BAN432 fall 2019 - Third assignment

Goal of this assignment

This assignment is based on the insight of Hoberg, G. and Phillips, G. (2016) "Text-based network industries and endogenous product differentiation". The task focuses on the implementation of a network analysis in a portfolio choice problem. The goal of this assignment is to evaluate the impact of term weighting schemes and filters applied on the raw text data. Based on a textual analysis, you will compose four portfolios, which performance is measured against a reference portfolio.

We touched on almost all used functions in our last lectures and the previous assignment. If you are unsure how to use a function, either use `R`'s own documentation (type `?function.name()`) or `www.stackoverflow.com`. We encourage you to code this assignment yourselves, and do not use purpose made solutions or packages as provided on the internet.

Please submit your assignment even though you were not able to find a solution to all tasks.

Learning outcomes

In this assignment you will apply the skills you acquired in the lectures on document clustering. Choises you make while preprocessing the data will have an impact on the results of the anlysis. The goal is to compose a portfolio that is tracking the development of a reference portfolio as close as possible. You will evaluate the impact of four factors on the composition of the tracking portfolio: weighting/no weighting of term frequencies and using all words vs. only using nouns and adjectives.

Formalities

This assignment will be handed out on 15 October, 2019 at 16:00 and has to be submitted no later than 22 October, 2019 at 14:00. Submit your commented coding file and a pdf including the numerical results as well as answers to questions posted. You do not need to describe your coding in the pdf file. Please comment your code in the .R file shortly so that the grader can reconstruct your thinking. You do not need to explain the used functions.

Please work together in groups of four and submit the assignment via Canvas.

In order to solve the tasks you need to download the file `data_for_assignment_3.Rdata` from Canvas.

Task 1: Network analysis in a portfolio choice problem

It is the 31.12.2013 and you are a portfolio manager constructing a portfolio for 2014. You can allocate 1 million NOK across the 500 companies available to you. You are convinced that oil companies will perform very well in 2014. Unfortunately, regulation prevents you to invest a single NOK in this sector, in your data defined as (`industry.fama.french.49 == 30 Oil` in the raw.data data frame). Luckily, from Hoberg G. and Phillips G. (2016) you know that industry assignment is not perfect and that there might be firms which business is actually in the oil sector while not classified as such.

You develop the following strategy using insights from textual analysis: You identify a subset of firms that, based on their business description in their annual report, sound like oil firms even though they are not classified as such. You invest an equal amount in each of those firms.

In the data file you find:

- `raw.data`: a data frame with meta information about the firms, such as CIK, industry classification, and monthly returns
- `section.1.pos`: a data frame that contains all business descriptions of the firms in `raw.data`, tagged for part-of-speech.

The basic procedure for your analysis is as follows:

1. Identify the *oil firms* in the data set and paste the tokens of all oil firm's business descriptions into one large character string. Make two variations:

   (a) Consider all word classes
   (b) Consider only nouns and adjectives (Part-of-speech-tags: NN, NNS, NNP, NNPS, JJ, JJR, JJS)

2. Identify the *non-oil firms* in the data set and for each document, paste the tokens into one string. This should result in a vector with 481 elements, the non-oil firm's business descriptions. Make the same two variations as in step 1.

   (a) Consider all word classes
   (b) Consider only nouns and adjectives (Part-of-speech-tags: NN, NNS, NNP, NNPS, JJ, JJR, JJS)

3. Concatenate the corresponding vectors with tokens from all word classes (1a) and (2a) into one vector and the ones with only nouns and adjectives (1b) and (2b) into another. These vectors should have 482 elements each now. Construct a set of four DocumentTermMatrixes based on these two vectors. Make and explain your choices regarding punctuation, stopwords, numerics and word lengths. Only consider terms that appear in minimum 5 and maximum 50 documents. You create these DTMs:

   (a) a DTM based on all word classes with a weighting and normalization as in Hoberg/Phillips (2016)
   (b) a DTM based on all word classes without any weighting/normalization
   (c) a DTM based on only nouns and adjectives and with a weighting and normalization as in Hoberg/Phillips (2016)
   (d) a DTM based on only nouns and adjectives without any weighting/normalization

   The binary weighting as in Hoberg/Phillips (2016) can be done by applying the function `weightBin()` from the `tm` package to a DTM. The second preprocessing step in (a) and (c) is unit length normalization as used by Hoberg/Phillips which is defined as $tf_{adj} = tf/\sqrt{sum(tf^2)}$.

4. For each of the firms outside the Oil sector, you compute the Cosine Similarity between their own term vector and the one representative for the Oil industry. The assumption is, that the higher the computed Cosine Similarity, the closer the firm's business is to the Oil sector. You choose the 25 firms with the highest Cosine Similarity score and invest equally in them for the year 2014.

5. This portfolio is called Oil-tracking portfolio as opposed to the Oil industry portfolio that contains firms classified as being in the Oil sector (`industry.fama.french.49 == 30 Oil`). Compute the return of the Oil industry portfolio - the one you are not allowed to invest in – for each month in 2014. Monthly returns are named `return.monthly.NY.m01-m12` for January to December 2014 in your data. Equally weighted returns are simple averages. Then, compute the return for each of the four Oil-tracking portfolios – the ones you are allowed to invest in – for each month in 2014.

   Report the correlation between the portfolio's monthly return as shown in Table 1 and comment on your results (note that your values probably are different due to your choices when creating the DocumentTermMatrixes). The first cell in the table shows the correlation between the oil industry portfolio and the oil tracking portfolio that was composed, based on the term vectors containing tokens from all word classes and where a binary weighting was applied:

Table 1: Correlations between the oil industry portfolio's monthly returns and the four oil tracking portfolio's return.

	all.wordclasses	nouns.adj
weightBin	0.7453267	0.7996220
no.weight	0.7682501	0.7512502

Shortcut

The main objective for this assignment is to give you an opportunity to practice your problem solving skills. We are aware that going through steps 1–5 requires you to spend some time on the task. While we encourage you to solve the complete task in your group, we want to provide you a shortcut that enables you to skip steps 1. and 2. After downloading the `shortcut` data file from Canvas, you will find 2 lists that contain 2 vectors each in your environment. These vectors are made based on the data frame `section.1.pos`.

(a) vector `text.non.oil.firms$all.wordclasses` that contains all business descriptions of the non-oil firms with tokens from all word classes.
(b) vector `text.non.oil.firms$nouns.adj` that contains all business descriptions of the non-oil firms with only the tokens that are tagged as noun or adjective.
(c) vector `text.oil.firms$all.wordclasses` that contains all business descriptions of the oil firms pasted into one large vector. All word classes are considered.
(d) vector `text.oil.firms$nouns.adj` that contains all business descriptions of the oil firms pasted into one large vector. Only nouns and adjectives are considered.

The names of vector (a) and (b) are the CIKs of the firm. Now you can start with step 3. above.

```
# SAMPLE SOLUTION

load("data_assignment_3.Rdata")

require(dplyr)
# STEP 1: Identify oil firms, filter the data frame "section.1.pos", and construct 2 vectors; one with the
#         tokens from all wordclasses, and one with only nouns and adjectives.

idx.oil.docs <- grep("30 Oil", raw.data$industry.fama.french.49)

section.1.pos %>%
  filter(doc_id %in% idx.oil.docs) -> oil.firms

# Construct a list that will hold the two vectors we are going to construct in the following steps (a) and (b)
text.oil.firms <- list()

# (a) Consider all wordclasses (no filtering) and paste them into one vector. All oil firms will be one document
#     in the Document-Term-Matrix we construct later on.
oil.firms %>%
  pull(token) %>%
  paste(collapse = " ") -> text.oil.firms[[1]]

# (b) Keep only nouns and adjectives (part-of-speech tags start with NN and JJ)
oil.firms %>%
  filter(grepl("NN.*|JJ.*", tag)) %>%
  pull(token) %>%
  paste(collapse = " ") -> text.oil.firms[[2]]

names(text.oil.firms) <- c("all.wordclasses", "nouns.adj")
names(text.oil.firms[[1]]) <- "oil.firms"
names(text.oil.firms[[2]]) <- "oil.firms"
```

```r
# STEP 2: Identify the non-oil firms, and for each of the two variations (a) and (b), paste the tokens of a
#         given wordclass in a document into a single string. This results in a vector of length 481.

# Indexes of the non-oil firms:
section.1.pos %>%
  filter(!doc_id %in% idx.oil.docs) %>%
  pull(doc_id) %>%
  unique() -> idx.non.oil.docs

# Empty list that will hold three vectors
text.non.oil.firms <- list()

# (a) Consider all wordclasses.
section.1.pos %>%
  filter(doc_id %in% idx.non.oil.docs) %>%
  select(doc_id, token) %>%
  group_by(doc_id) %>%
  summarise(text = paste(token, collapse = " ")) %>%
  pull(text) -> text.non.oil.firms[[1]]

# (b) Consider only nouns and adjectives
section.1.pos %>%
  filter(doc_id %in% idx.non.oil.docs) %>%
  filter(grepl("NN.*|JJ.*", tag)) %>%
  select(doc_id, token) %>%
  group_by(doc_id) %>%
  summarise(text = paste(token, collapse = " ")) %>%
  pull(text) -> text.non.oil.firms[[2]]

names(text.non.oil.firms) <- c("all.wordclasses", "nouns.adj")

# We name the elements of "text.non.oil" with the CIKs
cik <- raw.data$cik[idx.non.oil.docs]
names(text.non.oil.firms[[1]]) <- cik
names(text.non.oil.firms[[2]]) <- cik
rm(cik)

# STEP 3: Construct 2 DocumentTermMatrices, one for the corpus with all tokens, and one for the corpus that
#         contains nouns and adjectives. Since we are going to apply two different weighting schemas to the dtms,
#         we will have to construct 4 dtms. Therefor we define a function that makes this step more efficient.
require(tm)
require(slam)

create.dtm <- function(input.vector, weighting = c("bin", "none")){
  VectorSource(input.vector) %>%
    Corpus() %>%
    DocumentTermMatrix(control = list(tolower = T,
                                      removePunctuation = T,
                                      stopwords = T,
                                      removeNumbers = T,
                                      wordLengths = c(3,20),
                                      bounds = list(global = c(5,50)))) -> dtm
  if(weighting == "bin"){
    dtm %>%
      weightBin() -> temp
    temp/sqrt(row_sums(temp^2)) -> dtm
    return(dtm)
  }
  if(weighting == "none"){
    return(dtm)
  }
}

# Create an empty list where we store the DTMs we will create
dtms <- list()

# Combination 1: all wordclasses and binary weighting as in Hoberg (2016)
```

```r
c(text.non.oil.firms$all.wordclasses, text.oil.firms$all.wordclasses) %>%
  create.dtm(weighting ="bin") -> dtms[[1]]

# Combination 2: all wordclasses and no weighting
c(text.non.oil.firms$all.wordclasses, text.oil.firms$all.wordclasses) %>%
  create.dtm(weighting = "none") -> dtms[[2]]

# Combination 3: only nouns/adjectives and binary weighting as in Hoberg (2016)
c(text.non.oil.firms$nouns.adj, text.oil.firms$nouns.adj) %>%
  create.dtm(weighting = "bin") -> dtms[[3]]

# Combination 4: only nouns/adjectives and no weigting
c(text.non.oil.firms$nouns.adj, text.oil.firms$nouns.adj) %>%
  create.dtm(weighting = "none") -> dtms[[4]]


# STEP 4: Compute the cosine similarity between the term vectors of each non-oil firm and the term vector of
#         the oil firms.

apply.cosine.similarity <- function(term.vector.non.oil, term.vector.oil){
  A = term.vector.non.oil
  B = term.vector.oil
  return( sum(A*B)/sqrt(sum(A^2)*sum(B^2)) )
}

# Get the term vectors of the oil firms. There is one row named "oil.firms" in each of the DTMs. This row is
# the term vector of the oil firms.
idx <- c(1:length(dtms))
term.vectors.oil <- lapply(idx, function(x){
  dtms[[x]] %>%
    as.matrix() %>%
    .['oil.firms',]
})

names(term.vectors.oil) <- c("all.wc.bin", "all.wc.no.weighting",
                             "nouns.adj.bin", "nouns.adj.no.weighting")

cos.sim <- list()

# Apply the cosine.similarity function to each row in each DTM in "dtms", sort the result and get the names (cik).
for(i in 1:length(dtms)){
  apply(X = as.matrix(dtms[[i]]),
        MARGIN = 1,
        FUN = apply.cosine.similarity, term.vector.oil = term.vectors.oil[[i]]) %>%
    sort(decreasing = T) %>%
    names() %>%
    .[2:26] -> cos.sim[[i]]
}

# STEP 5: Compute the returns of the four tracking portfolios and calculte the correlation between those and
#         the oil industry portfolio.

monthly.returns <- tibble(month = c(1:12),
                          return.oil = NA,
                          return.oil.tracking = NA)

# Fill in the monthly (mean) returns of the oil portfolio
raw.data[idx.oil.docs, c(28:39)] %>%
  colMeans()-> monthly.returns$return.oil

# We know the CIKs of the 25 firms with the highest cosine similarity

sapply(idx, function(x){
  raw.data %>%
    filter(cik %in% cos.sim[[x]]) %>%
    .[,c(28:39)] %>%
    colMeans() -> monthly.returns$return.oil.tracking
  # Correlation
  cor(monthly.returns$return.oil, monthly.returns$return.oil.tracking)

}) %>%
  matrix(ncol = 2, byrow = F) %>%
  as.data.frame() -> result
row.names(result) <- c("weightBin", "no.weight")
colnames(result) <- c("all.wordclasses", "nouns.adj")
```

```
##           all.wordclasses nouns.adj
## weightBin       0.7453267 0.7996220
## no.weight       0.7682501 0.7512502
```

Bonus: Take a random sample of 25 non-oil firms and calculate the mean correlation of that portfolio and the oil portfolio when repeating the sample taking 50 times.

```
require(dplyr)
cor.values <- numeric()

for(i in 1:50){
 s <- sample(idx.non.oil.docs,25)
 id <- unlist(raw.data[s,1])
 raw.data %>%
   filter(cik %in% id) %>%
   .[,c(28:39)] %>%
   colMeans() -> monthly.returns$return.oil.tracking
 # Correlation
 cor.values[i] <- cor(monthly.returns$return.oil, monthly.returns$return.oil.tracking)
}
 print(paste("Mean correlation value for 25 randomly sampled non-oil firms with 50 repetitions:", mean(cor.values)))
```

```
## [1] "Mean correlation value for 25 randomly sampled non-oil firms with 50 repetitions: 0.283184544267607"
```