Machine Learning for Social Science - Lab 4

Marc Sparhuber

Table of contents

Part 1: Topic modeling

Task 1

Begin by importing fb-congress-data3.csv. Report basic information about the data set; how many rows and column it has, as well as the name of the variables.

```
[1] 6752    4
```

> 6752 rows and 4 variables. It contains screen names, the persons party and their documents based on facebook messages.

Task 2

As you may have noticed from your inspection in #1, this data set has yet to be pre-processed (it contains punctuation, etc.). Hence, that is what you shall do now. More specifically, perform the following steps:

i.

Use quanteda's corpus() function to create a corpus of your data set. Hint: For the argument x select your data set, for the argument text select the column name which stores the text, for the argument docid_field select the id variable, and finally, add the names of remaining variables to the meta argument (in a list).

ii.

Tokenize your corpus using the tokens() function. This splits each document into a vector of so-called tokens. Make the following specifications (which will remove punctuation, numbers, non-alpha-numeric symbols, and urls): • remove_punct = TRUE • remove_numbers = TRUE • remove_symbols = TRUE • remove_url = TRUE • padding = FALSE

iii.

Exclude english stopwords using the tokens_remove() function. Setting x to the output from the previous step, setting the second argument to stopwords("en"), and setting padding=FALSE.

iv.

To get a feel of how your data looks like now, print the first 3 texts by simple subsetting of the output from iii.

v.

As mentioned in the lecture, topic models expect the data to be in a document-term-matrix form. Transform your tokens into a document-term-matrix using the quanteda's function dfm().

vi.

As a last pre-processing step, we want to exclude (a) words which are very infrequent (below 5). and (b) documents which have very few words (below 10). When you have done a–b, report the dimensionality of your resulting document-term-matrix. Hint: To do trim infrequent words, use quanteda's function dfm_trim(). To exclude documents with too few words, you may use the following code (where dtm is the object in which you have stored your document-term-matrix):

Task 3

Now we are ready to do some topic modeling! To do so, we will use the topicmodels package, and the function LDA(). Set x to your document-term-matrix and specify method="Gibbs" (note: Gibbs is the name of a particular estimation procedure; see the Appendix of the lecture for more details). Set the number of iterations to 1000, and specify a seed number to ensure replicability (hint: to specify iterations and seed number, use the control argument). Finally, set the number of topics, K = 50. With these settings specified, start the estimation. This could take a minute or two.

```
K = 50; V = 5484; M = 5485
Sampling 1000 iterations!
Iteration 100 ...
Iteration 200 ...
Iteration 300 ...
Iteration 400 ...
Iteration 500 ...
Iteration 600 ...
Iteration 700 ...
Iteration 800 ...
Iteration 900 ...
Iteration 1000 ...
Gibbs sampling completed!
```

Task 4

Once the estimation is finished, use the get_terms() function to extract the 15 words with the highest probability in each topic. In a real research setting, we would carefully examine each of the topics. Here, I only ask you to briefly skim them, and then focus on 5 that

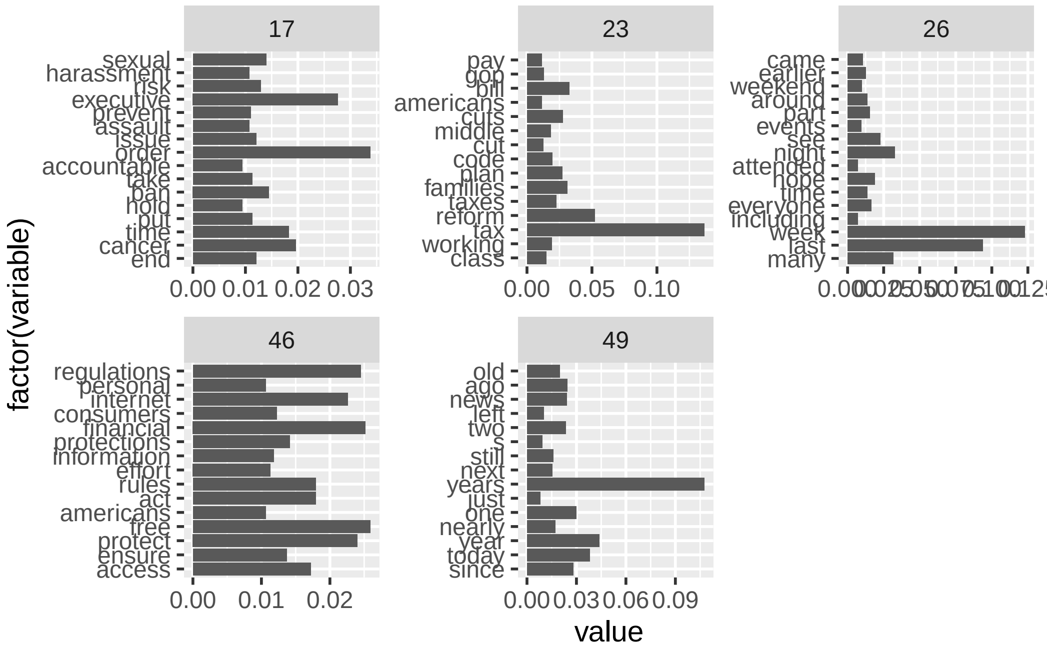
(i) you think are interesting,
(ii) has a clear theme, and
(iii) are clearly distinct from the other topics.

Provide a label to each of those based on the top 15 words. Complementing your label, please also provide a bar chart displaying on the y-axis the top 15 words, and on the x-axis their topic probabilities. Hint: you can retrieve each topic's distribution over words using topicmodels's function posterior.

Lastly, please also report a general assessment—based on your skim—about the general quality of the topics; do most of them appear clearly themed and distinct, or are there a lot of "junk" topics?

```
      Topic 17      Topic 23    Topic 26    Topic 46      Topic 49
 [1,] "order"       "tax"       "week"      "free"        "years"
 [2,] "executive"   "reform"    "last"      "financial"   "year"
 [3,] "cancer"      "bill"      "night"     "regulations" "today"
 [4,] "time"        "families"  "many"      "protect"     "one"
 [5,] "ban"         "cuts"      "see"       "internet"    "since"
 [6,] "sexual"      "plan"      "hope"      "act"         "ago"
 [7,] "risk"        "taxes"     "everyone"  "rules"       "news"
 [8,] "end"         "code"      "part"      "access"      "two"
 [9,] "issue"       "working"   "time"      "protections" "old"
[10,] "put"         "middle"    "around"    "ensure"      "nearly"
[11,] "take"        "class"     "earlier"   "consumers"   "still"
[12,] "prevent"     "gop"       "came"      "information" "next"
[13,] "assault"     "cut"       "weekend"   "effort"      "left"
[14,] "harassment"  "americans" "events"    "americans"   "s"
[15,] "hold"        "pay"       "including" "personal"    "just"
```

Topics: - 17 new energy - 23 drug epidemic - 26 health care - 46 state of american economy - 49 foreign aid



The topics seem to be fairly distinct, not a lot of "junk" topics, though there are a few.

Task 5

Out of the 5 topics that you labeled, select two which you think are particularly interesting. For these two, identify the three documents which have the highest proportion assigned of this topic (hint 1: use topicmodels's posterior() to extract documents' distribution over topics | hint 2: to identify the document ids which correspond to each row of what you extract from posterior(), you can use ldaobject@documents. See help file for more details.), and do a qualitative inspection (= 2 × 3 documents to read). Does your readings corroborate your labels? Are they about what you expected?

Actually, they don't and I think this might make me reconsider my labels for the topics.

Task 6

Now, estimate a topic model—as in #3—but with K=3 instead. Extract the top 15 words from each topic, (try to) label each, and then make an assessment of the overall quality of them. To further explore the quality of this topic model, reconsider the documents you read in #5: extract the distribution over topics for these documents (from your new K=3 model).

How well does this topic model capture the theme of these documents? Based on your analysis, which of the two K's do you prefer? Motivate.

```
K = 3; V = 5484; M = 5485
Sampling 1000 iterations!
Iteration 100 ...
Iteration 200 ...
Iteration 300 ...
Iteration 400 ...
Iteration 500 ...
Iteration 600 ...
Iteration 700 ...
Iteration 800 ...
Iteration 900 ...
Iteration 1000 ...
Gibbs sampling completed!
```

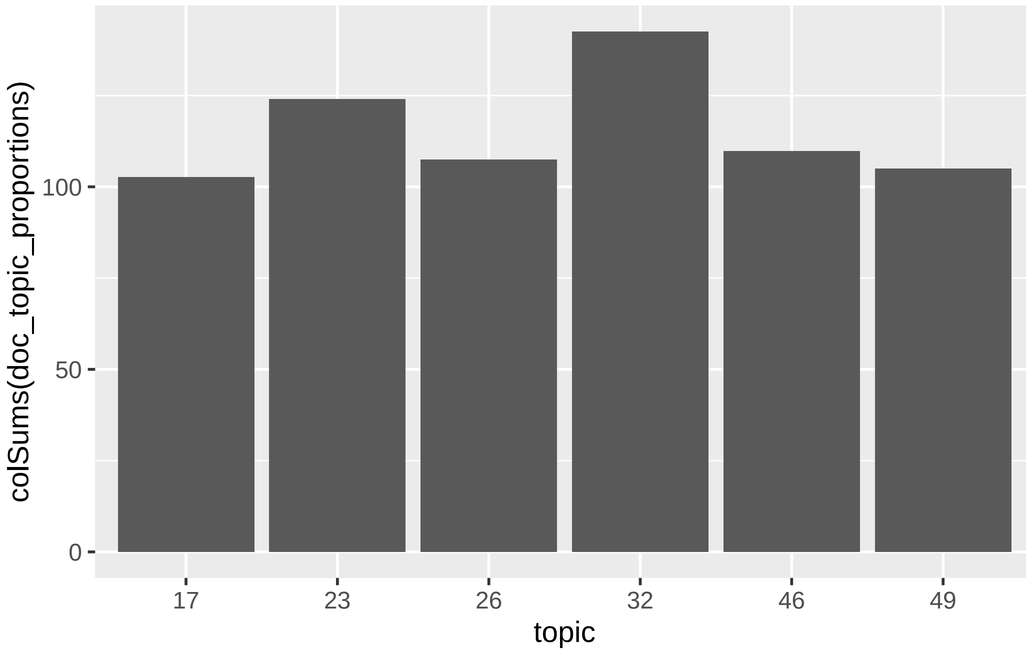
> To label these is pretty difficult as they are so broad that it has become difficult to give them discerning labels - in this case looking at the documents with the highest proportions also does not offer any clarity. Therefore, I definitely prefer the k = 50 model, though I do think a good k would lie somewhere in the middle.

Task 7

Continuing with the topic model you concluded the most appropriate, perform the following sets of analyses:

i.

Compute the prevalence of each topic, across all documents. Report which is the most prevalent topic, overall, and then report—in the form of a single plot; e.g., a bar chart—the prevalence of the topics you labeled.



> Topic 32 is the most prevalent overall. 23 is the second most prevalent one and one that I chose earlier. The other topics I chose trail behind these two.

ii

Compare the prevalence on your labeled topics between democrats and republicans. You can for example fit a fractional regression model using glm(family="quasibinomial") or using t-tests of difference in means. Interpret.

> Apart from topic 23 the topic prevalences of Democrats and Republicans are significantly different across the topics I picked. Topic 23 was possibly misidentified by me and actually be about cancer and since this is not a topic that divides the political lines much, it seems feasible that there is not significant difference in how democracts and republicans talk about this issue.

Part 2: Word embeddings

Task 1

Because word embeddings are not negatively affected by stop words or other highly frequent terms, your first task is to re-import the fb-congress-data3.csv file, and re-process the data; performing step i–ii in task #2, but skipping #3. Here, we also do not want to transform our documents into a document-term matrix. Instead, after having tokenized and cleaned the documents, paste each back into a single string per document. Hint: for this, you could for example write: sapply(mytokens,function(x)paste(x,collapse = " ")). As a last pre-processing step, transform all your text into lowercase (hint: you can use the function tolower() for this).

Task 2

Now we are set to fit word embeddings! To begin, let us fit one word embedding model to all documents—not separating posts by democrats and republicans. Use word2vec's word2vec() function to fit a cbow model (type="cbow") using 15 negative samples per real context/observation (negative=15), and setting dim=50, the number of dimensions of the word vetors/embeddings. This will take a minute or two.

```
   user  system elapsed 
  35.09    0.00   35.11 
```

Task 3

When the estimation in #2 is finished, identify the 10 nearest terms to 3 focal words of your choice/interest. Make sure to select words which occur frequently in your data. Hint: to retrieve the closest words in embedding/word vector space, you may use the following code: predict(w2v,c("word2","word2","word3"),type="nearest",top_n = 10), where wv2 is the object storing the fitted model of the word2vec function. Does the results you find makes sense? Why/why not?

```
$worker
    term1      term2 similarity rank
1  worker      child  0.7227702    1
2  worker      civil  0.7225662    2
3  worker    workers  0.7064255    3
4  worker indonesian  0.6989232    4
5  worker     racial  0.6771286    5
6  worker astronauts  0.6647820    6
7  worker    founder  0.6594008    7
8  worker  employees  0.6558373    8
9  worker  mountains  0.6554101    9
10 worker  discharge  0.6550218   10

$energy
    term1          term2 similarity rank
1  energy      investing  0.7572529    1
```

```
2  energy           rail  0.7258346    2
3  energy             fy  0.7247275    3
4  energy         supply  0.7118220    4
5  energy transportation  0.7108954    5
6  energy          aging  0.7043882    6
7  energy       economic  0.6966298    7
8  energy     increasing  0.6788267    8
9  energy     innovators  0.6761596    9
10 energy     innovation  0.6745971   10

$drug
   term1      term2 similarity rank
1   drug     opioid  0.7580397    1
2   drug     heroin  0.7432082    2
3   drug prevention  0.7295574    3
4   drug    lenders  0.7249255    4
5   drug        hiv  0.7151387    5
6   drug  substance  0.7113166    6
7   drug  hazardous  0.7089950    7
8   drug    violent  0.7045968    8
9   drug    nuclear  0.7021890    9
10  drug     misuse  0.7021416   10
```

I think these results make sense generally, though especially the results for worker are a bit perplexing, with jews having the highest similarity.

Task 4

What initially made people so excited about word embeddings was their surprising ability to solve seemingly complex analogy tasks. Your task now is to attempt to replicate one such classical analogy result, first with the embedding vectors that you have already estimated, and second using a pre-trained embedding model. To do so, please perform the following steps:

i.

Extract the whole embedding matrix: embedding <- as.matrix(w2v).

ii.

Identify the rows in the embedding matrix which correspond to king, man, woman, and create a new R object kingtowoman which is equal to the vector for king, minus the vector for man, plus the vector for woman. Hint: to extract the row corresponding to a particular word (e.g., “king”), you may use w2v[rownames(w2v)==“king”,].

iii.

Use word2vec’s function word2vec_similarity() to identify the 20 most similar words to king-towoman. Do you find “queen” in the top 20? Why do you think you get the result you do?

> “queens” is not in the top 20. However, king and woman are quite high up. Might be due to the corpus size being too small and royalty might not be discussed very often in US politics.

iv.

Next, we will consider a pre-trained embedding model (trained on all Wikipedia articles that existed in 2014 and about 5 million news articles). The embedding vectors from this model are stored in the file “glove6B200d.rds”.[5] Note: this file is large; more than 300MB. Use readRDS() to import it, and stored it in an R object called pretrained. Each row stores the embedding vector for a particular word. With this info in mind, report how many embedding dimensions were used for this model, and how many words we have embedding vectors for.

> We have 400k words with 200 dimensions each, though some “words” are not actually words but artefacts.

v.

Repeat steps ii–iii for pretrained. Does “queen” appear in the top 20 here? What do you think explains this difference/similarity to the self-trained result?

> King is still first but followed by queen in second place and other words describing royal titles. This is due to us now having larger dimensions and a different corpus that was trained on more general content that would more likely contain all three words we’re looking for.

vi.

Given the result in (v), what do you expect, if you were to construct a measure of occupational gender bias along the lines of Garg et al. (2018), that is by comparing the distance between different occupations and gendered words, for example: $\overrightarrow{\text{occupational bias}} = \text{dist}(\overrightarrow{\text{statistician}}, \overrightarrow{\text{man}}) - \text{dist}(\overrightarrow{\text{statistician}}, \overrightarrow{\text{woman}})$, would this score be “more correct” than the one you would obtain from the same calculation on your facebook/congress model? Why/why not?

From my intuition and without having read that paper I would assume that the facebook model would not yield very good results because statistician may not appear very often in the underlying corpus.

Task 5

Now we shall make a comparison between democrats and republicans. Split the data from step #1 into two based on party affiliation. Then, repeat 2–3, but now separately for republicans and democrats. For #3, select words which you expect might be used differently between the two political camps (but still are frequently used by both; for example "abortion", "obamacare"). Do you find any differences? Do they align with your expectations?

```
   user  system elapsed
  15.82    0.00   15.83

   user  system elapsed
  17.03    0.00   17.05

$energy
    term1         term2 similarity rank
1  energy         solar  0.7173687    1
2  energy        markup  0.7159953    2
3  energy        mining  0.7085263    3
4  energy       regular  0.7023969    4
5  energy      economic  0.6947049    5
6  energy          ease  0.6804765    6
7  energy capabilities  0.6761994    7
8  energy        reduce  0.6666977    8
9  energy    production  0.6664684    9
10 energy           dod  0.6651874   10

$drug
   term1        term2 similarity rank
1   drug       opioid  0.8360588    1
2   drug prescription  0.8008302    2
3   drug    substance  0.7690840    3
4   drug        abuse  0.7267777    4
5   drug        drugs  0.7137839    5
6   drug      violent  0.7087275    6
7   drug    addiction  0.7077022    7
8   drug    increases  0.7060919    8
```

```
9   drug           fda  0.6975871    9
10  drug        heroin  0.6932911   10

$energy
    term1            term2 similarity rank
1  energy     technologies  0.7875090    1
2  energy        renewable  0.7651994    2
3  energy      sustainable  0.7128788    3
4  energy          sources  0.6978495    4
5  energy            water  0.6950299    5
6  energy   infrastructure  0.6845268    6
7  energy responsibilities  0.6807579    7
8  energy   appropriations  0.6756822    8
9  energy         creation  0.6731221    9
10 energy        standards  0.6706706   10

$drug
   term1        term2 similarity rank
1   drug prescription  0.7433769    1
2   drug        drugs  0.7208377    2
3   drug     monument  0.7088599    3
4   drug       prices  0.6975424    4
5   drug     lifeline  0.6445686    5
6   drug    reduction  0.6409312    6
7   drug        small  0.6402814    7
8   drug          gap  0.6360195    8
9   drug         debt  0.6354303    9
10  drug     payments  0.6349630   10
```

First of all it's telling that worker doesn't even occur in any of the messages by the republicans. This is likely because this term is "left-loaded" politically speaking. Energy meanwhile reflects ongoing political trends in the US with the dems leaning more renewable and sustainable technologies, while the republicans highlight other factors such as agriculture and mining, focusing less on innovation. Meanwhile the parties do not differ as much when talking about drugs. The opioid pandemic and its causes are highlighted by both camps though there seems to be a greater lean of the dems toward the personal outcomes associated with drugs, whereas the republicans seems to connect it more to economic factors associated with medicine manufacturers. This might also point toward the word "drug" meaning not just addictive substance but also medicine in the US context which might hamper the strength of the word embedding approach.