Data Analytics for Business

Homework 5 – Part 1

Topic Modeling

In this homework exercise, we will perform probabilistic topic modeling by applying Latent Dirichlet Allocation (LDA) to analyze a collection of news articles on business/finance topics. The news articles were collected from an Asian news website during 2015-2016 and include business/finance news on various markets around the globe. We would like to analyze the texts and extract main topics from the news collection. It would provide us with a useful tool to automatically manage and understand a large volume of business news, which could help detect relevant news on interested topics in the future.

Download the data file "**News.csv**," which contains the following fields. Complete the tasks outlined below.

Data Description

Column Name	Variable Description
doc_id	Document ID (each document corresponds to a news article)
text	Full text content of the news article
date	Date of the news article
heading	Title of the news article

Tasks

1. Install and load the four packages that will be needed: **"tm"**, **"SnowballC"**, "**topicmodels"**, and **"wordcloud"**.

2. Import the data into R. Take the following steps:

   - Import the data from the file "**News.csv**" using **read.csv()** as usual, and save it to a (data frame) variable (e.g., "textdata");

   - Convert the data frame into a corpus using the **DataframeSource()** function in this way: **Corpus**(**DataframeSource**(textdata)). Save the corpus to a variable;

3. Perform the following preprocessing of the text:

   - Remove extra white spaces, using **tm_map**(…, **stripWhitespace**);
   - Remove punctuations, using **tm_map**(…, **removePunctuation**);
   - Remove numbers, using **tm_map**(…, **removeNumbers**);
   - Remove stopwords, using **tm_map**(…, **removeWords, stopwords("english")**);
   - Stem the documents, using **tm_map**(…, **stemDocument**);

   Note: **Keep the original corpus** so that we can retrieve the original full news content later. Save the processed corpus to a different variable.

4. Based on the processed corpus, construct the **document-term matrix** using the **DocumentTermMatrix()** function. Set the **minimum frequency** for terms to be included to be **3**. Display the dimension of the matrix and explain what the numbers of rows and columns mean.

5. Set the seed to 1000, and perform the LDA analysis using the **LDA()** function. Set the **number of topics** to be **20** and the method to be "**Gibbs**." In the "**control**" argument, specify the **number of iterations** to be 1000 and one status print every 50 iterations.

6. Extract the posterior distributions from the LDA result, and do the following:

   - Retrieve the "terms" matrix. What is this matrix about? What is its dimension? What does each row/column represent? Display the **first 5 columns** of this matrix.
   - Retrieve the "topics" matrix. What is this matrix about? What is its dimension? What does each row/column represent? Display the **first 5 rows** of this matrix.

7. Display the 10 most relevant terms for each of the 20 topics, using the **terms()** function. **Pick 3 topics**, think what each of them is about, and come up with a **label/description** for each of them.

8. Pick a news article that interests you the most. (You may go back to the original **News.csv** file to examine the text content.) For example, news article #1082 describes the global financial market reactions right after 2016 U.S. presidential election. Complete the following tasks:

- Display the text content of the news article of your choice. **Note**: you need to retrieve the text from the original (NOT the processed) corpus.
- Create a **bar plot** showing the **topic distribution** associated with this document.
- Pick the topic that is the most relevant to this document. Sort and retrieve the **50 most relevant terms** associated with this topic, and create a **word cloud** of these terms. Think about what this topic is describing and come up with a **label/description** for it.