Supervised LDA Module

Linh Phan

2/8/2023

Lab Objective

1. Better understand LDA
2. Learn how to use the topicmodels package
3. Attempt our own LDA

Supervised Latent Dirichlet Allocation

Topic modeling is a type of statistical modeling for discovering the abstract "topics" that occur in a collection of documents. Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. * The basic idea of LDA is to take a number of features and condense them town to a few "topics."

Associated Press Example

For this module will be do LDA topic modeling on articles from the Associated Press. To start, we need to install and load the topicmodels package. The Associated Press data is already loaded in our topicmodels package.

```
# install.packages("topicmodels")
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 4.1.2
```

```
data("AssociatedPress")
```

After loading the data, we need to specify the value of k, or the number of topics in the corpus. For this dataset, which is larger and more diverse, I am going to select a larger topic number, in this case the k = 10. * A common mistake is choosing the wrong topic number, this can lead to over-generalization or over-specialization of the model.

```
AP_topic <- LDA(AssociatedPress, k=10, control = list(seed = 321))
```

The control argument is used to seed the assignment of topics to each word in the corpus. The control argument is also used to specify a number of different options as well, such as the maximum number of iterations that we want our topic model to perform.

Examine Terms within Topics

To interpret the topics, we can start by seeing which terms are most probable for each topic.

```
get_terms(AP_topic, 10)
```

```
##       Topic 1   Topic 2     Topic 3    Topic 4      Topic 5      Topic 6
##  [1,] "percent" "soviet"    "iraq"     "government" "police"     "year"
##  [2,] "million" "united"    "military" "party"      "people"     "percent"
##  [3,] "billion" "states"    "air"      "political"  "two"        "new"
##  [4,] "year"    "west"      "force"    "president"  "killed"     "people"
##  [5,] "market"  "east"      "two"      "soviet"     "government" "children"
##  [6,] "new"     "german"    "iraqi"    "new"        "south"      "years"
##  [7,] "stock"   "union"     "american" "opposition" "army"       "health"
##  [8,] "prices"  "germany"   "united"   "minister"   "man"        "last"
##  [9,] "company" "president" "kuwait"   "national"   "death"      "aids"
## [10,] "last"    "world"     "plane"    "people"     "three"      "study"
##       Topic 7  Topic 8     Topic 9    Topic 10
##  [1,] "i"      "workers"   "court"    "bush"
##  [2,] "people" "people"    "case"     "house"
##  [3,] "years"  "officials" "attorney" "dukakis"
##  [4,] "dont"   "new"       "judge"    "president"
##  [5,] "think"  "water"     "federal"  "senate"
##  [6,] "school" "fire"      "trial"    "bill"
##  [7,] "just"   "two"       "charges"  "campaign"
##  [8,] "new"    "city"      "state"    "committee"
##  [9,] "going"  "miles"     "law"      "budget"
## [10,] "time"   "area"      "drug"     "republican"
```

We see some overlap in these topics for several terms, such as “government,” and “president,” This is not surprising given that we are looking at a news outlet dataset.

Using the tidytext package we can produce bar grpahs that describe the topic term for each topic.

```
library(tidytext)
```

```
## Warning: package ’tidytext’ was built under R version 4.1.2
```

```
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.1.2
```

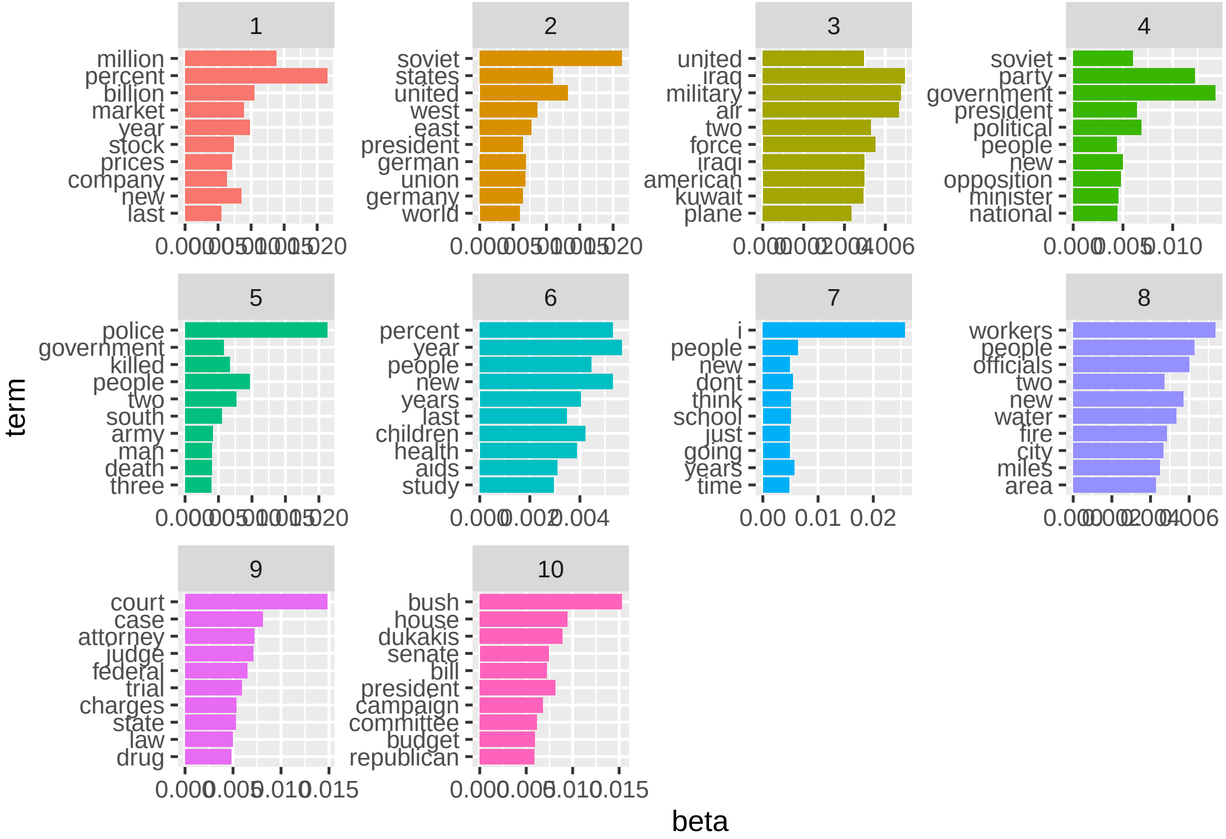
```
library(ggplot2)
```

```
## Warning: package ’ggplot2’ was built under R version 4.1.2
```

```
AP <- tidy(AP_topic, matrix = "beta")

ap_top_terms <-
  AP %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)


ap_top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



References:

Chris Bail: https://github.com/cbail/textasdata