Fundamentals of Computing and Data Display

Term paper template

Tim Raxworthy & Carlos Cristiano

2022-12-05

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.

```
Data.science <- search_tweets(
  q = "monkey", # search for Tweets with "data" AND "science",
  n = 4000
)
data = Data.science %>%
  select(full_text) %>%
  mutate(doc_id=seq(n())) %>%
  data.frame()
summary(data)
```

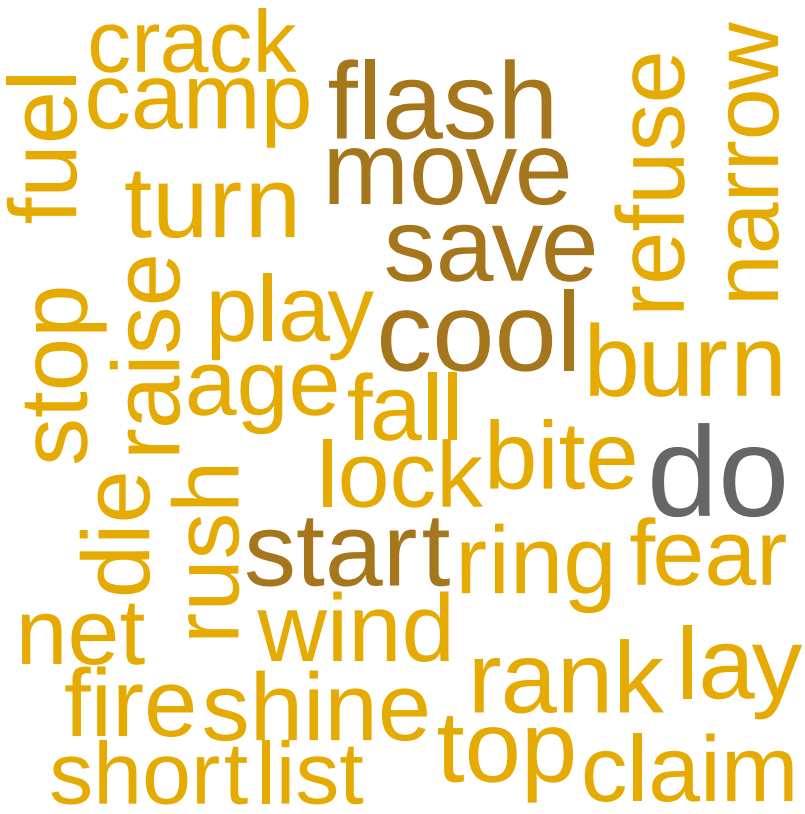
```
##   full_text            doc_id
##  Length:3441        Min.   :   1
##  Class :character   1st Qu.: 861
##  Mode  :character   Median :1721
##                     Mean   :1721
##                     3rd Qu.:2581
##                     Max.   :3441
```

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.
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.
3. We split the tweeter text up into individual words as tokens.
4. We identify the end bits of words that can be chopped off to leave the core root of the word. This process is called Stematization. Lemmatisation in linguistics is the process of grouping 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.
5. 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.

```
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))
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"))
```



Results

This section presents the main results.

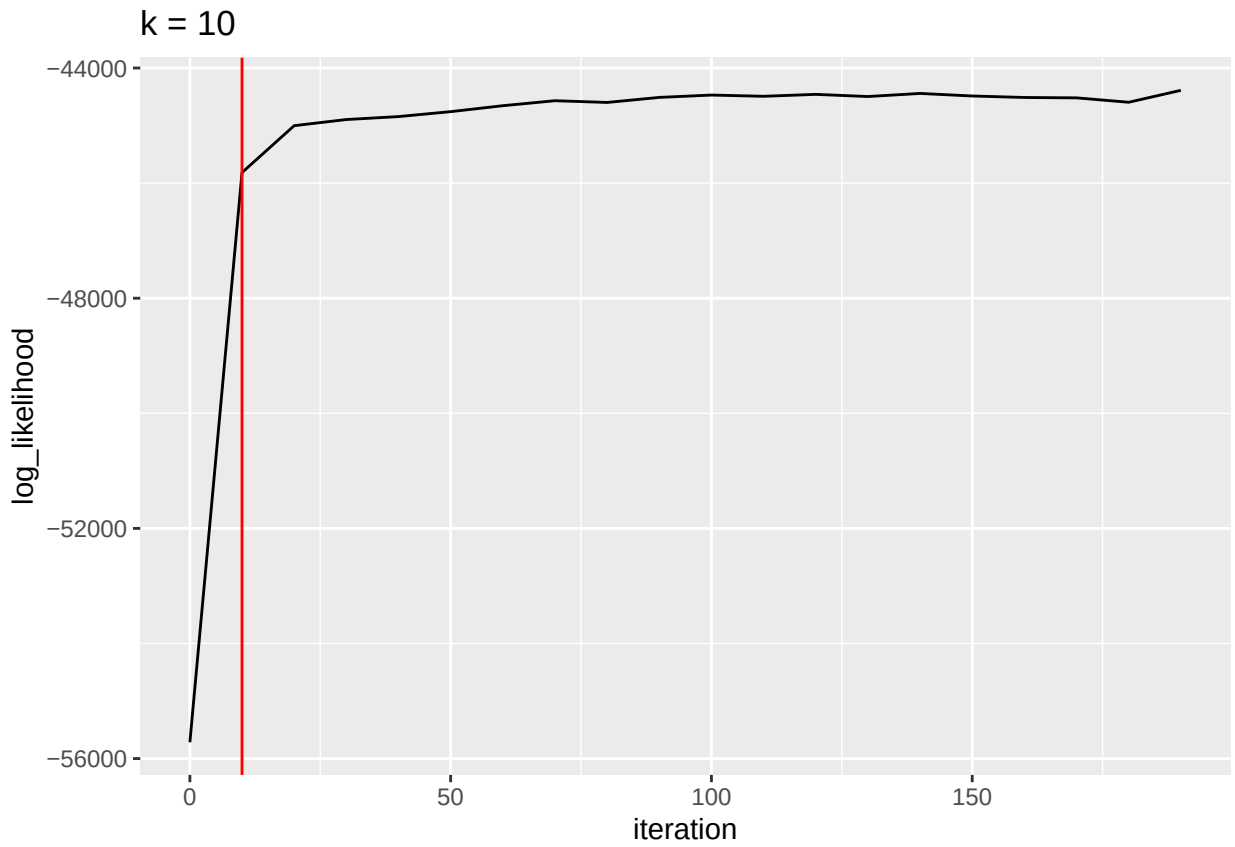
Data exploration

The results section may have a data exploration part, but in general the structure here depends on the specific project.

```
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)
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, ]

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

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

```
## K = 10; V = 323; M = 1819
## 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!
```

```
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)
```

```
## annekari_ latte_honmono daisenpai_sanyi hima_nandaga etarinn
##                                                  0.10543125
##                                     call good year never say
##                                                  0.09987983
##                           play takumin_monkey look use ortega
##                                                  0.09961824
##                                           w take day tweet h
##                                                  0.09948436
##                                         get go amp love even
##                                                  0.09947640
##                                           gt one see think d
##                                                  0.09936642
##                  annaclemm man httpstcomnnxcntc trump suggest
##                                                  0.09934188
##                       enjoy full golden orange buitengebieden
##                                                  0.09920793
##                                         just can pox now box
##                                                  0.09909987
##                                u monkey_i_am_ time ansiale en
##                                                  0.09909381
```

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

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

```
topicModel2 <- LDA(DTM,
                   K,
```

```
                  method="Gibbs",
                  control=list(iter = 500,
                               verbose = 25,
                               alpha = 0.2))#replace alpha
```

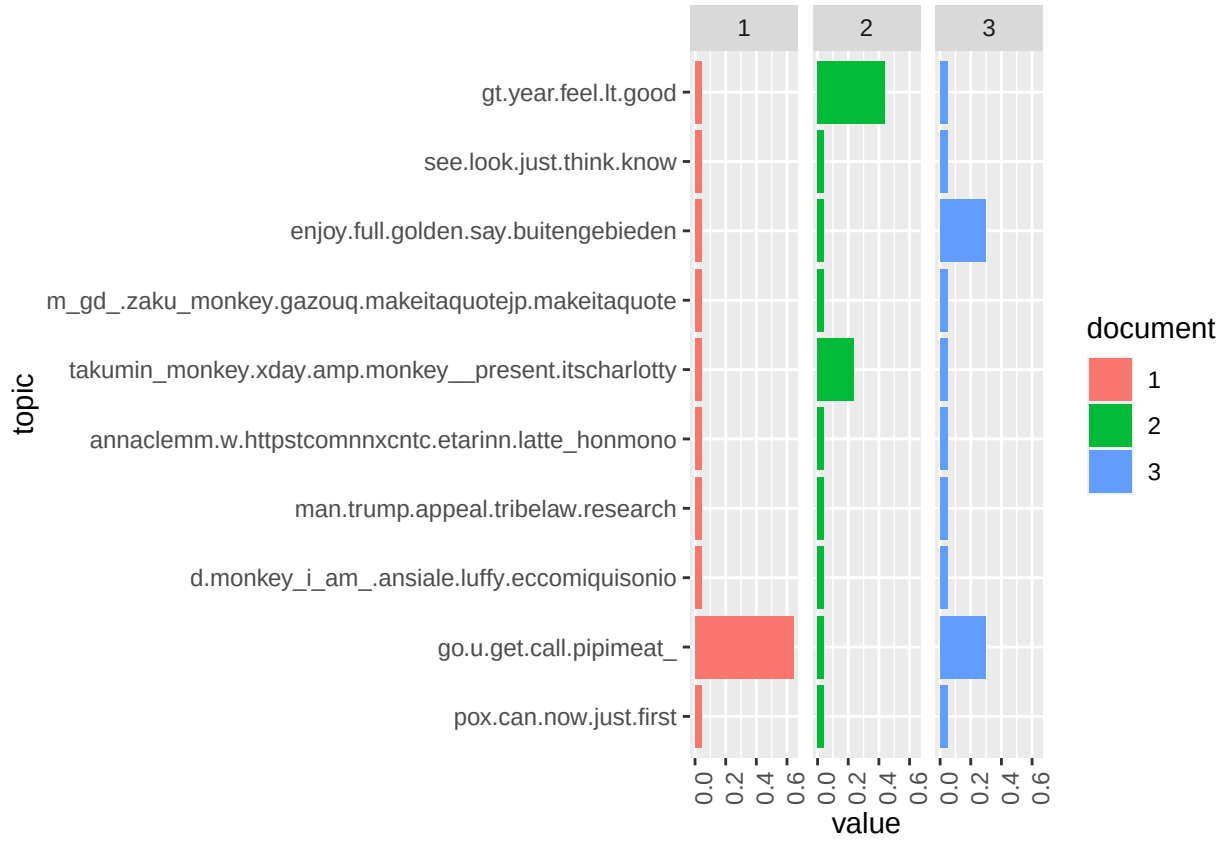
```
## K = 10; V = 323; M = 1819
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

```
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)
```

```
##                                    call good year never say
##                                               0.13556163
##                                u monkey_i_am_ time ansiale en
##                                               0.11220387
##                            play takumin_monkey look use ortega
##                                               0.11113086
##                                          gt one see think d
##                                               0.10505305
##                                       get go amp love even
##                                               0.10420214
## annekari_ latte_honmono daisenpai_sanyi hima_nandaga etarinn
##                                               0.09874417
##                                        w take day tweet h
##                                               0.09669123
##                                       just can pox now box
```

```
##                                                 0.08737086
##                     enjoy full golden orange buitengebieden
##                                                 0.08171293
##                 annaclemm man httpstcomnnxcntc trump suggest
##                                                 0.06732926
```

```
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
```

Analysis

This section presents the main results, such as (for example) stats and graphs that show relationships, model results and/or clustering, PCA, etc.

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

```
# 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