MiniProj4

Text Mining and Network Analysis of Digital

Libraries in R

4.1 Introduction

we will focus on

* a space of scientific publications related to some science discipline
* and extract patterns
* and text-related information from the dataset.

This chapter will discuss about:

* text preparation techniques in R,
* the use of the latent Dirichlet allocation (LDA) to classify content,
* and the analysis of text features, like co-occurrence matrices.

This chapter will make extensive use of the

* tm,
* sna,
* and the lda R packages.
* Theoretical aspects of the various algorithms will also be discussed,
* as well as R code snippets and relevant charts.

the different steps of our analysis:

1. Dataset preparation:

Using the tm R package, we parse the text content of our dataset.

to get rid of undesirable characters and terms.

2. Exploring the document-term matrix:

* The document-term matrix describes the relationships between documents and terms.
* find associations between terms.
* the term frequency-inverse document frequency measure
* and how it can be used in R.

3. Topical analysis and content clustering using the LDA:

* The LDA is a model where documents are viewed as a mixture of topics.
* learn about a topic representation from the documents of a dataset.
* how to run the learning algorithm,
* validate the model using log-likelihood values,
* describe the different topics resulting from the LDA model,
* and how to visually paint the topic representation of each document.

4. Document cohesion using a similarity measure:

* how one can compute a simple cosine similarity measure on the documents of the dataset,
* and use a heatmap to illustrate the different document clusters.

5. Social Network Analysis of authors:

As a final analysis, we build a network of authors based on coauthorship

interactions between the different participants.

The igraph R package will allow us to

* construct a graph representing our network,
* and, using the sna package, we apply a measure of centrality to the nodes of our graph to describe which authors are more important (in terms of collaboration).

4.2 Dataset Preparation

ArXiv.org is an archive of electronic preprints of scientific publications

**The first dataset to be explored** is the set of all abstracts of papers ,The data are included in the 2011-papers.Rdata file .

* Not founded??

**The *tm* library**

* includes text processing functions.
* tolower, removePunctuation, removeWords, and stripWhitespace
* We use the removeWords function to remove various stop words
* stripWhitespace will collapse repeated whitespaces resulting from the previous text manipulations.
* more transformation functions, such as removeCitation, removeMultipart, removeNumbers, and more.

4.3 Manipulating the Document-Term Matrix

**4.3.1 The Document-Term Matrix**

**The document-term matrix** is simply a matrix describing the frequencies of all terms occurring in the collection of text documents.

dtm

A document-term matrix

* 0 in the matrix is considered a sparse entry
* nonzero value is a nonsparse entry.

we have 1870 documents and 20,176 terms,

yielding 37,729,120 entries in total.

Among these entries, 80,702 entries are nonzero values,

and the rest of the matrix, the other 37,648,148 entries, is zero.

* **Sparsity** is the proportion of sparse entries in the entire matrix.
* The maximal term length is the number of terms in the longest document in our corpus. The default weighting method when instantiating a DocumentTermMatrix is the term-frequency weighting. A detailed discussion on this parameter is given below.

removeSparseTerms(dtm, 0.99)

terms which do not appear in at least 80% of all documents are removed from the original DocumentTermMatrix object

* 4.3.2 Term Frequency-Inverse Document Frequency

**reduce** the size of the matrix by **removing** **sparse** terms without significantly affecting the overall analysis.

4.3.3 Exploring the Document-Term Matrix

findFreqTerms(dtm, 400)

Here, the words with frequency higher than 400, across the whole collection of documents, are shown.

findAssocs(dtm.2, “graph”, 0.3)[1:3]

Given a term, one can also find out which words are highly correlated with that term by using the findAssocs function.

4.4 Clustering Content by Topics Using the LDA

4.4.1 The Latent Dirichlet Allocation

The LDA:

introduces a way to attach topical content to text documents. Each document is viewed as a mix of multiple distinct topics.

4.4.2 Learning the Various Distributions for LDA

* Each word is then associated with a topic,
* and each topic has a term distribution that helps make sense of it.

One way to turn this into a **learning algorithm** is by using the collapsed Gibbs sampling method.

1. Before the first iteration, the algorithm starts by assigning a random topic to each word in each document.
2. Then, at each iteration, the algorithm goes through each word wi,j in document di. For each topic Tk, it computes p(Tk ∣ di), the observed portion of words assigned to topic Tk in document di, and p(wi,j ∣ Tk), the portion of assignment to topic Tk that comes from the word wi,j.
3. The algorithm then resamples a new topic T’i,j for wi,j with probability p(Tk ∣ di)\*p(wi,j ∣ Tk) before it jumps to the next word. An iteration is completed when all words in all documents are revisited.

After a large number of iterations, the model tends to converge to a steady state of topic assignment. We will use the LDA model with R to model topics for our corpus from arXiv.org.

Lexicalize : formate the input to be suitably for the LDA

We are now ready to run the LDA on the resulting lex object:

> res <- lda.collapsed.gibbs.sampler(lex$documents, 10, lex$vocab, 100, 0.1, 0.1, compute.

log.likelihood¼T)

the lda.collapsed.gibbs.sampler function takes the following arguments:

• an object resulting from lexicalize, in our case the lex$documents object

• the number of topics in the model

• the vocabulary associated with the corpus, lex$vocab

• a number of iterations for the Gibbs sampling

• the a parameter describing the term distribution for each topic

• the parameter describing the topic distribution for each document

• a flag indicating whether the log-likelihood should be computed and returned. The log- likelihood is useful when one wants to determine convergence of the LDA process. Details about this return value are discussed below.

The resulting object, res in our case, contains a few important attributes:

• res$assignments: a list of vectors representing the topic association of each term in each document of our corpus. Each entry of res$assignments corresponds to

a document.

• res$document\_sums: a matrix representing the number of times the terms in each document were associated with each of the topics. Rows represent topics and columns are documents.

• res$log.likelihoods: a matrix with two rows. The first row lists the full log-

likelihood values for each iteration, and the second row lists the log-likelihood values of the observations conditioned on the arguments for each iteration.

4.4.3 Using the Log-Likelihood for Model Validation

In our case,

the set of **observations** is the set of existing **documents** and the different **topic** associations for each word occurring in each document.

The parameters are

the **a** and parameters, the multinomial and dirichlet distributions,

and K, the number of topics.

By itself, the log-likelihood value does not have a significant meaning.

But, different models can be compared based on their log-likelihood values. In other words, given the same set of observations, two models having different parameters can be compared based on their likelihood: a model with **greater** **likelihood** is considered more **appropriate** given the set of observations.

4.4.4 Topics Representation

The lda package also includes the top.topic.documents function which returns the top documents for each topic. The output is also in matrix format, just like the top.topic. words function.

4.4.5 Plotting the Topics Associations

Using a combination of the reshape, ggplot2, and RColorBrewer packages, one can create a stacked bar chart illustrating the weight of each topic for all documents in our corpus.

4.5 Using Similarity Between Documents to Explore Document Cohesion

4.5.1 Computing Similarities Between Documents

4.7 Conclusion

From the content of a digital library such as arXiv.org, R has been used with a wide variety of packages to analyze the textual content of the scientific publications, the term occurrences within the different documents, the co-occurrences of terms with other terms, the clustering of papers into different arbitrary topics, and the network analysis of their authors. The LDA algorithm allowed the user to learn a topic assignment model given a set of documents and words. A few plots and charts were plotted using the ggplot2 package to visually enhance the analytical content of the different techniques used in the current chapter.