FALSE,
                       encoding = "UTF-8",
                       stringsAsFactors = F)
lemmaData = lemmaData %>%
  filter(!is.na(V1))
corpus_sotu_proc <-  tokens_replace(corpus_sotu_proc,
                                    lemmaData$V1,
                                    lemmaData$V2,
                                    valuetype = "fixed")

corpus_sotu_proc <- corpus_sotu_proc %>%
  tokens_remove(stopwords("english")) %>%
  tokens_ngrams(1)

cloud =lemmaData %>%
  group_by(V2) %>%
  mutate(freq=n()) %>%
  distinct(freq,V2) %>%
  filter(freq<40) %>%
  arrange(desc(freq))
cloud = cloud[1:200,]
wordcloud(words = cloud$V2, freq = cloud$freq, min.freq = 1,
          max.words=200, random.order=TRUE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```

After the cleaning data, we create a term-document matrix (DTM) that represents the relationship between terms and documents, where each row stands for a term and each column for a document, and an entry is the number of occurrences of the term in the document. The matrix is created with the function DFM, which constructs a sparse document-feature matrix from the above corpus. The following steps are done with dfm_trim, dfm_select, and dfm_compress. The last functions allow us to estimate the frequencies of the words, select the features of interest, and post-recombine our DFM by identical dimension elements in their respective order. The DTM is the following:

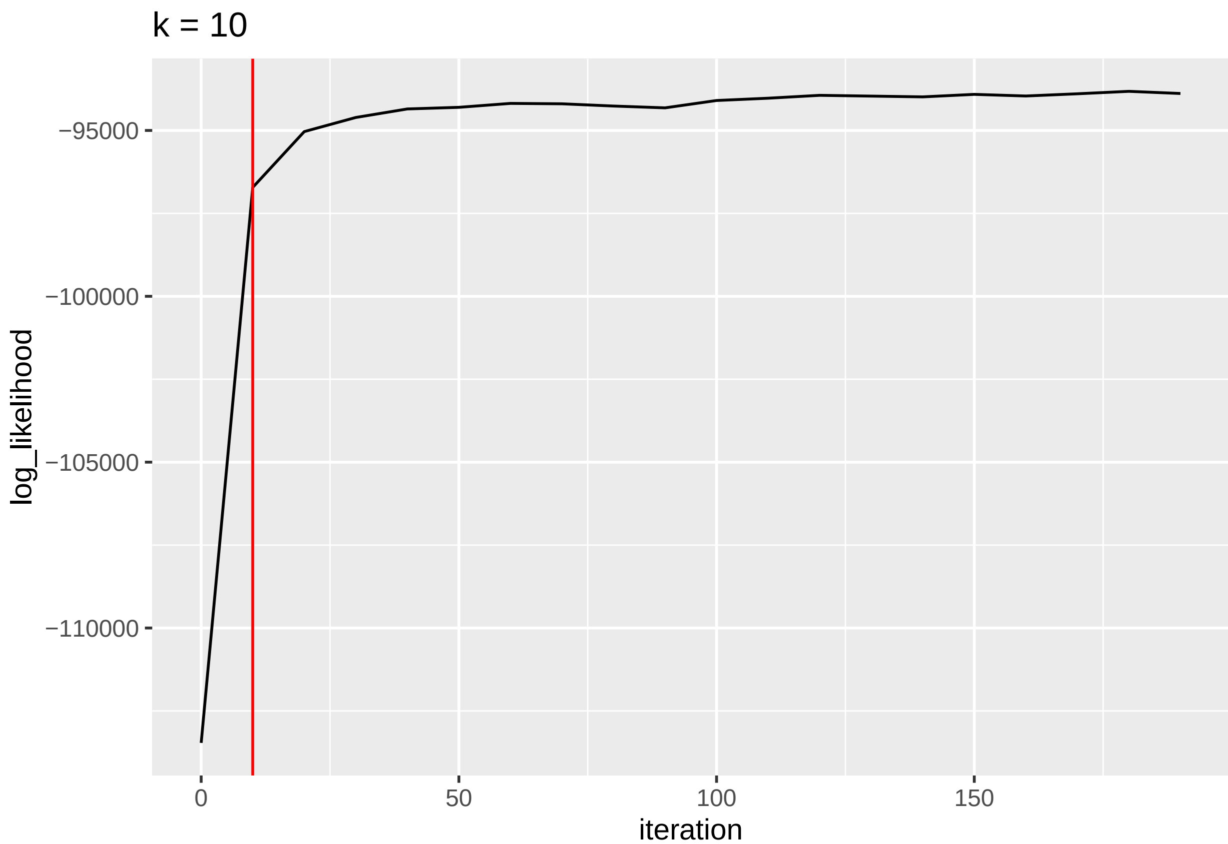
```
DTM <- dfm(corpus_sotu_proc)
minimumFrequency <- 10
DTM <- dfm_trim(DTM,
                min_docfreq = minimumFrequency,
                max_docfreq = 100)
DTM  <- dfm_select(DTM,
                   pattern = "[a-z]",
                   valuetype = "regex",
                   selection = 'keep')
colnames(DTM) <- stringi::stri_replace_all_regex(colnames(DTM),
                                                 "[^_a-z]","")

DTM <- dfm_compress(DTM, "features")
sel_idx <- rowSums(DTM) > 0
DTM <- DTM[sel_idx, ]
textdata <- data[sel_idx, ]
dim(DTM)
```

```
## [1] 2805  559
```

```
model <- FitLdaModel(dtm = DTM,
                     k = 20,
                     iterations = 200, # I usually recommend at least 500 iterations or more
                     burnin = 180,
                     alpha = 0.1,
                     beta = 0.05,
                     optimize_alpha = TRUE,
                     calc_likelihood = TRUE,
                     calc_coherence = TRUE,
                     calc_r2 = TRUE,
                     cpus = 2)
```

```
## dtm is not of class dgCMatrix, attempting to convert...
```

```
model2=as.data.frame(model$log_likelihood)
ggplot(model2,aes(x=iteration,y=log_likelihood))+
  geom_line()+
  geom_vline(xintercept = 10, col="red")+
  labs(title = "k = 10")
```




```
K <- 10
```

```r
topicModel <- LDA(DTM,
                  K,
                  method="Gibbs",
                  control=list(iter = 500,
                               verbose = 25))
```

```
## K = 10; V = 559; M = 2805
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

```r
tmResult <- modeltools::posterior(topicModel)
beta <- tmResult$terms

theta <- tmResult$topics

#terms(topicModel, 10)
top5termsPerTopic <- terms(topicModel,
                           5)
# For the next steps, we want to give the topics more descriptive names
#than just numbers. Therefore, we simply concatenate the five most likely
#terms of each topic to a string that represents a pseudo-name for each topic.
topicNames <- apply(top5termsPerTopic,
                    2,
                    paste,
                    collapse=" ")
topicProportions <- colSums(theta) / nrow(DTM)  # average probability over all paragraphs
names(topicProportions) <- topicNames     # Topic Names
sort(topicProportions, decreasing = TRUE)
```

```
##                  les cest il et mais        neuralink brain human play guy
##                           0.10460629                            0.10006899
##                 can will back think u                know make thing big d
##                           0.09988028                            0.09982997
```

```
##     black test cup starbucks employee           amp year come release today
##                            0.09961446                            0.09938810
##               elon musk give new every            show island put ah video
##                            0.09937138                            0.09918464
## november early fall da pdxjapanesegdn                   du est hour late y
##                            0.09912081                            0.09893508
```

```r
attr(topicModel, "alpha")
```

```
## [1] 5
```

```r
topicModel2 <- LDA(DTM,
                   K,
                   method="Gibbs",
                   control=list(iter = 500,
                                verbose = 25,
                                alpha = 0.2))#replace alpha
```

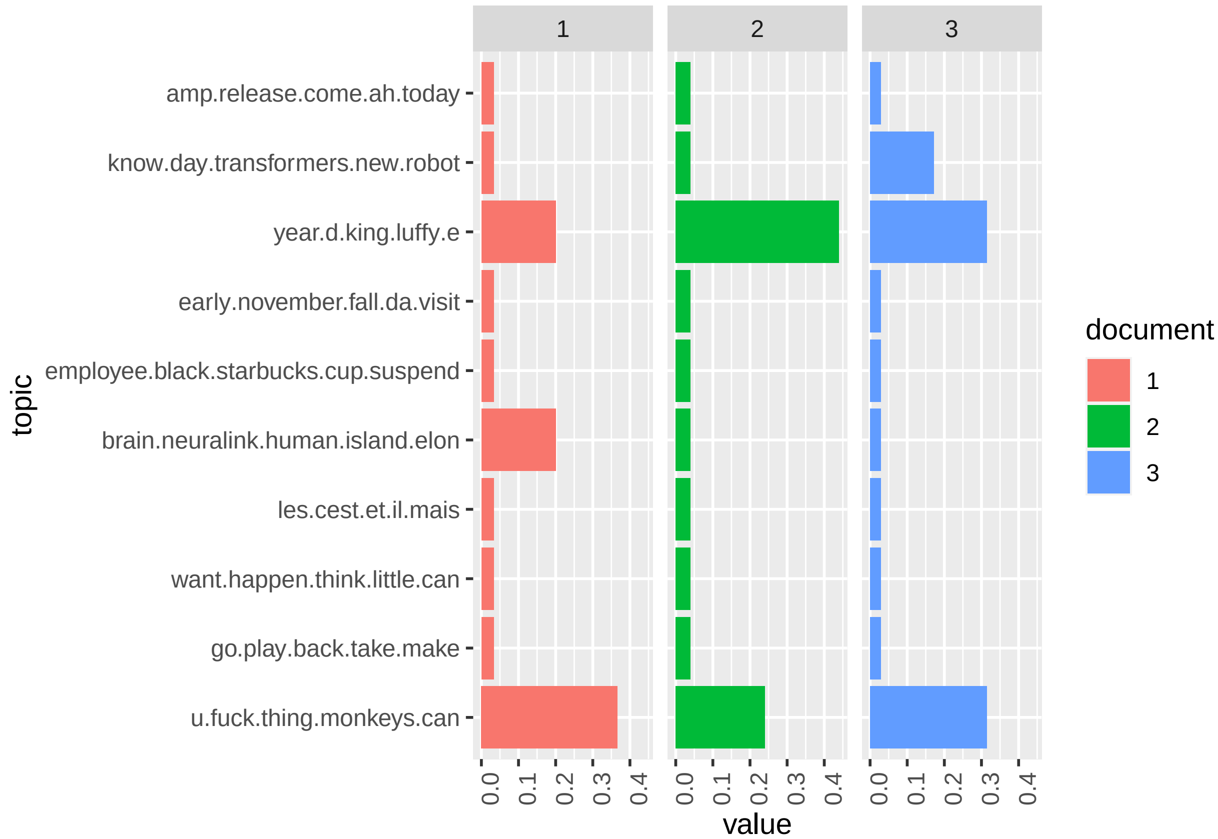
```
## K = 10; V = 559; M = 2805
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

```r
tmResult <- modeltools::posterior(topicModel2)
theta <- tmResult$topics
beta <- tmResult$terms

topicProportions <- colSums(theta) / nrow(DTM)  # average probability over all paragraphs
names(topicProportions) <- topicNames     # Topic Names
sort(topicProportions, decreasing = TRUE)
```

```
##               show island put ah video november early fall da pdxjapanesegdn
##                             0.14706050                            0.11811061
```

```
##           amp year come release today                  know make thing big d
##                            0.11195308                             0.11099546
##                  les cest il et mais                       du est hour late y
##                            0.10748245                             0.09365338
##               elon musk give new every     black test cup starbucks employee
##                            0.09348997                             0.07651576
##                 can will back think u        neuralink brain human play guy
##                            0.07134275                             0.06939602
```

```
topicNames <- apply(terms(topicModel2, 5), 2, paste, collapse = " ")
exampleIds <- c(2, 100, 200)
N <- length(exampleIds)

topicProportionExamples <- as.tibble(theta) %>%
  slice(exampleIds)

colnames(topicProportionExamples) <- topicNames

vizDataFrame <- melt(cbind(data.frame(topicProportionExamples),
                           document = factor(1:N)),
                     variable.name = "topic",
                     id.vars = "document")

ggplot(data = vizDataFrame,
       aes(topic, value,
           fill = document),
       ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90,
                                   hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document,
             ncol = N)
```




```
# What happens here depends on the specific project
```

```
# What happens here depends on the specific project
```

Discussion

This section summarizes the results and may briefly outline advantages and limitations of the work presented.

References