CW2: Understanding Data

Manuel Llamas
University of Southampton
Id: 28191943
mlf1g15@soton.ac.uk

ABSTRACT
In this paper I am going to study and apply different data mining techniques for document analysis with the aim of classify and organize them by finding relationship, in most of the cases this relationship will be obvious while in some others I will need further information to check the performance of the algorithms.

Keywords
Zipf's law, tf-idf, cosine similarity, hierarchical clustering, k-means, multidimensional scaling.

1. INTRODUCTION
There is many different techniques in text mining and document classification, which results can vary a lot. From the criteria followed to clean and pre-process the data, the weights chosen, to the classification algorithm used, the results can differ a lot.

The software chosen for all the pre-processing and analysis carried out in this work has been R-Studio, which contains a big collection of packages built in for this precise purpose (mainly tm [1] but also ls or SnowballC are some examples).

2. PREPROCESSING
All these algorithms that analyses and classifies documents need the data in a readable format as they work with counts of words. Given this all the pre-processing stage is focused on transforming the documents in word occurrence (very large) vectors.

The raw html files we download are scanned pages from original books using an OCR system. However these files contain too much information for our purpose, as there is not only the text, but other features like the format of the words wrote in the html code.

The first step on the pre-processing stage is to extract the words or tokenise the text. It is possible to search the words in the documents using regular expressions. It was built a double for loop that queries page by page for all the books returning a nested list with all the terms for each one of the books: the corpus.

The following step is to clean the corpus by converting to lower case all the letters (the system reads the capital letter as different as lower case, so could read the very same word as different), removing numbers, stop words (I will speak briefly about Zipf's law later on), strange symbols and white spaces. Also we have to stem the words, this is removing mainly endings of words like –ing or –s to reduce them to their root [2].

The third stage is to create a bag of words by building a document-term matrix (dtm) where we have the count of all the words for each book, it is obvious that this matrix will be very sparse with many of the terms appearing few times.

But before starting with the exploratory analysis we need to have a look to Zipf's law: there is going to be many rare words (I wrote about the sparse dtm) and then they are going to be not only useless but even harmful to the analysis since they are going to have an important weight "isolating" the texts from the others. This summed to the fact that the more terms the more inefficient the algorithms are reasons enough to delete the very sparse terms by leaving a matrix with around 5% of sparsity (after some trials this percentage gave the best results).

The same law tells us in terms of very frequent words (like the mentioned above stop words) that could obscure the classification just due to random factors that make them appear more in some texts without having any semantic value, that is why it makes sense to remove them.

3. EXPLORATORY ANALYSIS
The first exploratory analysis we should carry on is having a look to the most frequent words of the dtm to "summarize" the topics of the books and try to find some first relationships.

Here we can observe that the most common word, *citi-*, is very frequent in every text and give us no insight but the in second one, *roman*, we find it will be very important when classifying since we can observe two cases: it is either very frequent (Roman History texts) or it hardly appears (Greek texts).

Despite this specific case, the most common words, in general, are 'equally' frequent in every text (*time, name, war, peopl-*…) and they are not going to add any value to our classification (we cannot extend Zipf's law here as we could remove meaningful words like *roman*).

We can solve this problem by weighting the terms using tf-idf instead of just term frequency. This means increasing the weight proportionally to the number of times a word appears in the document, tf, but offsetting it by the frequency of the word in the corpus, idf [3]: words that appears very frequently in every document have smaller weights.

3.1 Hierarchical Clustering
There is different ways to perform a hierarchical clustering, *fig. 1* shows the resulting dendrogram after applying an agglomerative clustering (bottom up, starting by treating each book as a cluster and linking them sequentially) that clusters at each step the pair of documents that are closer (average method). It was chosen an agglomerative method as divisive ones (e.g. DIANA) are less efficient [4]. However this algorithm usually has efficiency problems, but in this case, and thanks to removing lots of very sparse terms in advance, I am not facing this problem.

The method that worked the best was a combination of average method and cosine similarity (instead of Euclidean distance), most commonly used in high-dimensional spaces like this. Each term is assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document (dtm). Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter [5], what makes all the sense to apply to text mining.

The results show that it seems to properly fit our data, we can see how all the Gibbon's books (Decline and Fall Rome) are clustered together and close to Tacitus (Roman Empire History) but not as

much to Livius books which subject is ancient Rome (Roman Republic) while the books of war topics are in the opposite side, despite the specific war treated. Another reflection we can make is that we find the Greek topics in the opposite side (Description, Dictionary and Peloponnesian war), showing the weaker relationship with Roman History.

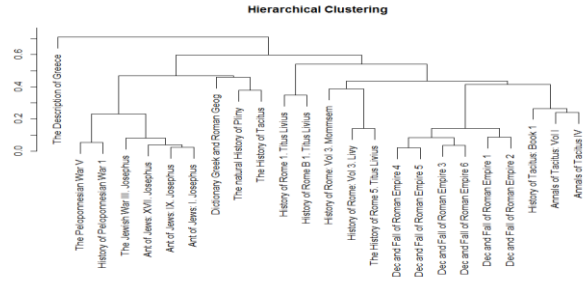

Figure 1

3.2 K-Means Clustering

K-means clustering aims to divide all of our observations in k clusters predefined by us [4]. This limits the method since we first need to have some prior understanding about the data. In addition k-means works by randomly throwing the centres of the clusters and iteratively assigning the nearest observation to them and updating the centres until no more changes are possible, introducing a random factor that makes it not fully reliable. However this is a powerful method when combined with some other exploratory analysis like the hierarchical clustering.

In *fig. 2* we can see one classification performed by a k-means algorithm where k=6, chosen after analysing the dendrogram and carrying out some trials.

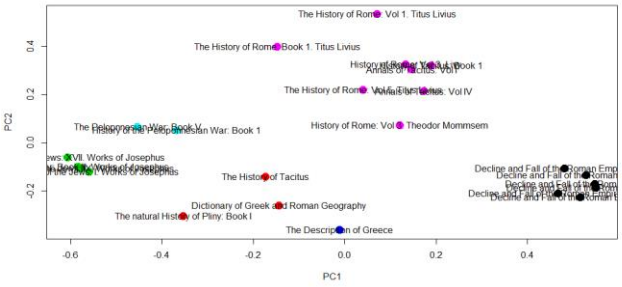

Figure 2

Here the result has some differences from the dendrogram: Works of Josephus (Jewish War) and Peloponnesian War are two clearly different clusters (different wars) and the Geography Dictionary is closer to Description of Greece (what makes sense as both have Greece as common topic. However there is important to remember that due to the random nature of this algorithm there was different results in each run of the algorithm.

3.3 Multidimensional Scaling

Since our data has many dimensions (terms) it may be useful trying to project them in a lower dimensional space to study if we can get different results. MDS is one method that performs precisely this: Using the Euclidean distance between the observations the algorithm does the projections by maintaining this distances in a 2-D space [6]. We can see its results in *fig. 3.*

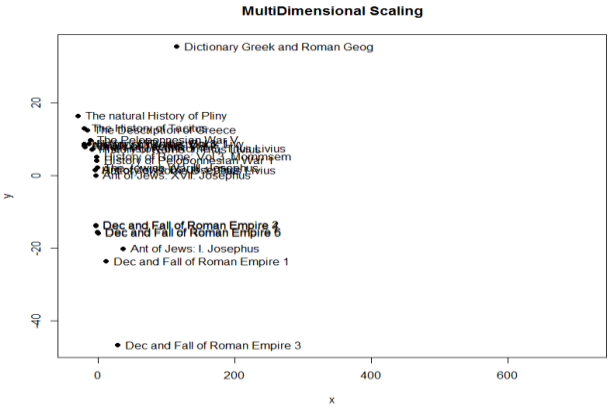

Figure 3

4. RESULTS AND DISCUSSION

I got different results for the different methods used. And after retrieving information about these authors and works (interval of time and specific topics treated in the books) I conclude that, although the k-means seems better, the hierarchical clustering outperforms the other methods since it uses the cosine similarity, what makes more sense when studying the topic of a text. However there is some outliers like the Tacitus History book classified near the Dictionary and the Natural History books. This could be because this Tacitus' book is a preface or an index of all his works what would make it more similar to those other two.

I am not confident with k-means due to its random nature, but despite this fact all the results it gave are always close to the hierarchical clustering due to the fact that it uses Euclidean distance but with the document vectors normalised, this makes it proportional to the cosine of the angle between vectors:

$$\|x\|_2 = \|y\|_2 = 1$$

$$\|x - y\|_2^2 = (x-y)^T(x-y) = x^Tx - 2x^Ty + y^Ty = 2 - 2x^Ty$$
$$= 2 - 2\cos\angle(x,y)$$

Finally about the MDS , it is very confusing and it doesn't show the relationships we expect from these texts, this is due to the fact that it automatically computes the Euclidean distance (without normalising) which results in a poor classification using texts.

5. REFERENCES

[1] Feinerer. An introduction to text mining in R. R News, 8(2):19–22, Oct. 2008. DOI= http://CRAN.R-project.org/doc/Rnews/

[2] Hare J. Searching and Ranking. University of Southampton, Data Mining COMP6237.

[3] Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". *Journal of Documentation* 28: 11–21

[4] Rokach, Lior, and Oded Maimon. "Clustering methods." Data mining and knowledge discovery handbook. Springer US, 2005. 321-352

[5] Singhal, Amit (2001). "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43

[6] Borg, I., Groenen, P. (2005). *Modern Multidimensional Scaling: theory and applications* (2nd ed.). New York: Springer-Verlag. pp. 207–212.