SAPIENZA
UNIVERSITÀ DI ROMA

Document clustering techniques: comparative analysis and development of a novel approach

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

Corso di Laurea Magistrale in Statistical methods and applications

Candidate

Gian Mario Sangiovanni

ID number 1889445

Thesis Advisor

Prof. Maria Brigida Ferraro

Co-Advisor

Prof Paolo Giordani

Academic Year 2023/2024

Thesis defended on 25 July 2024
in front of a Board of Examiners composed by:

Prof. Luca Tardella (chairman)
Prof. Maria Brigida Ferraro
Prof. Agostino Di Ciaccio
Prof. Giovanna Jona Lasinio
Prof. Filomena Maggino
Prof. Valentina Minnetti
Prof. Maurizio Vichi
Prof. Roberto Zelli

Document clustering techniques: comparative analysis and development of a novel approach
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: sangiovanni.1889445@studenti.uniroma1.it

Abstract

In today's era of abundant textual data on the web, automated clustering into thematic folders is essential. This thesis extensively explores document clustering techniques, elucidating their operational frameworks and their main advantages and disadvantages. Subsequently, three novel algorithms are introduced and compared with the already existing ones. Initially, foundational tools such as Latent Dirichlet Allocation (LDA) and the Bootstrap method are presented. Then, the new clustering algorithms are theoretically delineated. The innovation lies in proposing a novel "distance" measure based on the p-value of an hypothesis test of the homogeneity of topic distributions, obtained by LDA, between two documents, mounting on it three different clustering procedures. The first two directly employs the new dissimilarity using an hierarchical approach and a fuzzy relational clustering approach while the other is a test-based approach to clustering, an unprecedented endeavor. The performance of the clustering methods is then assessed using two benchmark datasets with comparisons drawn against established methodologies. Finally, agglomerative hierarchical clustering is applied to analyze thematic changes over three different years at the CMStatistics conference, providing insightful conclusions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In a world where the number of available web documents is growing at an exponential rate each day, it is crucial to introduce methods that are effective for clustering documents into defined "folders". For instance, consider a scenario in which the objective is to identify all the documents in a database associated with a specific query (which could be considered as a topic search).

In Chapter 2 the principal methodological approaches in document clustering are briefly presented, as well as the main clustering techniques used, highlighting the challenges of clustering in a high dimensional regime with sparse data. Then, the main similarity measures in the document clustering context and the most used cluster validity techniques (both internal and external measures) are explored. The potential need for a new validity index, more specifically made for document clustering, is also highlighted.

In Chapter 3 the basic idea of the three clustering methods is outlined and the main tools used to build them are introduced, including the Bootstrap method and LDA. Following this, the hypothesis test for homogeneity between topic distributions and the procedure to estimate the p-value is introduced, outlining advantages and disadvantages of this approach. Finally, the three novel clustering algorithms are critically presented.

A small-scale application of all the clustering procedures is then conducted on one benchmark dataset (20 Newsgroups dataset) and on a new one, namely the

Books of Abstracts (BoAs) dataset. This application includes a comparison of the three newly defined clustering methods with well-known clustering algorithms from the literature. The performance of these methods is evaluated using both cluster validity indices and interpretative analysis. The main drawbacks and advantages of the new methodologies, as well as potential extensions, are discussed.

In Chapter 4, document clustering is applied to a real case scenario using the BoAs dataset. Specifically, agglomerative hierarchical clustering is employed for three different years to identify the main topics of interest, determine if there are discernible trends over time, and assess which documents belong to specific clusters. The primary goal is to create "folders" for each year that group documents by their associated topics.

Aim of the thesis

Before defining the main purposes of this thesis, it is essential to clearly identify the primary research questions and develop a framework to address them:

- **Which are the main document clustering techniques? Which are their main limitations?**
- **Is it possible to define a new document clustering scheme that provides both a methodological and interpretative perspective?**

In order to give an answer to the involved questions, firstly the main techniques in the literature have to be studied into detail in order to understand advantages and disadvantages. Then, three new methods of document clustering based on the estimation of the p-value associated with the hypothesis test of homogeneity between document topic distributions [35] have been discussed.

It should be noted that this is a first attempt to expand the methodological boundaries of document clustering and this is also an initial idea. Specifically, it is noteworthy that this is the first instance where LDA is not directly employed for clustering, as is traditionally done in Mixture Models where each topic is treated as a cluster.

A gap between classic techniques based on the Bag-Of-Words (BOW) approach and those employing embeddings directly have been identified. The attempt to define a methodology that does not rely on language models but remains somehow effective has been done.

Chapter 2

Theoretical Framework

2.1 Literature Review

Document clustering stands out as a prominent subject in the realm of Information Retrieval (IR), with seminal works by [58, 36] striving to leverage its application to enhance the precision of retrieval systems. The basic idea is to extract information from a text, using concepts of Data Mining and then to create clusters, namely set of documents that share some common characteristics according to some similarity measure that was previously defined, using concepts of Natural Language Processing and methodological statistics. Recall that clustering is an unsupervised learning type of method, meaning that document clustering is significantly different from document classification due to the fact that usually the labels associated to each document are unknown a priori. In that field, a wide range of machine learning techniques have been applied by [33].

The goal of document clustering is to create a partition of documents such that the intra-cluster similarity is maximized while the inter-cluster similarity is minimized. This methodology finds extensive applicability across diverse domains, ranging from large-scale web searching [16] to hierarchical models classifying web services such as YAHOO using concise descriptors [34]. Furthermore, it plays a pivotal role in the automatic organization of vast corpora [3]. Notably, the significant applications of document clustering can be categorized as follows [30]:

- **Similarity**: this feature proves invaluable in uncovering documents that share not just words but also similar conceptual content. For instance, it is instrumental in conducting a thorough literature review for a thesis, where understanding the prevalence of specific topics is crucial.

- **Organization**: in the current era, grappling with the enormous volume of data necessitates effective organization. Document clustering addresses this challenge by categorizing large document collections into interpretable structures, facilitating human comprehension, such as the categorization of books based on literary genres.

- **Duplication**: identifying duplicate documents within an extensive collection is essential. In the context of plagiarism detection in academic works, this capability is instrumental in gauging the extent of textual overlap in theses or research papers.

- **Recommendation**: drawing upon the user's reading history, the objective is to suggest content with similar themes, minimizing time wastage and enhancing user experience.

- **Web Search Optimization**: clustering significantly contributes to the improvement of search engine quality and efficiency. By initially comparing user queries to clusters rather than directly to individual documents, search results can be more efficiently organized and retrieved.

From a methodological standpoint, various clustering techniques have been employed to address the challenges outlined previously. According to [12], unsupervised document clustering methods can be broadly categorized into two groups, i.e. hard clustering and soft (fuzzy) clustering. In hard clustering, each document is directly assigned to one and only one cluster. This results in a set of disjoint clusters, where documents are exclusively associated with a single cluster. Conversely, in the other situation, a document can be assigned to multiple clusters. This approach generates a set of overlapping clusters, where documents may belong to more than one cluster with a certain membership degree.

Within the clustering methodology, three distinct approaches are commonly found. **Partitioning algorithms**, such as K-Means, attempt to allocate documents into a fixed number of K clusters. **Hierarchical algorithms**, such as Bisecting K-Means [38], aim to create a nested partition of clusters, where each partition of K clusters is deemed the best given the $K-1$ cluster partition. The last one involves **frequent itemset-based clustering**, where terms are clustered based on their frequency and documents are then clustered based on those terms [5].

Another typical approach frequently used is **Model-Based clustering** [67]. The fundamental idea behind this method is that the population of documents is not homogeneous; instead, there are distinct sub-populations that govern the data generative process. The key challenge lies in identifying which mixture model suits the text data best and determining the appropriate number of mixture components. Subsequently, an Expectation-Maximization [19] algorithm can be employed to estimate the parameters. From a practical point of view, one approach proposes to apply a dimensionality reduction technique before implementing the cluster modeling [49], while the authors in [65] use a Dirichlet-Multinomial Mixture model to cluster short texts. Indeed, the most widely utilized technique in this category is the Gaussian Mixture Model (GMM) [24].

An explored approach involves the use of **Density-Based methods**. In particular, it has been already explored a general application to document clustering [15] as well as a more specific one on Twitter data [28]. Another approach involves constructing an undirected weighted graph from the initial corpus and applying spectral clustering techniques to it [21]. Specifically, the concept of graph partitioning relies on computing the eigenvectors associated with the graph matrix and subsequently applying a clustering algorithm within this new subspace.

It is noteworthy to mention that, given the reliance of the algorithm which will be developed during this thesis on LDA, various papers have explored the use of clustering methods following LDA. For example, the usage of an ant algorithm by [52], the application of clustering methods based on LDA to scientific documents [64] or the employment of a combination of LDA and Word2Vec to cluster documents

with a small text such as abstracts of papers [40]. Finally, the application of a hierarchical approach based on a matrix of p-values derived from a two-sample test to classify ARMA models has been explored by [43].

2.2 Useful tools

After delineating the application domain of document clustering as well as the basic techniques used, it becomes crucial to intricately present the theoretical tools that will be important in this thesis. It is imperative to provide a concise overview of the techniques to be employed. For a more in-depth analysis, readers are encouraged to refer to the linked papers. While certain topics may initially appear unrelated, it is in the subsequent Section/Chapter that these elements will be seamlessly integrated into a unified methodology. As a matter of fact, document clustering is a mix of different steps and methods and it is noteworthy to mention all of them.

2.2.1 Vector Space Model

Let us introduce some notations that will prove instrumental for subsequent discussions. A corpus $\mathcal{C} = \{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$ can be defined as a collection of documents, where M is the total number of documents. Subsequently, each individual document $\mathbf{d}_i = \{w_1^i, \ldots, w_{n_{\mathbf{d}_i}}^i\}$ is defined as an ensemble of words, with $n_{\mathbf{d}_i}$ denoting the total number of words within the document $\mathbf{d}_i$. This representation aligns with the BOW paradigm, wherein the sequence and semantic relationships between words are disregarded. This assumption represents the most basic and rudimentary conceptualization, as it oversimplifies the intricate nature of language. Notably, more recent language models, such as the one developed by OpenAi [53], have transcended this elementary framework, adopting more sophisticated and nuanced approaches that capture the intricacies of semantic relationships and contextual dependencies among words. Lastly, denote $V = \{w_1, \ldots, w_{|V|}\}$ as the vocabulary associated with the corpus, constituting the set of all the unique words in the corpus.

The Vector Space Model (VSM), conceptualized in the 1960s but formally articulated in a book in 1983 [59], underpins the representation of each word $\mathbf{w}_l$

($l = 1, \ldots, |V|$) as a *one hot encoded* vector. Through this specific encoding, $\mathbf{w}_l$ is transformed into a $|V|$-dimensional vector, wherein w_{lj} equals one exclusively when it corresponds to the word in position j inside the vocabulary, otherwise it assumes a value of zero. Consequently, each document $\mathbf{d}_i$ can be depicted as a matrix, where each row constitutes a one hot encoded vector representing a specific word in the vocabulary, denoted as $\mathbf{d}_i = [\mathbf{w}_1^i \ldots \mathbf{w}_{n_{\mathbf{d}_i}}^i] \in |V| \times n_{\mathbf{d}_i}$.

A more prevalent representation of a document is the vectorial one, where each document is encapsulated as a $|V|$-dimensional vector. In the broader context of VSM, a corpus comprising M documents with $|V|$ unique terms undergoes transformation into an $M \times |V|$ matrix. Each row within this matrix corresponds to one of the aforementioned vectors and is commonly referred to as the **Document Term matrix** (DTM). Each entry in this matrix represents the weight of the associated word within the specific document. Notably, at least three distinct weighting schemes can be defined, though numerous others exist in literature [1] [63]:

1. **Binary representation**: each document is encoded such that the generic entry dt_{ij} is set to one if the word in position j in the vocabulary is utilized at least once in the document $\mathbf{d}_i$, otherwise it is set to zero. This encoding implies the presence of at least one occurrence of the word within the document, manifesting as a binary indicator in the corresponding row of the document's matrix representation.

2. **Term frequency**: the generic entry of a document dt_{ij} is equal to the number of occurrences of the word in position j in the vocabulary within the document $\mathbf{d}_i$. Essentially, it corresponds to the frequency of the word in the document.

3. **Term frequency inverse document frequency**: the TF-IDF representation serves as a refinement of the term frequency scheme by incorporating considerations regarding the overall corpus. This approach acknowledges the significance of a word based on both its frequency within a document and its prevalence across the entire corpus. In fact, the more a word is used within

[1] The last two methods are within the count BOW (CBOW) approach.

the corpus, the lower it is the amount of information "owned" by that word. More into detail:

$$dt_{ij} = tf_{ij} * log\left(\frac{M}{M_j}\right)$$

where M_j represents the number of documents including the word in position j in the vocabulary. An easy extension of the method aforementioned involves applying a convex transformation to decrease the emphasis placed on the frequency of each word. Specifically, the term frequency is replaced by $\log\left(\mathrm{tf}_{ij} + 1\right)$, where the plus one is included to prevent the transformation from resulting in a value of negative infinity.

2.2.2 Text Preprocessing

In the BOW context, before applying whatever algorithm, the crucial step is document preprocessing [20]. The primary objective of these techniques is to eliminate irrelevant information to facilitate further analysis. The choice of preprocessing techniques depends on the specific domain of application. For example, in legal contexts, numbers may be essential for uniquely identifying laws, whereas in analyzing political speeches in order to understand the political orientation, numbers may not provide valuable insights and can be omitted. The key preprocessing techniques commonly used include:

1) **Punctuation and Special Characters**: the primary step in any preprocessing procedure is to remove punctuation from text and identify which special characters ("!&%()...") should be eliminated. For example, in Twitter data, the symbol "#" is directly associated with hashtags and may hold valuable information for analysis. Special characters also include URLs or emails. Taking Twitter data as an example again, it is essential to exclude mentions, which are words starting with the symbol "@". The rationale behind removing symbols and punctuation is that words are analyzed individually, making these elements irrelevant for analysis.

2) **Lowercasing**: the next common step in preprocessing is to convert all words to lowercase since it does not affect the meaning of a word. For example, "War"

and "war" carry the same significance, and there is no need to include the same word multiple times in the dictionary.

3) **Tokenization and Stopwords**: the tokenization process involves splitting each word in a text using the white space between words as a delimiter. Subsequently, stopwords, which are non-significant words in a given language, are removed from the documents. For most languages, there are one or more lists of stopwords available (for instance in Italian you can check this list).

4) **N-gram**: following the concepts outlined in [45], an n-gram is defined as a sequence of n contiguous tokens. It is crucial to include these expressions in the vocabulary as they carry significant meaning. Typically, emphasis is placed on bigrams and trigrams, with a focus on adding the most frequent ones. For example, terms like "political" and "compass" may seem unrelated individually, but when considered together, they convey a specific interpretation.

5) **Lemmatization & Stemming**: the decision regarding the utilization of either stemming or lemmatization techniques is discretionary and you should opt for one over the other based on task specific requirements. Stemming conventionally denotes the process of transforming a token into its root form. Among the most widely employed algorithms for stemming is the Porter algorithm [56]. For Italian stemming, the Snowball algorithm serves as a viable alternative [57]. Conversely, lemmatization entails the transformation of a word into its lemma, a process intricately linked with part-of-speech tagging. Generally, it is advisable to start with stemming techniques. Only if the performance of the model prove unsatisfactory, consideration of lemmatization becomes pertinent. This recommendation is based on lemmatization's reliance on an external vocabulary and its dependence on a model for the assignment of part-of-speech tags to individual words.

6) **Frequent words**: another step that can be taken involves deleting very frequent or rare words. There is no specific rule for this process. Some approaches involve defining a threshold to remove frequent and rare words, while others use the frequency of documents containing that word to determine

which words to delete, typically removing the ones appearing in more than $85-95\%$ or less than $0.5-1\%$ of documents.

2.2.3 Similarity Measures

In the context of document clustering, selecting an appropriate similarity measure is a crucial step. This measure enables the determination of similarity or dissimilarity between documents within a corpus. To compute such similarities, it is imperative to represent documents as vectors, as for example the TF-IDF representation where each word in the vocabulary corresponds to a feature. The chosen distance in order to be a metric should adhere to certain properties. More formally, a distance function, denoted as $D : \Re^{|V|} \times \Re^{|V|} \to \Re^{+}$, is considered a metric if it satisfies the following criteria when applied to any two documents d_i and d_j within the corpus::

1. non-negativity, $D(\mathbf{d}_i, \mathbf{d}_j) \geq 0$;
2. symmetry, $D(\mathbf{d}_i, \mathbf{d}_j) = D(\mathbf{d}_j, \mathbf{d}_i)$;
3. reflexivity, $D(\mathbf{d}_i, \mathbf{d}_i) = 0$
4. triangle inequality, $D(\mathbf{d}_i, \mathbf{d}_l) \leq D(\mathbf{d}_i, \mathbf{d}_j) + D(\mathbf{d}_j, \mathbf{d}_l)$

It's important to note that a similarity measure assesses the degree of similarity between two documents and can be viewed as the opposite of a distance function. Moreover, not all similarity measures possess the properties necessary to be classified as a metric. Even though there exists different measures [27], the most pertinent ones are discussed below:

Cosine Similarity: widely utilized in information retrieval due to its independence from document length, cosine similarity measures the angle between two vectors and is bounded within the range of $[0, 1]$, where 1 indicates nearly coinciding vectors, while 0 indicates perpendicular vectors. It is not a metric because it does not satisfy the triangle inequality.

$$\cos(\mathbf{d}_i, \mathbf{d}_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{|\mathbf{d}_i||\mathbf{d}_j|} \tag{2.1}$$

Jaccard (Tanimoto) coefficient: it measures the similarity between two elements using the ratio between the cardinality of the intersection set and the cardinality of the union set. It ranges between 0 and 1 and it is a similarity measure

$$JC(\mathbf{d}_i, \mathbf{d}_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{|\mathbf{d}_i|^2 |\mathbf{d}_j|^2 - \langle \mathbf{d}_i, \mathbf{d}_j \rangle} \tag{2.2}$$

Euclidean Distance: a standard metric extensively used in clustering problems, satisfying all four metric requirements. Sometimes, the square version is adopted as well as the norm in L_p, although the first one is not a metric

$$D_E(\mathbf{d}_i, \mathbf{d}_j) = |\mathbf{d}_i - \mathbf{d}_j|_2 = \left(\sum_{r=1}^{|V|} (d_{ir} - d_{jr})^2 \right)^{\frac{1}{2}} \tag{2.3}$$

Kullback-Leibler divergence [37]: this measure relies on the assumption that each document is a probability distribution over a space (for example over the vocabulary space), quantifying the degree of dissimilarity between two probability distributions. Although not a metric due to its lack of symmetry and triangle inequality fulfillment, it can be interpreted as the difference between cross-entropy and document entropy, bounded in $\Re^+$

$$\begin{aligned} KL(\mathbf{d}_i, \mathbf{d}_j) = \mathbb{E}\left[\log\left(\frac{\mathbf{d}_i}{\mathbf{d}_j} \right) \right] = \sum_{r=1}^{|V|} \left(d_{ir} \log\left(\frac{d_{ir}}{d_{jr}} \right) \right) = \\ = \sum_{r=1}^{|V|} \left(d_{ir} \log\left(d_{ir} \right) \right) - \sum_{r=1}^{|V|} \left(d_{ir} \log\left(d_{jr} \right) \right) = H(\mathbf{d}_i, \mathbf{d}_j) - H(\mathbf{d}_i) \end{aligned} \tag{2.4}$$

Bhattacharyya distance [6]: employed for quantifying divergence between probability distributions, Bhattacharyya distance is also not a metric due to its failure to satisfy the triangle inequality. It is bounded in $\Re^+$ and is zero when the distributions are identical. It is related to Kullback-Leibler divergence through the Hellinger distance

$$B(\mathbf{d}_i, \mathbf{d}_j) = -\log\left(BC(\mathbf{d}_i, \mathbf{d}_j) \right) = -\log\left(\sum_{r=1}^{|V|} (d_{ir} * d_{jr})^{\frac{1}{2}} \right) \tag{2.5}$$

2.3 Clustering Techniques

Let's formalize the task. Given a corpus $\mathcal{C} = \{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$, the goal of this unsupervised method is to partition these documents into clusters in the most optimal manner. The aim is to ensure that documents within the same cluster exhibit high similarity (cohesion), while those in different clusters show lower similarity (separation).

Essentially, the starting point is the representation of documents using VSM. Ideally, each document clustering technique should begin with a DTM, where words in the vocabulary act as variables (features), with their frequencies serving as scores. In this matrix, each document has been normalized to have a unit length, which means that each document vector has been divided by its norm in order to take into account the different sizes of documents [2]. Formally, for the DTM:

$$\begin{aligned} T &= \sum_{i=1}^{M}\sum_{j=1}^{|V|}\left(dt_{ij} - \overline{dt}_{\cdot j}\right)^2 = \sum_{k=1}^{K}\sum_{i=1}^{M_k}\sum_{j=1}^{|V|}(\overline{dt}_{kj} - \overline{dt}_{\cdot j})^2 + \sum_{k=1}^{K}\sum_{i=1}^{M_k}\sum_{j=1}^{|V|}\left(dt_{ij} - \overline{dt}_{kj}\right)^2 = \\ &= \sum_{k=1}^{K}\sum_{j=1}^{|V|} M_k(\overline{dt}_{kj} - \overline{dt}_{\cdot j})^2 + \sum_{k=1}^{K}\sum_{i=1}^{M_k}\sum_{j=1}^{|V|}\left(dt_{ij} - \overline{dt}_{kj}\right)^2 = B + W \end{aligned} \tag{2.6}$$

where $\overline{dt}_{kj}$ is the j-th component of the centroid of the cluster k, namely the mean of the documents inside the cluster. Assuming that the generic cluster k has size n_k, the centroid $\overline{dt}_{k\cdot} = \sum_{i=1}^{M_k} dt_{i\cdot}/n_k$.

Based on the above discussion, the objective is to maximize the variance between clusters or, conversely, minimize the variance within clusters. It's important to note that in this context, squared Euclidean distance is generally not considered a suitable metric; instead, cosine similarity is preferred. To delve further, the distance between two documents can be defined as follows:

$$D(\mathbf{d}_i, \mathbf{d}_j) = 1 - \cos\left(\mathbf{d}_i, \mathbf{d}_j\right)$$

[2] As a matter of fact, in the upcoming chapters, clustering techniques will be applied to various representations of documents beyond just their frequency. In these cases, even though normalization is not necessary, the same reasoning can be applied.

Three important points can be made. Firstly, alternative dissimilarity measures can also be used. Secondly, the defined distance remains valid for all other methods as well. The other one is that intrinsically document clustering can be considered as an instance of clustering with high dimensional data because the DTM has usually many features, often exceeding the number of documents. In particular, there are several problems linked with high dimensional clustering:

- **Curse of dimensionality** [55]: due to the high dimensionality of the feature space, distance measures becomes meaningless making it difficult to distinguish between distant and nearby points;
- **Hard conceptualization**: visualizing and interpreting units in a high-dimensional space is challenging.

Thus, before applying clustering techniques which does not rely directly on a dimensionality reduction technique, is a good idea to apply a transformation to the DTM in order to reduce the sparsity. Although various transformations exist in the literature (both linear and nonlinear), the two most used in this context are Principal Component Analysis (PCA) and Singular Values Decomposition (SVD).

PCA reduces the feature space applying a linear transformation based on the eigenvalues and eigenvectors associated to the variance and covariance matrix of the original data. Without going too much into detail, the simplistic idea behind this method is to find components, defined as a linear combination of the original features, such that each component (which is indeed defined by the eigenvector) is orthogonal with respect to the others, the variability (and thus the information) is maximized and the components are chosen in a sequential way. Furthermore, the variance of each component is equal to the eigenvalue corresponding to that eigenvector.

On the other hand, SVD factorizes the original rectangular DTM into three different components in the following way:

$$\mathbf{DTM} = \mathbf{UDV}^T$$

where $\mathbf{U}$ is a $M \times M$ matrix containing the left singular vectors of the DTM, $\mathbf{D}$ is a $M \times |V|$ diagonal matrix containing the corresponding singular value in each cell of the diagonal while V is a $|V| \times |V|$ matrix containing the right singular vectors. The number of non zero singular value is equal to the rank of the DTM (which is M in this case) and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.

2.3.1 Hierarchical methods

The purpose of this technique is to create nested partitions, where, given two levels of dissimilarity $\delta_1 < \delta_2$, the partition obtained with δ_1 is contained in the partition obtained with δ_2. Two different approaches can be employed depending on the starting point. If the starting point is the trivial partition containing all the different documents and a split of the previous cluster into two new clusters is done at each step, a divisive method is applied. In the other case, the method is called agglomerative. The main focus will be on the latter.

Agglomerative methods tend to produce a continuous sequence of partitions where similar documents or clusters of documents are merged together at each step. Specifically, given an $M \times M$ dissimilarity matrix, the related algorithm can be defined as follows:

Step 1. merge the two documents exhibiting the lowest dissimilarity into a cluster. Compute the dissimilarities between the newly obtained clusters and the remaining documents, defining an $(M-1) \times (M-1)$ matrix, where the dissimilarities between singletons are taken from the previous matrix;

Step l. for l ranging from 2 to $M-1$, merge the two clusters with the lowest dissimilarity and reconstruct the new matrix of shape $(M-l) \times (M-l)$. The final outcome is the trivial partition where all documents are within the same cluster.

A natural question arises: how is the dissimilarity between two clusters computed? While various linkage methods exist, the discussion here is limited to the most interesting ones:

- **Single Linkage**: the dissimilarity between two clusters k_1 and k_2 is defined as the minimum dissimilarity between any element of the two clusters, often leading to narrow shape clusters

$$d_{k_1 k_2} = \min D(\mathbf{d}_i, \mathbf{d}_j) \quad i \in k_1 \ , \ j \in k_2$$

- **Complete Linkage**: this method computes the dissimilarity between two clusters k_1 and k_2 as the maximum dissimilarity between any elements of the respective clusters, typically creating spherical clusters

$$d_{k_1 k_2} = \max D(\mathbf{d}_i, \mathbf{d}_j) \quad i \in k_1 \ , \ j \in k_2$$

- **Unweighted Pair Group Method with Arithmetic Mean (UPGMA)**:[29] here, the dissimilarity between two clusters k_1 and k_2 is equal to the average dissimilarity between every pair of elements in k_1 and k_2.

$$d_{k_1 k_2} = \frac{1}{M_{k_1} M_{k_2}} \sum_{i \in k_1} \sum_{j \in k_2} D(\mathbf{d}_i, \mathbf{d}_j)$$

- **Weighted Pair Group Method with Arithmetic Mean (WPGMA)**: it is the weighted version of the previous method, which does not directly take into account the size of the merged clusters. In fact, assuming that the cluster k and k^* have been merged together in the previous step into a new cluster G_1, the dissimilarity between clusters k_1 and k_2 is the arithmetic mean of the average dissimilarity between k and k_2 and between k^* and k_2.

$$d_{k_1 k_2} = \frac{1}{2} \left(\frac{1}{M_k M_{k_2}} \sum_{i \in k} \sum_{j \in k_2} D(\mathbf{d}_i, \mathbf{d}_j) + \frac{1}{M_{k^*} M_{k_2}} \sum_{i \in k^*} \sum_{j \in k_2} D(\mathbf{d}_i, \mathbf{d}_j) \right)$$

Hierarchical methods are often represented graphically via dendrograms, which serve as a mapping function correlating a given level of similarity/dissimilarity to the corresponding partition. The dendrogram is represented as a tree, where clusters constitute the nodes and the dissimilarity values between them form the corresponding edges. An increase in dissimilarity along the edges signifies a coarser partition. In order to select the optimal number of clusters, a heuristic approach is to select the partition before a "jump," namely, between the chosen partition and the next one, there is a relatively high difference in the two merging distances.

In general, there is no strict rule to determine the exact number of clusters. However, when the next partition occurs at a notably higher dissimilarity measure, it becomes feasible to make a cut. Alternatively, the "elbow" method employing R^2 can be utilized. This involves selecting the number of clusters such that an increase by one results in only a marginal improvement in explained variability.

This type of method suffers from two significant disadvantages. Firstly, once a document is assigned to a particular cluster, it cannot be assigned to any other cluster. Secondly, the computational time complexity is $O(M^2)$, making it less suitable for large corpora.

2.3.2 Partitioning methods

Hard *K*-means

The basic idea behind this methodology (also known as prototype based) addresses the two primary issues of hierarchical clustering. Specifically, it eliminates the need for computing a dissimilarity matrix and produces a flat partition instead of a sequence of nested partitions.

By predefining the final number of clusters K, the method represents each generic cluster k with a centroid $\mathbf{h}_k$, which typically does not correspond to an actual data point [3]. The centroid matrix $\mathbf{H}$ is $K \times |V|$. Introducing the allocation matrix $\mathbf{U} \in M \times K$, where each element u_{ik} is a binary indicator indicating whether the i-th document is in the k-th cluster or not. In this hard version, the assumption is that a document can be assigned to only one cluster, such that $u_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} u_{ik} = 1$. The objective is to minimize the variance within clusters, formally:

$$\begin{cases} \min\limits_{\mathbf{U},\mathbf{H}} \sum\limits_{i=1}^{M}\sum\limits_{k=1}^{K}\sum\limits_{j=1}^{|V|} u_{ik}\left(dt_{ij} - \overline{dt}_{kj}\right)^2 = \min\limits_{\mathbf{U},\mathbf{H}} \sum\limits_{i=1}^{M}\sum\limits_{k=1}^{K}\sum\limits_{j=1}^{|V|} u_{ik}\left(dt_{ij} - h_{kj}\right)^2 \\ u_{ik} \in \{0,1\} \quad \forall i = 1,\dots,M \ , \ \forall k = 1,\dots,K \\ \sum\limits_{k=1}^{K} u_{ik} = 1 \quad \forall i = 1,\dots,M \end{cases}$$

[3] Conversely, in the Hard K-medoids, each cluster center corresponds to an actual data point.

To solve this constrained optimization, the following algorithm is required:

Step 0 randomly initialize the centroid matrix $\mathbf{H}^t$, where $t = 0$;

Step 1 update the allocation matrix $\mathbf{U}^{t+1}$ using the following rules:

$$u_{ik}^{t+1} = \begin{cases} 1 & k = \arg\min \sum_{j=1}^{|V|} \left(dt_{ij} - h_{kj}^{t} \right)^2 \\ 0 & \text{otherwise} \end{cases}$$

Step 2 update the centroid matrix $\mathbf{H}^{t+1}$ as follows:

$$h_{kj}^{t+1} = \frac{\sum_{i=1}^{M} u_{ik}^{t+1} * dt_{ij}}{\sum_{i=1}^{M} u_{ik}^{t+1}}$$

Step 4 if $|\mathbf{H}^{t+1} - \mathbf{H}^{t}| < \varepsilon$ convergence is reached and the algorithm can be stopped; otherwise, start again from Step 1.

The power of this algorithm lies in the random choice of centroids at Step 0, necessitating multiple iterations to avoid local minima. Furthermore, the authors in [2] propose K-Means ++, where the center of the cluster is chosen based on probabilities proportional to the Euclidean distance between a point and the previous center. In the K-Means scenario, the time complexity is linear and the optimal number of clusters can be selected again using the elbow method applied to the R^2 statistic for different partitions. However, the method has some drawbacks, including a tendency to produce spherical clusters primarily because it relies on the Euclidean distance, which does not take into account the variance and covariance structure of the data [4]. Moreover, it is sensitive to outlier values in the data as the centroids are strongly affected by them.

Fuzzy *K*-means

Up until now, the previous clustering methods assumed that each document could only belong to one cluster. However, it is possible to relax this assumption.

[4] Clusters have similar variances in all dimensions.

Now, each document can have a membership degree to belong to a certain cluster, denoted by u_{ik}, where $u_{ik} \in [0,1]$ [54]. The goal remains the same: to minimize the variance within clusters. Once again, just for the sake of simplicity, the use of the squared Euclidean distance is displayed while the cosine similarity is used.

Formally, the aim is to solve the following constrained minimization problem:

$$\begin{cases} \min\limits_{\mathbf{U},\mathbf{H}} \sum\limits_{i=1}^{M}\sum\limits_{k=1}^{K}\sum\limits_{j=1}^{|V|} u_{ik}^{m}\left(dt_{ij}-h_{kj}\right)^2 \\ u_{ik} \in [0,1] \quad \forall i = 1,\dots,M \ , \ \forall k = 1,\dots,K \\ \sum\limits_{k=1}^{K} u_{ik} = 1 \quad \forall i = 1,\dots,M \end{cases}$$

Here, m is the fuzziness parameter, typically between 1.5 and 2. The values that minimize this objective function can be found by solving the Lagrangian:

$$\mathcal{L} = \sum_{i=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{|V|} u_{ik}^{m}\left(dt_{ij}-h_{kj}\right)^2 - \sum_{i=1}^{M}\lambda_i\left(\sum_{k=1}^{K}u_{ik}-1\right) \tag{2.7}$$

By computing the derivatives with respect to the parameters involved:

$$\begin{cases} \frac{d\mathcal{L}}{du_{ik}} = m u_{ik}^{m-1}\sum\limits_{j=1}^{|V|}\left(dt_{ij}-h_{kj}\right)^2 - \lambda_i = 0 & \forall k = 1,\dots,K \ , \ \forall i = 1,\dots,M \\ \frac{d\mathcal{L}}{dh_{kj}} = \sum\limits_{i=1}^{M}(2h_{kj}u_{ik}^{m} - 2u_{ik}^{m}dt_{ij}) = 0 & k = 1,\dots,K \ , \ j = 1,\dots,|V| \\ \frac{d\mathcal{L}}{d\lambda_i} = \sum\limits_{k=1}^{K} u_{ik} - 1 = 0 & \forall i = \dots,M \end{cases}$$

Solving the previous minimization problem, it is possible to obtain that the values which minimize the objective function are:

$$h_{kj} = \frac{\sum\limits_{i=1}^{M} u_{ik}^{m} dt_{ij}}{\sum\limits_{i=1}^{M} u_{ik}^{m}} \tag{2.8}$$

$$u_{ik} = \frac{1}{\sum\limits_{k^*=1}^{K}\left(\frac{\sum\limits_{j=1}^{|V|}\left(dt_{ij}-h_{kj}\right)^2}{\sum\limits_{j=1}^{|V|}\left(dt_{ij}-h_{k^*j}\right)^2}\right)^{\frac{1}{m-1}}} \tag{2.9}$$

For the computation of the centroid of a cluster, the membership degree of the unit is considered. It is possible to generalize the previous algorithm as follows:

Step 0 randomly initialize the membership degree matrix $\mathbf{U}^t$, where $t = 0$;

Step 1 update the centroid matrix $\mathbf{H}^{t+1}$ using the updating rule in 2.8 and $\mathbf{U}^t$;

Step 2 update again the membership degree matrix $\mathbf{U}^{t+1}$ using again the rule contained in 2.9 and $\mathbf{H}^{t+1}$;

Step 4 if $|\mathbf{H}^{t+1} - \mathbf{H}^t| < \varepsilon$ or $|\mathbf{U}^{t+1} - \mathbf{U}^t| < \varepsilon$, convergence is reached and the algorithm stops. Otherwise, start again from Step 1.

Despite being more robust than its hard version, fuzzy clustering suffers from an increased computational complexity, no longer linear. Additionally, it still requires multiple initializations to avoid local optima and it still tends to produce spherical clusters. To assign a unit to a specific cluster, a heuristic approach can be employed by determining the argmax per row to identify the highest value of the membership coefficient for each unit.

Bisecting *K*-means

Bisecting K-Means is an hybrid variant of the basic K-Means, merging aspects of both divisive hierarchical clustering and non-hierarchical clustering, offering advantages such as deleting the need for a dissimilarity matrix and "automatically" determining the number of clusters. The algorithm proceeds as follows:

Step 0 starting with a single cluster containing all documents, the basic K-Means algorithm is applied to split it into two clusters;

Step l for each iteration from 2 to $M-1$, the cluster with the highest internal variance is selected and the basic K-Means algorithm is applied to split it further. This process continues until all singleton clusters are formed.

The authors in [38] propose to generalize this algorithm for document clustering, where the cluster to split is chosen using specific criteria (highest size, lower overall

similarity, etc.) and after each split, the procedure is repeated multiple times to select the split with the highest overall similarity. The optimal number of clusters can be determined using methods like dendrograms, similar to hierarchical clustering. Furthermore, the computational time complexity of this algorithm is linear, making it faster compared to the basic K-Means algorithm, as units are only compared to two centroids rather than K centroids.

2.3.3 Non-Euclidean Fuzzy Relational Clustering

It is possible that the information about the entire dataset is missing, but the similarity between pairs of documents is known. In this particular framework, it can be introduced a new approach, namely fuzzy relational clustering. Here, the term "relational" refers to the fact that only the similarity measures among units are possessed. One of the most important algorithms in this framework is FANNY [32], which however relies on the idea that the similarities should be defined based on the Euclidean distance. To address the aforementioned issue and extend the algorithm's applicability, the Non-Euclidean Fuzzy Relation Clustering (NEFRC) algorithm [17] can be employed. Following a similar notation to the fuzzy K-Means algorithm, the constrained minimization problem can be expressed as follows:

$$\begin{cases} \min\limits_{\mathbf{U}} \sum\limits_{k=1}^{K} \dfrac{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{M} u_{ik}^m u_{jk}^m D(\mathbf{d}_i, \mathbf{d}_j)}{2\sum\limits_{i=1}^{M} u_{ik}^m} \\ u_{ik} \in [0,1] \quad \forall i = 1, \dots, M \ , \ \forall k = 1, \dots, K \\ \sum\limits_{k=1}^{K} u_{ik} = 1 \quad \forall i = 1, \dots, M \end{cases}$$

where m is the general fuzzifier. In this case, the Lagrangian function can be written as:

$$\mathcal{L} = \sum_{k=1}^{K} \frac{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{M} u_{ik}^m u_{jk}^m D(\mathbf{d}_i, \mathbf{d}_j)}{2\sum\limits_{i=1}^{M} u_{ik}^m} - \sum_{i=1}^{M} \lambda_i \left(\sum_{k=1}^{K} u_{ik} - 1 \right) - \sum_{i=1}^{M}\sum_{k=1}^{K} \psi_{ik} u_{ik} \tag{2.10}$$

Computing the derivatives with respect to the involved parameters ($k = 1, \ldots, K$, $i = 1, \ldots, M$):

$$\begin{cases} \frac{d\mathcal{L}}{du_{ik}} = \frac{mu_{ik}^{m-1}\sum_{j=1}^{M} u_{jk}^{m} D(\mathbf{d}_i, \mathbf{d}_j)}{\sum_{i=1}^{M} u_{ik}^{m}} - \frac{mu_{ik}^{m-1}\sum_{i=1}^{M}\sum_{j=1}^{M} u_{ik}^{m} u_{jk}^{m} D(\mathbf{d}_i, \mathbf{d}_j)}{2\left(\sum_{i=1}^{M} u_{ik}^{m}\right)^{2}} - \lambda_i - \psi_{ik} = 0 \\ \frac{d\mathcal{L}}{d\lambda_i} = \sum_{k=1}^{K} u_{ik} - 1 = 0 \\ \frac{d\mathcal{L}}{d\psi_{ik}} = u_{ik} = 0 \end{cases}$$

Defining:

$$a_{ik} = \frac{m\sum_{j=1}^{M} u_{jk}^{m} D(\mathbf{d}_i, \mathbf{d}_j)}{\sum_{i=1}^{M} u_{ik}^{m}} - \frac{m\sum_{i=1}^{M}\sum_{j=1}^{M} u_{ik}^{m} u_{jk}^{m} D(\mathbf{d}_i, \mathbf{d}_j)}{2\left(\sum_{i=1}^{M} u_{ik}^{m}\right)^{2}} \qquad b_{ik} = a_{ik} u_{ik}^{m-2} \tag{2.11}$$

the derivative with respect to u_{ik} can be written in a different form which allow to solve the minimization problem in a simple way:

$$\frac{d\mathcal{L}}{du_{ik}} = u_{ik}^{m-1} a_{ik} - \lambda_i - \psi_{ik} = u_{ik} b_{ik} - \lambda_i - \psi_{ik} = 0 \rightarrow \lambda_i = \frac{1}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} - \frac{\sum_{k=1}^{K} \frac{\psi_{ik}}{b_{ik}}}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} \rightarrow$$

$$\rightarrow u_{ik} = \frac{\frac{1}{b_{ik}}}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} + \frac{\psi_{ik}}{b_{ik}} - \frac{\sum_{k=1}^{K} \frac{\psi_{ik}}{b_{ik}}}{b_{ik}\sum_{k=1}^{K} \frac{1}{b_{ik}}}$$

Using the Karush-Kuhn-Tucker conditions, it can be claimed that the optimal value for the membership degree is:

$$u_{ik} = \begin{cases} \frac{\frac{1}{b_{ik}}}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} & g \in I_i^{+} \\ 0 & g \in I_i^{-} \end{cases} \tag{2.12}$$

where:

$$I_i^+ = \left\{ k : \frac{\frac{1}{b_{ik}}}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} > 0 \right\} \quad I_i = \left\{ k : \frac{\frac{1}{b_{ik}}}{\sum_{k=1}^{K} \frac{1}{b_{ik}}} \leq 0 \right\}$$

The algorithm to find the optimal value for the above problem can be outlined as follows:

Step 1. Randomly initialize $\mathbf{U}^t$, $t = 0$;

Step 2. update b_{ik}^{t+1} using 2.11 with u_{jk}^{t+1} if $j < i$, otherwise u_{jk}^t ($\forall i = 1, \ldots, M$, $\forall k = 1, \ldots, K$);

Step 3. update u_{ik}^{t+1} according to 2.12 ($\forall i = 1, \ldots, M$, $\forall k = 1, \ldots, K$);

Step 4. if $|\mathbf{U}^{t+1} - \mathbf{U}^t| < \varepsilon$ convergence is reached, otherwise repeat from **Step 2**.

Although this algorithm is highly effective in identifying hidden patterns in similarity measures, it is not without limitations. Its scalability is one of them as well as the huge influence of the sparsity of a similarity matrix, which can affect its performance.

2.3.4 Model-Based methods

As discussed in Subsection 2.1, another common clustering method involves using a finite mixture model. Here, each document is considered an independent multivariate sample from a random variable that originates from one of K different sub-populations. However, the assignment of each document $\mathbf{d}_i$ to a cluster is initially unknown, making the data incomplete. To handle this, an indicator vector variable $\mathbf{Z}_i$ is defined, assuming that each entry can take as value either 0 or 1, with the property that they sum up to 1 across all clusters:

$$Z_{ik} = \begin{cases} 1 & i \in k \\ 0 & \text{otherwise} \end{cases} \qquad \sum_{k=1}^{K} Z_{ik} = 1 \tag{2.13}$$

The idea is that each unobserved indicator random variable Z_{ik} follows a Multinomial distribution with weights $\pi_k = \mathbb{P}(Z_{ik} = 1)$, which can be considered as a prior distribution on the mixing components. In particular, a priori they do not depend on the specific document. These weights π_k are probabilities lying in the $(K-1)-$dimensional simplex, where $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$. More formally, defining $f_k(.|\theta_k)$ as the generic mixing component, the marginal distribution of a document can be written as:

$$f(\mathbf{d}_i|\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \pi_k f_k(\cdot|\theta_k) \tag{2.14}$$

To ensure cohesion within clusters and separation among clusters, the mixing densities are chosen from the same parametric family ($f_k(\cdot|\theta_k) = f(\cdot|\theta_k)$). Each mixing component is represented as a multivariate Gaussian distribution parameterized by its mean μ_k and variance-covariance matrix $\mathbf{\Sigma}_k$. Therefore, the parameter vector $\Psi = \{\mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_{K-1}, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K\}$ encapsulates these model parameters. The complete data likelihood of the model can be expressed as follows:

$$\mathbb{L}(\Psi) = \prod_{i=1}^{M} f(\mathbf{d}_i, \mathbf{Z}_i|\Psi) = \prod_{i=1}^{M} f(\mathbf{d}_i|\mathbf{Z}_i, \Psi) f(\mathbf{Z}_i|\Psi) = \prod_{i=1}^{M} \prod_{k=1}^{K} \mathbb{MVN}(\mathbf{d}_i|\mu_k, \mathbf{\Sigma}_k)^{Z_{ik}} \pi_k \tag{2.15}$$

Therefore, the complete data log-likelihood can be derived as follows:

$$l(\Psi) = \sum_{i=1}^{M} \sum_{k=1}^{K} Z_{ik} \log\left(\mathbb{MVN}(\mathbf{d}_i|\mu_k, \mathbf{\Sigma}_k)\pi_k\right) \tag{2.16}$$

To maximize the complete data log-likelihood and estimate the parameters Ψ, an Expectation-Maximization (EM) algorithm is employed. This iterative algorithm proceeds as follows: given the estimated parameters at iteration t denoted by $\widehat{\Psi}^t$, the algorithm computes the next iteration in two phases. In the E-step, the algorithm calculates the expected value of the complete data log-likelihood conditioned on the observed data and the current estimate $\widehat{\Psi}^t$:

$$Q(\Psi|\mathcal{C}, \widehat{\Psi}^t) = \mathbb{E}[l(\Psi)|\mathcal{C}, \widehat{\Psi}^t] = \mathbb{E}\left[\sum_{i=1}^{M} \sum_{k=1}^{K} Z_{ik}|\mathcal{C}, \widehat{\Psi}^t\right] = \sum_{i=1}^{M} \sum_{k=1}^{K} \mathbb{P}(Z_{ik}|\mathcal{C}, \widehat{\Psi}^t)$$

The E-step reduces to the computation of the posterior inclusion probabilities of a document inside a cluster for each document inside the corpus and for each cluster. More into detail, using the Bayes updating rule ($\forall k = 1, \ldots, K\ ,\ \forall i = 1 \ldots, M$):

$$\widehat{w}_{ik}^{t+1}|\mathbf{d}_i, \widehat{\Psi}^t = \frac{\widehat{\pi}_k^t \mathbb{MVN}(\mathbf{d}_i|\widehat{\mu}_k^t, \widehat{\mathbf{\Sigma}}_k^t)}{\sum_{k=1}^{K} \widehat{\pi}_k^t \mathbb{MVN}(\mathbf{d}_i|\widehat{\mu}_k^t, \widehat{\mathbf{\Sigma}}_k^t)} \tag{2.17}$$

In the M-step, the algorithm maximizes the expected value of the complete data log-likelihood conditioned on $\widehat{\Psi}^t$. Closed-form estimations for the parameters can be computed as detailed by [24]. When the convergence of the EM algorithm is reached, a clustering partition can be defined using the Maximum A Posteriori (MAP) criterion. Namely:

$$\widehat{Z}_{ik} = \begin{cases} 1 & k = \underset{k=1,\ldots,K}{arg\,\max}\ \widehat{w}_{ik} \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

The primary challenges of model-based clustering include selecting appropriate initial values for the EM algorithm and determining the optimal number of mixture components. Addressing the first challenge typically involves running the algorithm with multiple random starts. Alternatively, the initialization can start from the partition obtained by an agglomerative hierarchical method (MBHAC [4]). The second challenge can be tackled using criteria such as the Bayes factor or the Akaike Information Criterion (AIC). The Bayes factor compares the ratio of posterior to prior odds of two models using Jeffrey's scale, while AIC is a penalized likelihood method where the penalization takes into account the number of parameters in the entire model.

2.3.5 Density-Based method

The most commonly used density-based method is **DBSCAN** [23] (Density-Based Spatial Clustering of Applications with Noise). The fundamental concept of this method is that clusters of points can typically be identified as high density regions, while points outside these regions are regarded as "noise". While this method does not directly employ a graph representation for the clustering problem, it aims to identify density-connected regions that can be thought of as similar to connected components in a graph.

Before delving into the explanation of the method, it's important to define several key tools. Given a corpus $\mathcal{C}$, a distance function $D(\cdot, \cdot)$, a value ε and a minimum number of points m:

Definition 1. *The ε-neighborhood of a document $\mathbf{d}$ is defined as the set of all the documents within a distance ε with respect to $\mathbf{d}$. Formally:*

$$N_\varepsilon(\mathbf{d}) = \{\mathbf{d}^\star \in \mathcal{C} | D(\mathbf{d}, \mathbf{d}^\star) \leq \varepsilon\} \tag{2.19}$$

If the cardinality of the ε-neighbourhood of a document $\mathbf{d}$ is greater than m, then $\mathbf{d}$ is defined as a "core" point, while in the other case it can be either a "border" point or a "noise" point[5]. In particular, both the parameters ε and m are employed to distinguish between points.

Definition 2. *A document $\mathbf{d}$ is directly density-reachable from $\mathbf{d}^*$ with respect to ε and m if:*

- $\mathbf{d} \in N_\varepsilon(\mathbf{d}^*)$
- $|N_\varepsilon(\mathbf{d}^*)| \geq m$, *namely $\mathbf{d}^*$ is a core point.*

It is evident that the above relationship it is not symmetric because the starting point must be a core point.

Definition 3. *A document d is density-reachable from $\mathbf{d}^*$ with respect to ε and m if there exist a sequence of documents $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ ($\mathbf{d}_1 = \mathbf{d}$, $\mathbf{d}_n = \mathbf{d}^*$) connecting the two documents, such that each pair of consecutive documents $(\mathbf{d}_i, \mathbf{d}_{i+1})$ is directly density-reachable.*

As the previous case, density-reachability is not a symmetric relationship and a new linkage should be defined .

Definition 4. *A document $\mathbf{d}$ is density-connected to $\mathbf{d}^*$ with respect to ε and m if there exists a document $\mathbf{d}^+$ such that $\mathbf{d}$ and $\mathbf{d}^*$ are directly density-reachable from $\mathbf{d}^+$*

Definition 5. *A cluster k is a non-empty subset of $\mathcal{C}$ such that ;*

[5] **DBSCAN*** does not make the distinction between border and noise points.

- $\forall \mathbf{d}, \mathbf{d}^* \in \mathcal{C}$: *if* $\mathbf{d} \in k$ *and* $\mathbf{d}^*$ *is density-reachable from* $\mathbf{d}$, *then* $\mathbf{d}^* \in k$;
- $\forall \mathbf{d}, \mathbf{d}^* \in k$: $\mathbf{d}$ *is density-connected to* $\mathbf{d}^*$.

The DBSCAN algorithm proceeds by fixing values for ε and m, choosing an arbitrary document $\mathbf{d}$ and identifying the subset of density-reachable documents from $\mathbf{d}$. If $\mathbf{d}$ is a core point, a cluster is formed by including all documents within the ε-neighborhood, as well as those directly-reachable from these documents, and so forth. If $\mathbf{d}$ is a border point, no further density-reachable documents exist and the algorithm moves to another unexplored document to repeat the process. At the end of the process, documents not assigned to any cluster based on ε and m are labeled as "noise" and grouped into a single noise cluster. The challenge lies in effectively selecting the hyperparameters ε and m, which significantly impact the clustering analysis. One approach involves fixing m equal to r and compute the distances to the r-th nearest neighbor for all documents, sorting these distances in ascending order and selecting ε corresponding to the "elbow" of the curve.

Although DBSCAN offers several advantages (such as no need to specify the number of clusters a priori, noise point definition, flexibility in choosing distance functions etc.), it shows poor performances with high-dimensional data. Additionally, it operates with quadratic time complexity. An extension of this method, known as **HDBSCAN** (Hierarchical DBSCAN), gives a natural hierarchical extension to the previous method [11].

2.3.6 Graph-Based methods

The main concept of this method involves representing a corpus $\mathcal{C}$ as an undirect weighted graph, where each node represents a document and the edges denote similarities between pairs of documents. Then, the idea is that clusters can be represented by the different connected components in the graph. Two key questions arise: what measure of similarity is employed and how is the graph $G = (V, E)$ constructed?

While cosine similarity can be used, an alternative approach involves introducing

a new similarity measure using kernels [6]. To begin, recall that in an input space $\mathcal{C}$, a kernel is defined as a function $\mathcal{K} : \mathcal{C} \times \mathcal{C} \rightarrow \Re$. Mercer's theorem states that this function is a kernel if and only if there exists a feature space $\Re^{|V|}$ with an inner product $\langle \cdot, \cdot \rangle$ and a mapping function $\phi : \mathcal{C} \rightarrow \Re^{|V|}$, such that for every pair of documents $\mathbf{d}_i$ and $\mathbf{d}_j$:

$$\mathcal{K}\left(\mathbf{d}_i, \mathbf{d}_j\right) = \langle \phi(\mathbf{d}_i), \phi(\mathbf{d}_j) \rangle \tag{2.20}$$

Cosine similarity can indeed be interpreted as a kernel function when considering the mapping $\phi(\mathbf{d}_i) = \frac{\mathbf{d}_i}{|\mathbf{d}_i|}$. However, one of the most commonly used kernels in this context is based on strings matching [41]. Given two documents and a value l, the similarity between them can be measured by calculating the number of common substrings of length l. In principle, the more common substrings two documents share, the more similar they are. Mathematically:

$$\mathcal{K}\left(\mathbf{d}_i, \mathbf{d}_j\right) = \sum_{t \in A} \text{num}_t(\mathbf{d}_i) \text{num}_t(\mathbf{d}_j) \lambda \tag{2.21}$$

Here, A is the set of all strings of length l in the two documents, num_t counts how many times the substring t appears in the two documents and λ is a decay factor associated with each substring of length l. Different forms of λ result in different kernel functions (e.g., Constant, Exponential, Boundrange).

With this kernel, it becomes possible to construct a similarity matrix for the entire corpus. The simplest way to construct an undirected weighted graph is to assume that all vertices are linked (complete connectivity), with the weight associated with each edge being equal to the aforementioned similarity measure. Furthermore, let $\mathbf{\Omega}$ be the weighted adjacency matrix. If a sparser matrix is desired, a K-nearest neighbor graph can be computed. Specifically, two vertices are linked if one of them is among the K-nearest neighbors of the other and the weight associated with the edge is the usual similarity measure [7].

As highlighted by [61], to employ spectral clustering, it is necessary to introduce

[6] Cosine can be considered as a kernel.

[7] Another construction scheme based on the ε-neighborhood also exists.

the unnormalized Laplacian matrix:

$$\mathbf{L} = \mathbf{D} - \mathbf{\Omega} \tag{2.22}$$

where $\mathbf{D}$ is the degree matrix, a diagonal matrix where each element on the diagonal represents the sum of the weights between that vertex and all others. Not delving too deeply into the details, $\mathbf{L}$ is a semi-positive definite matrix and the number of connected components in the original graph equals the number of null eigenvalues of $\mathbf{L}$. Although clustering algorithms are typically applied to the normalized version of the Laplacian matrix, similar considerations apply:

$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{\Omega}\mathbf{D}^{-\frac{1}{2}} \tag{2.23}$$

The most well-known spectral clustering algorithm comprises the following steps [51]:

Step 1. construct the weighted graph G associated to one of three possible construction of the graph as specified earlier. Form the weighted adjacency matrix Ω;

Step 2. compute the normalized Laplacian matrix $\mathbf{L}_{sym}$ 2.23;

Step 3. compute the first K eigenvectors;

Step 4. build a new matrix using the previously computed eigenvectors;

Step 5. normalize the rows of the previous matrix to have unit length;

Step 6. apply the K-Means algorithm on the new matrix. [8]

The main advantage of this method is that it transforms the clustering problem from $\Re^{|V|}$ to $\Re^{K}$, with $K << |V|$. Some of the main drawbacks of spectral clustering include the necessity to select an appropriate similarity measure, increased computational time with graph complexity and the challenge of determining the optimal number of clusters.

[8] the author in [13] applies a fuzzy K-Medoids approach in this step.

2.4 Cluster Validity

After delving into the definitions of various techniques, it's crucial to introduce the metrics used for assessing the quality of a clustering partition. Specifically, despite of the typical methods employed to determine the optimal number of clusters, it's imperative to establish a measure to compare different algorithms. To elaborate further, when the clustering partition of the documents is known beforehand, it's feasible to compute *external* measures. Conversely, if this information is unavailable, one can compute *internal* measures [66] [38]. Note that external measures are used also in a document classification type of approach.

Assuming that in the original dataset documents are labelled within q classes, two key validity measures are:

- **Entropy**: it assesses the distribution of documents across different classes within various topics. Ideally, a perfect partition would result in singleton clusters. To compute entropy, each cluster's probability distribution of documents across classes is estimated. This involves computing the probability p_k^i that a document in cluster k belongs to class i based on the number n_k^i of elements of cluster k in class i and the total number n_k of elements in cluster k:

$$\widehat{p}_k^i = \frac{n_k^i}{n_k} \qquad E_j = -\sum_i \widehat{p}_k^i \log\left(\widehat{p}_k^i\right) \tag{2.24}$$

  the total entropy of a partition is obtained by computing a weighted average of the entropy within each cluster, with the weights proportional to the cluster sizes. The goal is to minimize entropy in order to maximize the relevance of the query. In particular, the minimization is achieved when each cluster represents perfectly the answer of a query, i.e. when the cluster is coincident with the class;

- **F-measure** [39]: evaluates the effectiveness of clustering based on recall (sensitivity) and precision, linked to the concept of a confusion matrix. Recall measures the proportion of correctly classified documents over the total number of documents in a class, while precision measures the proportion of correctly

classified documents over the total number of documents estimated to belong to a cluster. For each class, multiple (K) indicators are computed using the harmonic mean between recall and precision and the maximum value is chosen. A weighted average of these indicators yields a composite measure whereas higher values are better.

$$R_{ik} = \frac{n_k^i}{n^i} \quad P_{ik} = \frac{n_k^i}{n_k} \quad F_{ik} = \frac{2R_{ik}P_{ik}}{R_{ik} + P_{ik}} \quad F = \sum_i \frac{n^i}{M} \max_k \{F_{ik}\} \tag{2.25}$$

Additionally, when external information is unavailable, a heuristic approach involves the computation of a weighted average of the intra-cluster similarity. This measure assesses the similarity between the centroid of a cluster and its constituent elements, typically calculated using cosine similarity. It is knows as **Overall Similarity** and it lies in the interval $[-1, 1]$ due to the value assumed by the cosine similarity. Consequently, higher values indicate a higher cohesion inside clusters. However, due to the nature of this measure, it tends to favor partitions with a higher number of clusters, as the cosine similarity between a unit and itself is always one.

$$\mathrm{CS}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \cos(\mathbf{dt}_{i\cdot}, \mathbf{h}_{c\cdot}) \qquad \mathrm{CS} = \sum_{k=1}^{K} \frac{n_k}{M} \mathrm{CS}_k \tag{2.26}$$

Given that the Overall Similarity is a monotonic measure in the number of clusters and it is only a measure of cohesion, it is crucial to introduce a new measure, specifically the **Silhouette**. The Silhouette assesses the similarity of a document with respect to its assigned cluster and its dissimilarity from other clusters. Define a_i as the average distance between the i-th document and all the others in its clusters and b_i as the minimum distance between the i-th document and all the documents not inside the same cluster:

$$S(i) = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad S = \frac{1}{M} \sum_{i=1}^{M} S(i) \tag{2.27}$$

$S(i)$ lies in the interval $[-1, 1]$, where higher values mean that a document well-suited in its cluster and a value of -1 suggests that a document might be better placed in a neighboring cluster. Then, the average value for all the documents is computed to provide a composite measure. The simplistic idea is that the optimal value of the number of clusters is the one which maximize the average Silhouette. It is important

to note that working in a high dimensional regime could affect the stability of this measure due to the usual curse of dimensionality affecting the Euclidean distance. From an interpretative standpoint, one can employ the following rule of thumb to measure the quality of the average silhouette value. If:

- $S \in (0.7, 1]$, the score is very good;
- $S \in (0.5, 0.7]$, the score is good;
- $S \in (0.25, 0.5]$, the score is neutral;
- $S \in [-1, 0.25]$, the score is bad.

Given the distinct nature of fuzzy clustering approaches, a fuzzy version of entropy can be employed. While its interpretation differs from traditional entropy used in information retrieval, fuzzy entropy serves as a measure of variability within a cluster partition. The aim is to obtain a small value of this measure. Furthermore, it also exists a fuzzy version of the silhouette which directly takes into account the allocation matrix into the aggregate index. In particular:

$$FS = \frac{\sum_{i=1}^{M} (\mu_{ik} - \mu_{ik'})^{\alpha} S(i)}{\sum_{i=1}^{M} (\mu_{ik} - \mu_{ik'})^{\alpha}} \tag{2.28}$$

where u_{ik} and $u_{ik'}$ represent the highest and second-highest membership degrees, respectively, for document i. The generic weight, α, is also considered. In this case, the optimal number of clusters is determined by the value which maximizes the fuzzy silhouette.

As a final remark, I would like to emphasize that while external measures are always a viable option, internal measures are not specifically designed for this particular task. In the context of document clustering, which often involves high-dimensional and sparse data, using internal measures can lead to anomalous results.

Chapter 3

Hypothesis Test Based Clustering Algorithms

3.1 Introduction

Having outlined the principal algorithms employed in document clustering in Chapter 2, it becomes evident that the principal limitations, as outlined in Section 2.3, are that they operate in a high-dimensional space and it is difficult to interpret the different clusters once a partition has been obtained. Indeed, there exist two methods which have a very good performance in both interpretation and clustering, namely BERTopic [26] and Top2Vec [1], although they no longer rely on the BOW context, but rather on embeddings and, more generally, language models. It may be the case to introduce a new method which relies on their ideas to solve the problem.

The proposed methods for computing document clustering employs a well known approach, i.e. Tandem analysis, a two-stage process. Initially, it can be observed that the information contained within the DTM exhibits significant sparsity, so a direct application of a clustering technique would be highly inefficient, specially if Euclidean distance is employed. Consequently, a logical step is to employ dimensionality reduction techniques to streamline the data and extract the majority of the information. As outlined in the previous Chapter, methods such as PCA, SVD or

Uniform Manifold Approximation and Projection (UMAP) have already been used to accomplish this task by BERTopic and Top2Vec. However, the proposed strategy involves employing LDA to identify the main topics in the corpus under analysis. To be honest, the principal aim of LDA is not to reduce the dimensionality of the data matrix, rather to find common topics inside a corpus and define a set of words linked to each topic. However this approach has the potential to reduce the dimensionality of the DTM (from a $M \times |V|$ matrix to a $M \times G$ matrix, with $G << |V|$), with the advantage of a better interpretability rather than other methods.

Before the introduction of the clustering technique, it is important to introduce a new measure of similarity between documents. Without delving into detail, the idea is that this new measure compares the distribution of the topics inside two documents computing p-value associated to the hypothesis test of distributions homogeneity via a Bootstrap approach, having as test statistic the Kullback-Leibler divergence. Subsequently, two different techniques can be deployed to cluster the documents effectively.

3.2 Latent Dirichlet Allocation

LDA, as introduced by [9] (and later generalized by [7]), represents one of the most important techniques within the unsupervised learning paradigm. Its primary objective is the identification of "latent variables" in a collection of documents, specifically uncovering the underlying topics of interest within a given corpus. In a nuanced exploration, LDA operates on the premise that, within a collection of documents, each of them can be conceptualized as a mixture distribution across a predetermined set of topics. Moreover, it postulates exchangeability of each document conditioned on the topics which is noteworthy, given its implication of pairwise uncorrelatedness among documents [1]. Furthermore, LDA assumes homogeneity of topics throughout the entire corpus. This assumption adds an additional layer of constraint, asserting that the topics remain consistent across the entire document collection.

[1] in a further paper [7] this assumption was relaxed.

Since topics are latent variables, not directly observable, a novel path must be defined for investigation. The fundamental concept behind LDA is to view each topic as a mixture distribution over the entire dictionary. This implies that documents sharing similar words also share the same underlying topic. Furthermore, words are assumed to be exchangeable within each document given the topics, following the BOW representation. In order to be coherent with the previous assumptions, it is evident that the r-th word inside the i-th document, say $\mathbf{w}_j^r$ is assigned to an underlying reference topic, which can be denoted as z_j. This concept suggests that a topic can also be expressed through its one-hot encoded version, designated as $\mathbf{Z}_i^r$, a G-dimensional vector with a value of one corresponding to the reference topic j and zeros in all other positions. Here, G represents the predetermined number of topics within the corpus.

In light of these considerations, the focus shifts to the learning of two distinct sets of elements:

- the matrix $\theta \in M \times G$ (Document-Topic matrix), where each entry represents the probability distribution of the generic topic j inside document i. Basically, each row can be interpreted as the mixing weight associated to the relative topic for a document and so it lies in the $G-1$ dimensional simplex;
- $\beta \in G \times |V|$ (Topic-Word matrix), where each entry represents the probability distribution of a generic word given a specific topic. Each row in this matrix represents the mixing weights associated with each word for the generic topic and also in this case, they lies in the $|V|-1$ dimensional simplex.

I had not previously mentioned that LDA can be viewed as an instance of a Bayesian model, specifically within the framework of mixture models and Dirichlet processes. Without delving too deeply into the intricacies, the fundamental concept is that two distinct Multinomial-Dirichlet models are blended, one at the document level and the other at the word level. In the simplest situation, assuming that $Y_i \sim \mathfrak{Multinomial}_u(1, \psi)$ and $\psi \sim \mathfrak{Dirichlet}(\eta)$, the posterior distribution is conjugate with respect to the prior distribution.

In particular:

$$\pi(\psi|\mathbf{Y}) \propto f\left(\mathbf{Y}|\psi\right)\pi(\psi) \propto \prod_{j=1}^{u} \psi_i^{\sum_{i=1}^{N} Y_{i,j}} \prod_{j=1}^{u} \psi_j^{\eta_j - 1} = \prod_{j=1}^{u} \psi_j^{\left(\sum_{i=1}^{N} Y_{i,j} + \eta_j\right) - 1}$$

The entire generative process can be succinctly summarized as follows:

1. $\theta_{i\cdot} \sim \mathfrak{Dirichlet}_G(\alpha) \quad i = 1, \ldots, M$
2. $\beta_{j\cdot} \sim \mathfrak{Dirichlet}_{|V|}(\eta) \quad j = 1, \ldots, G$
3. $\mathbf{Z}_i^r|\theta_{i\cdot} \sim \mathfrak{Multinomial}_G(1, \theta_{i\cdot}) \quad i = 1, \ldots, M \ , \ r = 1, \ldots, n_{\mathbf{d}_i}$
4. $\mathbf{w}_j^r|Z_{rj}^i, \beta \sim \mathfrak{Multinomial}_{|V|}\left(1, \beta_{j\cdot}\right) \quad j = 1, \ldots, G \ , \ i = 1, \ldots, M \ ,$
   $r = 1, \ldots, n_{\mathbf{d}_i}$

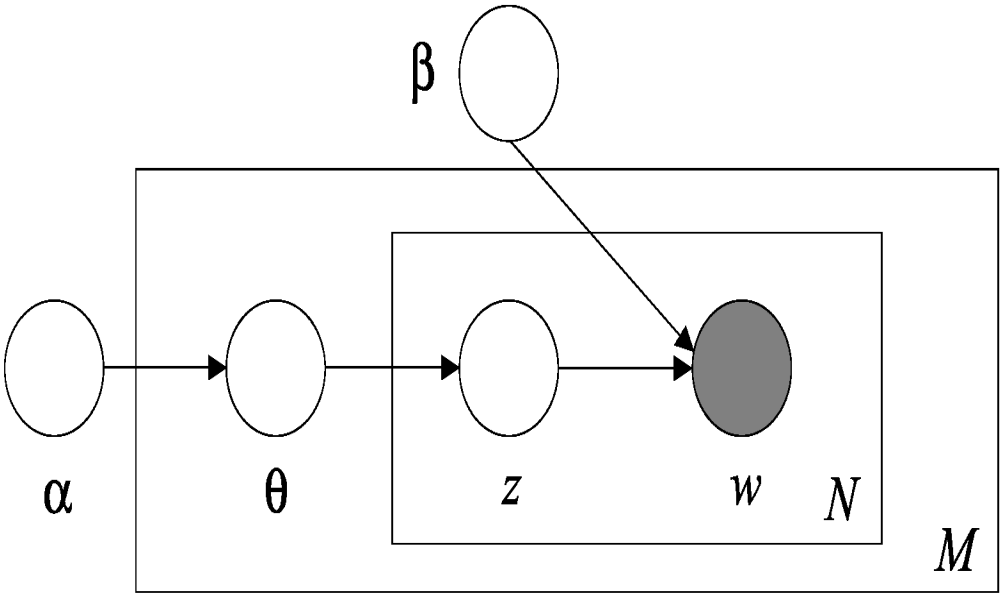

Figure 3.2.1. LDA graphical model [9].

It is pretty straightforward the derivation of the likelihood for the corresponding

model:

$$\pi\left(\mathcal{C}, \mathbf{Z} \mid \beta, \theta\right) = \prod_{i=1}^{M} \pi\left(\mathbf{d}_i \mid \mathbf{Z}_i, \beta\right) \cdot \pi\left(\mathbf{Z}_i \mid \theta_{i\cdot}\right) \propto \prod_{i=1}^{M} \prod_{r=1}^{n_{\mathbf{d}_i}} \prod_{j=1}^{G} \pi\left(\mathbf{w}_j^r \mid Z_{rj}^i, \beta\right) \theta_{ij}^{Z_{rj}^i} =$$

$$= \prod_{i=1}^{M} \prod_{r=1}^{n_{\mathbf{d}_i}} \prod_{j: Z_{rj}^i = 1} \pi\left(\mathbf{w}_j^r \mid Z_{ir}^j, \beta_{j\cdot}\right) \prod_{r=1}^{n_{\mathbf{d}_i}} \prod_{j=1}^{G} \theta_{ij}^{Z_{rj}^i} \propto$$

$$\propto \prod_{i=1}^{M} \prod_{r=1}^{n_{\mathbf{d}_i}} \prod_{j: Z_{rj}^i = 1} \prod_{v=1}^{|V|} \beta_{jv}^{w_{jv}^r} \prod_{j=1}^{G} \theta_{ij}^{\sum_{r=1}^{n_{\mathbf{d}_i}} Z_{rj}^i} = \prod_{i=1}^{M} \prod_{r=1}^{n_{\mathbf{d}_i}} \prod_{j=1}^{H} \prod_{v=1}^{|V|} \beta_{jv}^{w_{jv}^r * Z_{rj}^i} \prod_{j=1}^{G} \theta_{ij}^{\sum_{r=1}^{n_{\mathbf{d}_i}} Z_{rj}^i}$$

In the Bayesian framework, a fundamental principle is that the posterior distribution is proportional to the product of the likelihood and the prior distribution. In this context, when determining the hyperparameters for the two Dirichlet priors, you have the option to choose either the symmetric case or set them all to be equal to the inverse of the number of topics. Despite the existence of a closed form for the posterior distribution, the intricate dependencies among the LDA latent variables make analytical solutions challenging. Consequently, the use of computational methods (Gibbs Sampler and Variational Inference) is needed to approximate the posterior distribution.

In the context of selecting the optimal number of topics in topic modeling, the process involves maximizing the likelihood of the model, which is crucial for language modeling. Perplexity, introduced in 1977 by [31], serves as a measure to evaluate models by calculating the inverse of the geometric mean per word likelihood on a test set, where lower values indicate better performance [2]. However, since perplexity tends to favor higher numbers of topics, coherence measures are often used alongside it. Coherence measures, as proposed by [47] and [63], assess the semantic similarity within top-ranked words per topic (usually the first ten) and then an aggregate score is provided. More formally, the coherence of topic j is defined as the sum of the coherence score for all the possible couple of words in the n top-ranked. The two most used scores are the UCI [50] and the UMASS [47], which are respectively defined as:

$$\text{UCI}\left(\mathbf{w}_i, \mathbf{w}_{i^*}\right) = \log \frac{\mathbb{P}(\mathbf{w}_i, \mathbf{w}_{i^*})}{\mathbb{P}(\mathbf{w}_i)\mathbb{P}(\mathbf{w}_{i^*})} \quad \text{UMASS}\left(\mathbf{w}_i, \mathbf{w}_{i^*}\right) = \log \frac{\mathbb{P}(\mathbf{w}_i, \mathbf{w}_{i^*}) + 1}{\mathbb{P}(\mathbf{w}_i)}$$

[2] This is because it computes the exponential of the negative log-likelihood.

However, the optimal number of topics will be selected using four additional metrics which are already implemented in the ldatuning package. These metrics consider perplexity and coherence measures separately to provide a comprehensive view of the optimization problem.

Once the appropriate number of topics is determined, assessing the importance of each topic within the corpus can be done using a significance degree method introduced by [42]. This method computes a score for each topic in each document based on its importance and on its prevalence in the document, obtaining an $M \times G$ matrix. Then these scores are aggregated across the entire corpus for each different topic in order to obtain a measure which reflects its importance.

$$\begin{aligned} S\left(z_j, \mathbf{d}_i\right) &= \theta_{ij} \log\left(\frac{\sum_{i^*=1}^{M} \theta_{i^*j}}{\theta_{ij}}\right) \quad j = 1, \ldots, G \ , \ i = 1, \ldots, M \\ S(z_j) &= \sum_{i=1}^{M} S\left(z_j, \mathbf{d}_i\right) \quad j = 1, \ldots, G \end{aligned} \tag{3.1}$$

3.3 Test the Homogeneity of Topic Distributions

In Chapter 2 a comprehensive overview of the most commonly used similarity measures was presented. However, as pointed out in Section 3.1, it is important to introduce an additional measure to assess whether two documents exhibit substantial similarity.

The heuristic idea of comparing two pairs of different documents was outlined in [35], which postulates that two different documents can be considered "similar" if they share the same topics. In fact, given a fixed number of topics G, it is possible to use the posterior distribution of the Document-Topic matrix to compare two documents. More specifically, denoting as $\widehat{\theta}$ the posterior distribution of topics within documents and as $\widehat{\beta}$ the posterior distribution of words within topics, its primary objective is to evaluate whether the distributions of topics in any two given

documents, represented as $\mathbf{d}_i$ and $\mathbf{d}_{i*}$, are equivalent. Formally:

$$\begin{cases} H_0 : (\theta_{\mathbf{d}_i 1}, \ldots, \theta_{\mathbf{d}_i G}) = \left(\theta_{\mathbf{d}_{i*} 1}, \ldots, \theta_{\mathbf{d}_{i*} G}\right) \\ H_1 : (\theta_{\mathbf{d}_i 1}, \ldots, \theta_{\mathbf{d}_i G}) \neq \left(\theta_{\mathbf{d}_{i*} 1}, \ldots, \theta_{\mathbf{d}_{i*} G}\right) \end{cases} \tag{3.2}$$

In order to measure how much the two distributions are similar, the Kullback-Leibler divergence can be employed (2.4), as well as the Bhattacharyya distance (2.5). However, the p-value associated with the hypothesis test has to be computed. A p-value represents the probability, under the null hypothesis, of observing a test statistic value that contradicts (namely it is more extreme than) the observed value. Since the probability distribution of the test statistic in unknown a priori, another method must be defined to compute the p-value.

The (non-parametric) Bootstrap method, introduced by [22], is a suitable approach for this purpose. The essence of the Bootstrap method is to treat the original sample as a pseudo-population sharing the same characteristics as the actual population. Then, it assumes that each unit in the sample is independent and has the same sample weight. Concerning LDA, documents are not independent but they are conditional independent given topics. The procedure is straightforward. In fact, it is possible to sample with replacement from the pseudo-population and compute the test statistic estimate. Sampling with replacement from the pseudo population is equivalent to drawing a sample from a Multinomial distribution with equal weights. This process is repeated a fixed number of times, denoted as B and a sequence of estimated test statistic is generated.

What makes the Bootstrap method advantageous is its independence from an underlying probability distribution linked to the test statistic, which in the case under consideration is unknown. This characteristic makes it suitable for estimating probabilities, such as the p-value. To determine the p-value using Bootstrap, the comparison between the observed test statistic value, denoted as T^{obs} and the test statistic values obtained from the Bootstrap sample, denoted as T^b, is required. Due to the fact that higher values of the Kullback-Leibler divergence (Bhattacharyya distance) mean that two distributions are dissimilar, the p-value is estimated by

calculating the proportion of times the estimated test statistic in the Bootstrap samples exceeds the observed value. More formally:

$$I = \mathbb{P}(T > T^{obs}|H_0) \quad \text{and} \quad \widehat{I} = \frac{1}{B}\sum_{b=1}^{B}\mathcal{I}\{T^b > T^{obs}\} \tag{3.3}$$

Two distinct considerations can be discussed. Firstly, in the absence of an underlying probability distribution, Bootstrap methods do not exhibit the full suite of asymptotic properties of the usual Monte Carlo ones. Nevertheless, Bootstrap methodology stands out as a robust alternative. Secondly, it is feasible to compute confidence intervals of the estimated proportions. Given the nature of the parameter, only the Standard Normal confidence interval at level $1-\alpha$ can be computed.

$$CI_{1-\alpha} = \widehat{I} \pm q_{z,\alpha/2}se(\widehat{I}) = \widehat{I} \pm q_{z,\alpha/2}\sqrt{\frac{\widehat{I}(1-\widehat{I})}{B}} \tag{3.4}$$

Following this brief discussion about the Bootstrap methodology, the method to be employed in order to estimate the p-value linked with the hypothesis test of homogeneity of topic distributions can be easily introduced:

1) fit the LDA model on the entire corpus $\mathcal{C}$, determine the optimal number of topics using perplexity and coherence measures and then obtain the estimates $\widehat{\theta}$ and $\widehat{\beta}$ using either a Gibbs Sampler or Variational Inference. Also, establish a significance level α;

2) for each pair of distinct documents $\mathbf{d}_i$ and $\mathbf{d}_{i^*}$, where $i \neq i^*$, calculate the test statistic considering the corresponding rows of the matrix $\widehat{\theta}$ as reference probability distributions. If employing the Kullback-Leibler divergence, the test statistic $T^o_{ii^*}$ is defined as::

$$T^o_{ii^*} = \sum_{j=1}^{G}\widehat{\theta}_{\mathbf{d}_i j}\log\left(\frac{\widehat{\theta}_{\mathbf{d}_i j}}{\widehat{\theta}_{\mathbf{d}_{i^*} j}}\right)$$

at the end, an $M \times M$ matrix, denoted as $\mathbf{T}^o$, is obtained where the diagonal elements are all equal to zero. Note that since the Kullback-Leibler divergence is not symmetric, in principle there is no guarantee that $\mathbf{T}^o$ is symmetric;

3) given a pair of distinct documents $\mathbf{d}_i$ and $\mathbf{d}_{i^*}$ $(i \neq i^*)$, under the null hypothesis of topic distribution homogeneity, combine the two documents into a synthetic

one, denoted as $\mathbf{d}_{i-i^*}$. Practically, since the data for the model is the DTM, simply add the corresponding rows together to create a new $(M-1) \times |V|$ matrix and replace it in the position corresponding to $\mathbf{d}_i$. Then, delete the row in the DTM corresponding to $\mathbf{d}_{i^*}$, obtaining DTM^{merged} [3]. Essentially, given that a document can be represented only by word frequencies, it is entirely coherent to merge them in this way if they share the same topic distribution. Then, fit again the LDA model on $\mathcal{C} \bigcup \{\mathbf{d}_{i-i^*}\}/\{\mathbf{d}_i, \mathbf{d}_{i^*}\}$ and obtain the new estimates $\widehat{\theta}_{merged} \in (M-1) \times G$ and $\widehat{\beta}_{merged} \in G \times |V|$;

4) once the estimates of the two matrices have been obtained, the Bootstrap procedure can be initialized in the following way. Sample with replacement from the pseudo-population $\mathbf{d}_{i-i^*}$ to generate $\mathbf{d}_i^{new}$ and $\mathbf{d}_{i^*}^{new}$. The sampling process is straightforward, in fact just for a new word $\mathbf{w}^{new}$:

$$\pi\left(\mathbf{w}^{new}|\mathbf{d}_{i-i^*}\right) = \iiint \pi\left(\mathbf{w}^{new}|\mathbf{Z}_{i-i^*}, \widehat{\theta}_{i-i^*}, \widehat{\beta}\right) \pi\left(\mathbf{Z}_{i-i^*}, \widehat{\theta}_{i-i^*}, \widehat{\beta}|\mathbf{d}_{i-i^*}\right) d\widehat{\beta}\ d\mathbf{Z}_{i-i^*}\ d\widehat{\theta}_{i-i^*}$$

multiply the row corresponding to $\mathbf{d}_{i-i^*}$ in the $\widehat{\theta}_{merged}$ matrix for $\widehat{\beta}_{merged}$ in order to obtain a distribution over the entire vocabulary. In essence, each column of the Topic-Word matrix can be regarded as the probability distribution of a word conditioned on one of the topics. Consequently, by computing the dot product between the topic distribution within the pseudo-population and the generic column of the Topic-Word matrix, it is possible to weight each word with the probability of observing a specific topic. Then draw from a Multinomial distribution $n_{\mathbf{d}_i}$ times for $\mathbf{d}_i$ and $n_{\mathbf{d}_{i^*}}$ times for $\mathbf{d}_{i^*}$ (Dirichlet-Multinomial distribution).

$$\tau = (\widehat{\theta}_{merged})_i \times \widehat{\beta}_{merged} \in 1 \times |V|$$
$$\mathbf{w}^{new}|\mathbf{d}_{i-i^*} \sim \mathfrak{Multinomial}_{|V|}(1, \tau)$$

An alternative and still valid possibility is to draw from a Multinomial distribution to select a topic and then, using the corresponding row to that topic of $\widehat{\beta}_{merged}$ as the probabilities over the vocabulary, draw a word from

[3] This approach makes the implicit assumption that the index associated with the position of the document $\mathbf{d}_i$ in the DTM is lower than the index of the position of the document $\mathbf{d}_{i^*}$.

a Multinomial again. Intuitively, a simulation from the posterior predictive distribution is made in order to entirely re-build the two documents.

Next, re-fit the LDA model considering the updated $M \times M$ Document Term matrix, where the two selected documents have been replaced with the newly sampled ones. From the new estimate of $\widehat{\theta}^{new}$, compute the test statistic for the two documents:

$$T^*_{ii^*} = \sum_{j=1}^{G} \widehat{\theta}^{new}_{\mathbf{d}_i j} \log \left(\frac{\widehat{\theta}^{new}_{\mathbf{d}_i j}}{\widehat{\theta}^{new}_{\mathbf{d}_{i^*} k}} \right)$$

5) repeat Step 3 for all the pairs of different documents and for each new pair, repeat Step 4 B times in order to have a sequence of Bootstrap values for the test statistic, i.e. $\{T^{new,1}_{ii^*}, \ldots, T^{new,B}_{ii^*}\}$;

6) compute the p-value $\widehat{p}_{ii^*}$ for each pair using (3.3) and the corresponding confidence intervals using (3.4). If the Bootstrap estimate of the p-value falls below the chosen confidence level α, it allows to reject the null hypothesis regarding document homogeneity. Conversely, if the Bootstrap estimate exceeds the confidence level, it is not possible to reject the null hypothesis.

Once the procedure has been presented, it is important to highlight the advantages and disadvantages associated with it. In Figure 3.3.1, you can see an example of the described procedure applied to two distinct documents. The conclusion drawn is that it is not possible to reject the null hypothesis of homogeneity because the p-value is very high. At the end of this process, an $M \times M$ symmetric matrix of p-values is obtained. Three important considerations can be made. First of all, it should be noted that while there is no guarantee of symmetry in the estimated matrix initially, after a certain number of Bootstrap iterations, the values tend to stabilize and become quite similar, allowing to assume the symmetry of the matrix. The opposite assumption would be counter-intuitive. Moreover, under this assumption, only $(M) \times (M-1)/2$ computations are necessary. This is because the diagonal elements are assumed to be equal to 1, reflecting that a document is certainly similar to itself [4].

[4] Although, for theoretical coherence, it should technically be $1 - \alpha$, but this notation is avoided for simplicity.

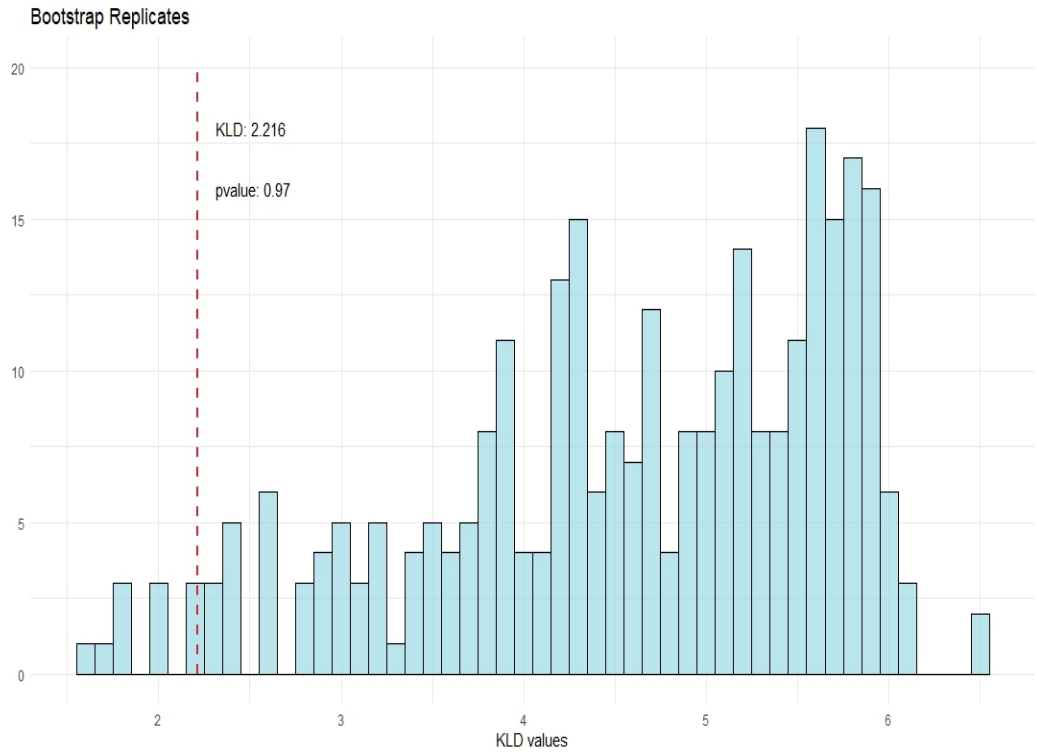

Figure 3.3.1. Bootstrap replicates of the homogeneity test with B = 300 applied to two distinct documents.

Another interesting problem concerns determining the optimal number of Bootstrap replicates required to obtain reliable p-values. According to [22, 44], a sufficiently large number of replicates is needed, typically at least 500 for the simplest statistics, which drastically increases for the estimation of a p-value. However, in this case, it is not feasible to employ a very high number of replicates due to the reliance of the method on subsequent repetitions of LDA, which can be computationally cumbersome [5]. Although a higher number of replicates would result in a better estimation of the p-value, a good trade-off between computational time and optimality is achieved with a number of iterations close to 500 (Figure 3.3.2). Of course, if significant computational resources are available, a higher number of replicates is recommended.

Lastly, it's crucial to emphasize that the outcome of LDA can vary if the

[5] On my computer, which has 3 available cores, it takes approximately 3 minutes and 20 seconds to compute a p-value using the distributed version of LDA with 1000 iterations [62].

distributions of parameters involved fail to converge, underscoring the necessity of achieving a situation of stationarity for the distributions. This issue is inherent to Bayesian problems requiring simulation and extends to the reliability of the sampling scheme utilized in the Bootstrap phase, potentially impacting the analysis. A minor issue arises from the possibility that the interpretation of a topic may vary across different runs. For instance, if topic 2 is initially associated with a specific phenomenon, it is conceivable that repeating the LDA process might influence the interpretation of topic 2 due to the lack of a strict labelling protocol.

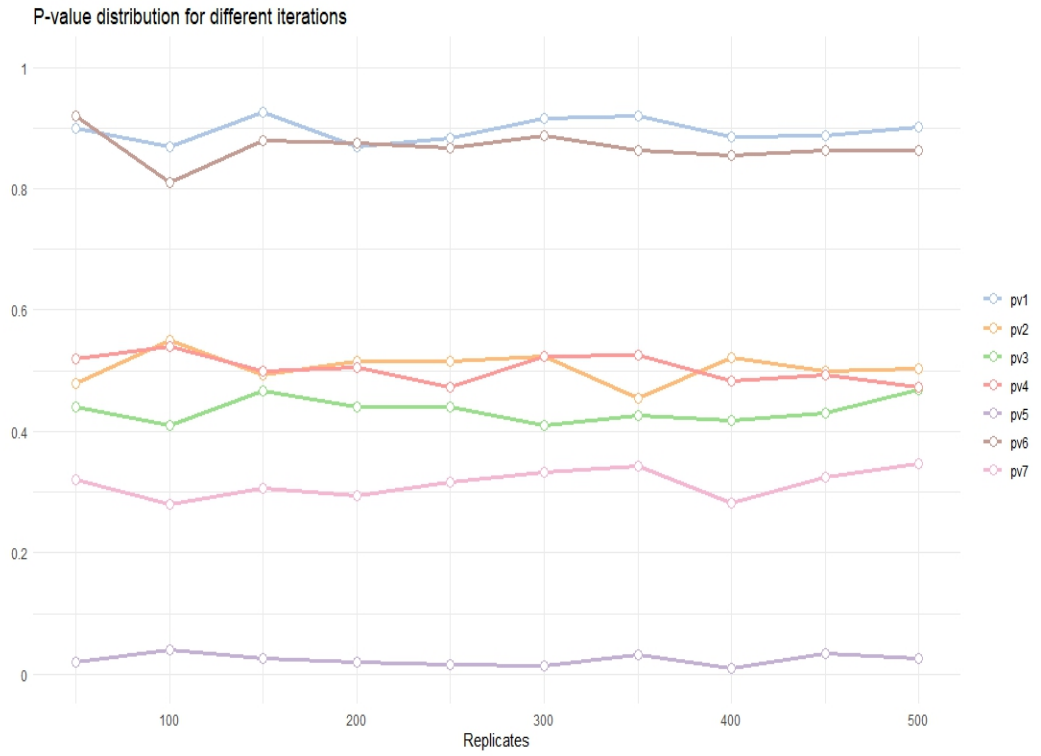

Figure 3.3.2. p-values for 7 distinct couples of documents for different values of the Bootstrap replicates.

Kullback-Leibler vs Bhattacharyya

One important research question concerns whether there exists a difference in the matrix of p-values when employing the Bhattacharyya distance (2.5) instead of the Kullback-Leibler divergence (2.4). The simplistic idea is that despite the significant differences between these two similarity measures, the resulting p-values

should be largely similar.

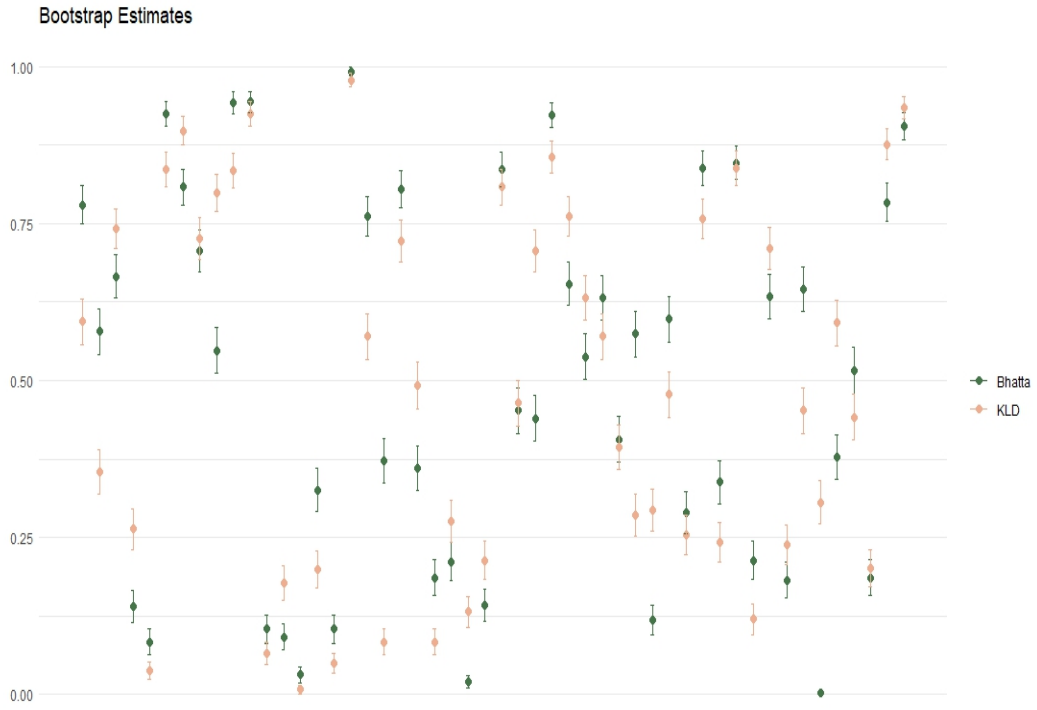

Figure 3.3.3. Bootstrap estimates of p-values and confidence intervals at level 0.9 of 50 distinct couple of documents using Kullback-Leibler divergence and Bhattacharya distance with 500 iterations.

From Figure 3.3.3, it seems reasonable to state that in the majority of the cases there is a substantial concordance between the Bootstrap p-values estimates obtained using as the similarity measure the Bhattacharya distance rather than the Kullback-Leibler. However, the two confidence intervals just in a few cases tend to overlap, so the choice of the similarity measure can have an small influence on the values attained by the matrix of p-values.

Although both metrics lack the triangle inequality property, the Bhattacharyya distance possesses symmetry, which can be particularly useful in establishing the symmetry of the p-value matrix. Until now, this assumption has been made to streamline computations. As pointed out in Subsection 2.2.3, these two measures are related and it is possible to demonstrate that the Kullback-Leibler divergence

is generally greater than the Bhattacharyya distance. More formally, it can be stated that the Bhattacharyya distance tends to be at least half of the value of the Kullback-Leibler divergence for the same probability distributions:

$$B(p,q) = -\log\left(\sum_x \left(p(x) * q(x)\right)^{\frac{1}{2}}\right) = -\log\left(\sum_x \left(p(x) * q(x) * \frac{p(x)^2}{p(x)^2}\right)^{\frac{1}{2}}\right) =$$

$$= -\log\left(\sum_x p(x)\left(\frac{q(x)}{p(x)}\right)^{\frac{1}{2}}\right) = -\log\left(\mathbb{E}\left[\left(\frac{q(x)}{p(x)}\right)^{\frac{1}{2}}\right]\right) \leq -\mathbb{E}\left[\log\left(\frac{q(x)}{p(x)}\right)^{\frac{1}{2}}\right]$$

$$= -\sum_x p(x) * \log\left(\frac{q(x)}{p(x)}\right)^{\frac{1}{2}} = \frac{1}{2}KL(p,q)$$

3.4 New Document Clustering proposals

After introducing the homogeneity between topic distributions Bootstrap test, it becomes interesting to explore how its p-value can be applied in the context of document clustering. Essentially, this estimated p-value can be interpreted as a measure of similarity between any two documents, where lower values indicate greater dissimilarity and higher values suggest stronger similarity. This is inherently linked to the definition of the p-value, as a low value indicates the rejection of the null hypothesis, while a high value implies the non-rejection of the null hypothesis [6].

	DOC 1	**DOC 2**	**DOC 3**	**DOC 4**
DOC 1	0	0.256	0.031	0.742
DOC 2	0.256	0	0.123	0.891
DOC 3	0.031	0.123	0	0.06
DOC 4	0.742	0.891	0.06	0

Table 3.4.1. Example of a dissimilarity matrix build using the estimated p-value ($D(\mathbf{d}_i, \mathbf{d}_{i^*}) = 1 - \widehat{p}_{ii^*}$) linked with the hypothesis test of homogeneity between topic distributions of 4 documents in the BoAs corpus.

[6] DOC1:"Adaptive treatment allocation and selection in multi-arm clinical trials: A Bayesian perspective ", DOC2: "Variable selection in distributed sparse regression under memory constraints", DOC3: "Double machine learning for (weighted) dynamic treatment effects", DOC4: "Group selection with Bayesian high dimensional modeling"

A straightforward approach would be to define the dissimilarity [7] between two documents $\mathbf{d}_i$ and $\mathbf{d}_{i^*}$ as $D(\mathbf{d}_i, \mathbf{d}_{i^*}) = 1 - \widehat{p}_{ii^*}$, which takes values in the range $[0, 1]$. By applying this calculation to every pair of documents, a dissimilarity matrix can be constructed. From Table 3.4.1, it is possible to find an example of a small dissimilarity matrix build using 600 iterations of the Bootstrap test.

Subsequently, three different approaches can be delineated, the first two of which utilise the p-value linked to the hypothesis test exclusively at an initial stage. The first one is an agglomerative hierarchical clustering procedure (which can employs any type of linkage method, as described in Subsection 2.3.1), while the other is an application of the NEFRC algorithm (Subsection 2.3.3) made on the previously defined dissimilarity matrix. Nevertheless, although these two approaches represent a novel contribution to the field of document clustering, they primarily address the inferential problem at an initial stage. Consequently, there is no strong assurance that clusters formed at subsequent stages of the clustering process maintain statistical equivalence based on the hypothesis test. Furthermore, it is crucial to emphasise that the two clustering methods are based one the estimated dissimilarity matrix. The precision of the estimation of each p-value relies on the number of replicates chosen in the Bootstrap phase. This implies that the complexity of the new algorithms is dependent on two factors. Firstly, the complexity of creating the dissimilarity matrix, which is typically high (recall Section 3.3 for a better explanation) and secondly, the complexity of the hierarchical/relational clustering procedure.

To tackle the first issue, following the ideas outlined in [25], a procedure can be implemented to overcome the challenge of maintaining statistical equivalence between partitions. Also in this case, an agglomerative approach can be defined. Begin with the trivial partition of all singletons and set a significance level α. Typically, the choice of the significance level α depends on the desired type I error probability, often ranging between 0.01 to 0.1. Compute the symmetric matrix of p-values using the estimated p-values via the Bootstrap procedure to test the homogeneity of topic distributions among documents described in Section 3.3. Then connect the two

[7] It is not, in fact, a distance because it fails to fulfil all the necessary properties.

documents with the highest p-value that exceeds the threshold α. If no p-value is greater or equal than the significance level, it means that the clusters are statistically distinct at that specific significance level and the clustering process can be concluded.

In further detail, the algorithm can be outlined as follows:

Step 1 Given a corpus $\mathcal{C}$, a significance level α, a fixed number G of topics, compute the matrix of p-values using the method described in Section 3.3;

Step 2 identify the pair of documents with the highest p-value score. If this p-value exceeds the significance level α, link these two documents together to form a cluster. To define the cluster centroid, merge the corresponding rows in the DTM;

Step 3 compute the p-values between the newly formed cluster and the remaining individual documents. Again, identify the highest p-value in this updated similarity matrix. If this p-value is greater than α, merge the clusters associated with this highest p-value. Otherwise, terminate the clustering procedure. It is possible to compute the within sum of squares using the formula 2.6;

Step 4 repeat **Step 3** iteratively until no p-value exceeds the significance level α, indicating that all remaining clusters are statistically distinct at this threshold.

There are several important considerations to take into account with this procedure:

- it is important to note that due to the reliance of LDA on the Multinomial distribution, cluster centroids are obtained by summing the word frequencies within the documents, not by computing an average value. From an interpretative perspective, since topics are essentially defined as mixtures of words, if two documents are statistically equivalent, they should share a significant number of words. Therefore, by employing the summation of the rows of the DTM, the common topic should be more apparent. On the other hand, this can introduce biases when comparing documents of different lengths, which highlights a potential challenge in maintaining document comparability during

the clustering process. An alternative approach would be to select a single document as the representative of the cluster (medoid approach). However, this type of approach presents issues in the selection procedure of the medoid for each cluster due to the fact the choice would be randomly made;

- the computational cost can be significant, particularly in the worst-case scenario. The initial computation involves generating the matrix of p-values for all document pairs, which requires $(M(M-1))/2$ iterations. Subsequent iterations involve computing p-values between clusters, which requires $(M-1)(M-2)/2$ iterations in the worst case. This cumulative computation can be computationally cumbersome for large datasets. Additionally, as the previous method, one must also consider the number of Bootstrap replicates in the calculation of p-values, which can further increase the computational costs. The estimation of p-values is particularly critical, as any loss in precision could dramatically impact the clustering method;

- unlike methods that require selecting an optimal number of clusters, this approach employs an objective stopping criterion based on the significance level. However, at each iteration it is possible to save the current clustering partition providing insight into the progression of clustering. It's also intriguing to note that this method can be seen as an instance of an agglomerative hierarchical approach. Additionally, with each iteration, the corresponding merging p-value should gradually decreases;

- the choice of confidence level impacts both the size of clusters and the final partition obtained. Increasing α typically leads to a high number of small clusters, as higher confidence levels make it less likely for documents or clusters to merge. Conversely, lowering α can result in a lower number of clusters with a greater size;

- the number of topics is maintained constant throughout each iteration of the clustering procedure, which may be perceived as a somewhat simplistic approach, not from an interpretative point of view but from a methodological perspective.

3.5 Comparison

For the subsequent analysis, two datasets have been selected. The first dataset consists of a collection of Books of Abstracts from the CMStatistics conference, and it is being used for clustering purposes for the first time. Unfortunately, this dataset is not publicly available unless PDF scraping is conducted, but it can be obtained from the author of this thesis upon request. It encompasses all published papers presented at the aforementioned conference from 2008 to 2020. I selected this dataset because it lacks labels and relates to a domain that I can readily interpret. Similarly to the previous situation, I opted to focus on a subset of the original dataset, specifically considering only 100 documents from the conference's last two years chosen completely at random. After applying basic preprocessing to this dataset (detailed in Subsection 2.2.2), I chose to adopt a lemmatizer from the NLTK library. Subsequently, I removed words that appeared in less than 4% or more than 95% of the corpus, leading to a vocabulary of 474 words was obtained (with 91% sparsity).

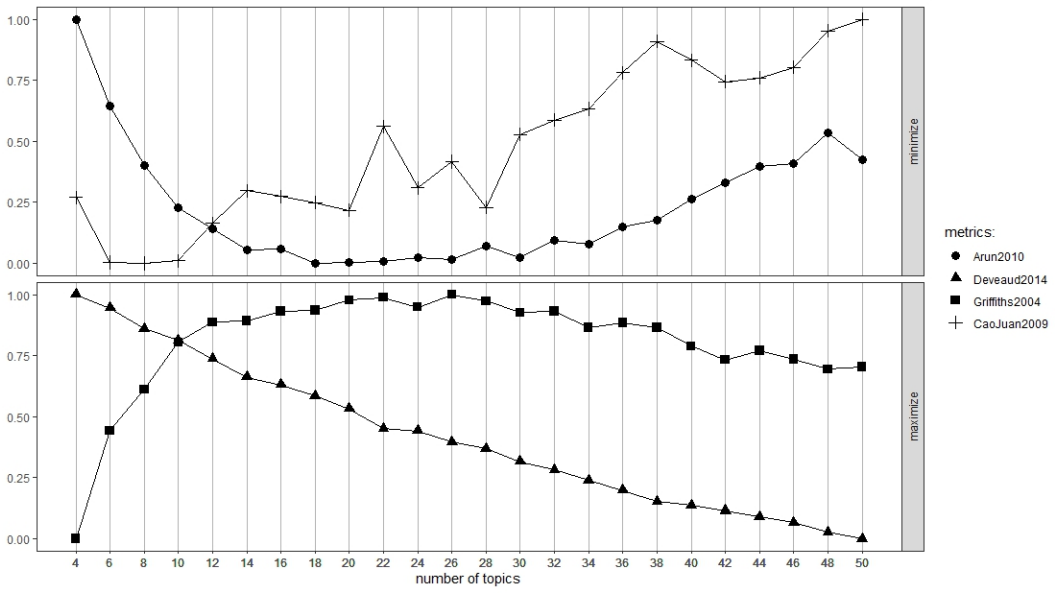

Figure 3.5.1. Coherence scores for the 20 Newsgroups corpus.

The second dataset is the well-known 20 Newsgroups (benchmark dataset), made of roughly 20,000 documents categorized into 20 different groups covering

topics ranging from science to political discourse and motorcycles. Following the methodology outlined in [46, 60, 14], a subset of the original dataset has been selected. Specifically, I chose to focus on the macro-area associated with science. This subset consists of four distinct labels: "sci.space," "sci.electronics," "sci.crypt," and "sci.med." I opted to work with a reduced number of documents, selecting 100 documents while preserving the proportion of documents within the different classes. After applying the same preprocessing steps as before, a final vocabulary of approximately 421 words (with 90% sparsity). It is noteworthy to mention that in the two preprocessing phases, unigrams, bigrams and trigrams have been considered.

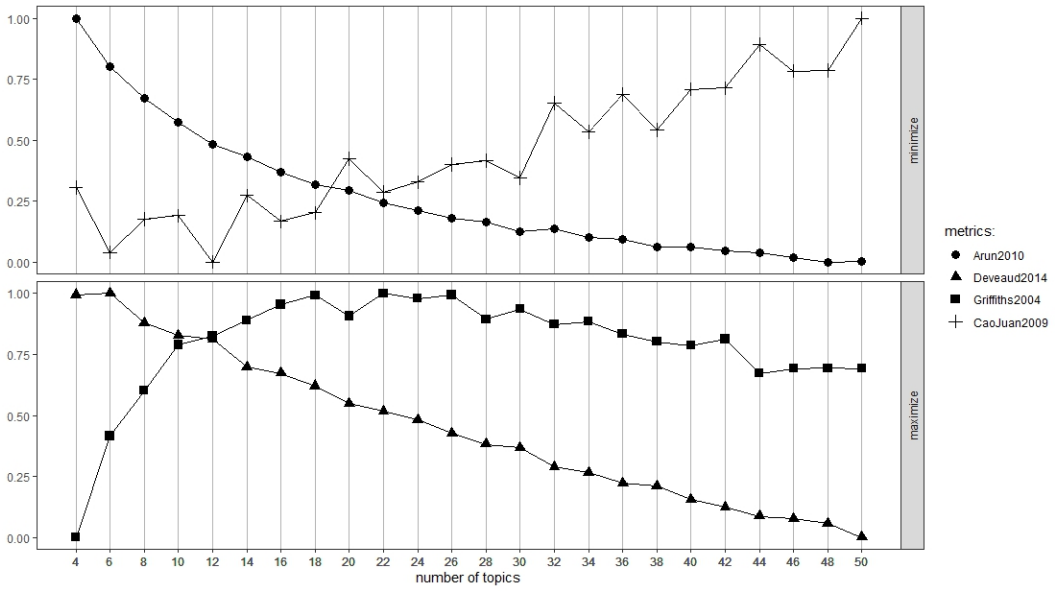

Figure 3.5.2. Coherence scores for the BoAs corpus.

After the preprocessing step, coherence measures were employed to find the optimal number of topics for the two datasets. From Figure 3.5.2, it is evident that for the BoAs corpus, the optimal number of topics is approximately 12, while for the other dataset (Figure 3.5.1) it is approximately 10. The idea is that the optimal number of topics lies between the intersection points of the two plots, as pointed out by Nikita Moor (creator of the package). Furthermore, in both situations, the chosen topics tend to represent slightly different aspects within a specific area. For example, in the 20 Newsgroups dataset, one of the topics discusses encryption in general, while another explicitly talks about internet privacy.

In conclusion, it is noteworthy to mention that it is unlikely that one would encounter a corpus of this size in a real-life scenario. However, this choice was made for computational and time-related reasons.

3.5.1 Notation and Results

Up until now, the three clustering methods have been presented from a theoretical point of view. A basic question could arise:

> **How do these new clustering methods perform compared to the ones commonly used in the literature?**

Before delving into the application of all the methods, it is mandatory to establish a common notation for all the different clustering techniques that will be applied for the experimental part [8].

LUC) latent unsupervised clustering, namely the third method described in Section 3.4 with α equal to 0.05 and G topics which are different in the two cases. The number of iterations have been fixed to 600;

HAC-P) hierarchical agglomerative clustering applied on the estimated matrix of p-values obtained with the application of the first method described in Section 3.4 with 600 iterations using WPGMA as linkage method;

HAC-C) hierarchical agglomerative clustering applied on the reduced matrix using cosine similarity and the complete linkage method;

KM-E) K-Means using Euclidean distance with K-Means ++ initialization applied on the reduced matrix;

KM-C) K-Means using cosine similarity applied on the reduced matrix;

FKM) Fuzzy K-Means with fuzziness parameter equal to 2 applied on the reduced matrix;

[8] It is important to state that every time I talk about the *reduced* matrix I am referring to the matrix of the score obtained applying PCA on the original DTM.

BKM) Bisecting K-Means with the bisecting criterion based on the greatest internal variability applied on the reduced matrix;

GMM) Gaussian Mixture Model with K components applied on the reduced matrix. The EM algorithm has been initialized using the partition obtained via a agglomerative hierarchical approach;

SC) spectral clustering with K clusters using the string kernel (length 6, λ equal to 1.2 and Boundrage kernel) as a similarity measure applied on the original corpus $\mathcal{C}$ without any preprocessing;

NEFRC-P) Non Euclidean Fuzzy Relational Clustering applied on the dissimilarity matrix based on the estimated matrix of p-values (second method in Section 3.4).

In the subsequent phase, the focus shifts to the application aspect the two small datasets, while the method's performance is analyzed in comparison with other methods. It's important to note that due to constraints in my computational resources throughout this thesis, all simulations were conducted using **Terastat 2.0** [10]. I utilized 16 cores, each with 2GB of RAM.

Before approaching the clustering procedure, it is crucial to emphasize a key point. Despite employing 600 iterations for the Bootstrapping method, the p-values matrix exhibits a certain degree of sparsity, approximately 30%. Consequently, the dissimilarity matrix maintains a similar proportion of 1 values (as depicted in Figure 3.5.3) which can have an influence on the subsequent clustering procedures.

From Table 3.5.1, it seems that no single method emerges as universally superior across all possible partitions. However, SC proves to be the most effective in half of the scenarios. Among the methods developed in Subsection 3.4, they tend to yield similar scores, albeit with the HAC-P showing slightly better performance. It is noteworthy to mention that LUC didn't stop prematurely due to the fact that the greater p-value was greater than the confidence level α, but rather merged all units into one cluster.

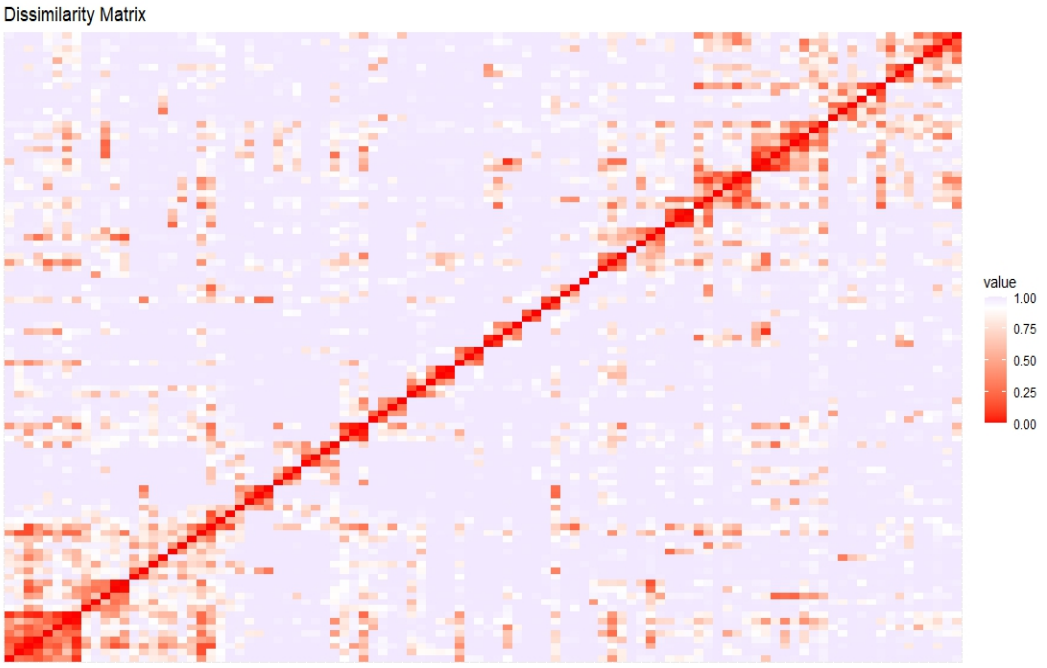

Figure 3.5.3. Dissimilarity matrix obtained using $D(\mathbf{d}_i, \mathbf{d}_{i^*}) = 1 - \widehat{p}_{ii^*}$ on the BoAs dataset.

Clusters	LUC	HAC-P	HAC-C	KM-E	KM-C	FKM	BKM	GMM	SC	NEFRC-P
2	0.375	0.372	0.369	0.382	0.378	0.379	0.3795	0.373	**0.385**	0.371
4	0.399	0.410	0.410	**0.429**	0.399	0.409	0.411	0.408	0.419	0.392
6	0.417	0.442	0.429	**0.449**	0.424	0.409	0.4367	0.434	0.445	0.412
8	0.440	0.459	0.448	0.478	0.453	0.433	0.460	0.468	**0.491**	0.431
10	0.468	0.478	0.462	0.4896	0.477	0.436	0.480	0.483	**0.515**	0.4485

Table 3.5.1. Overall similarity scores for different clustering partitions and for all the different methods applied to the BoAs corpus computed on the original DTM.

In conclusion, it seems that across all methods, the optimal solution is achieved with four clusters. This is because, when more than four clusters are used, the relative improvement in internal cohesion is minimal and having more clusters could compromise interpretability. Figure 3.5.4 illustrates the different assignments of units to the four clusters in the GMM and LUC approaches.

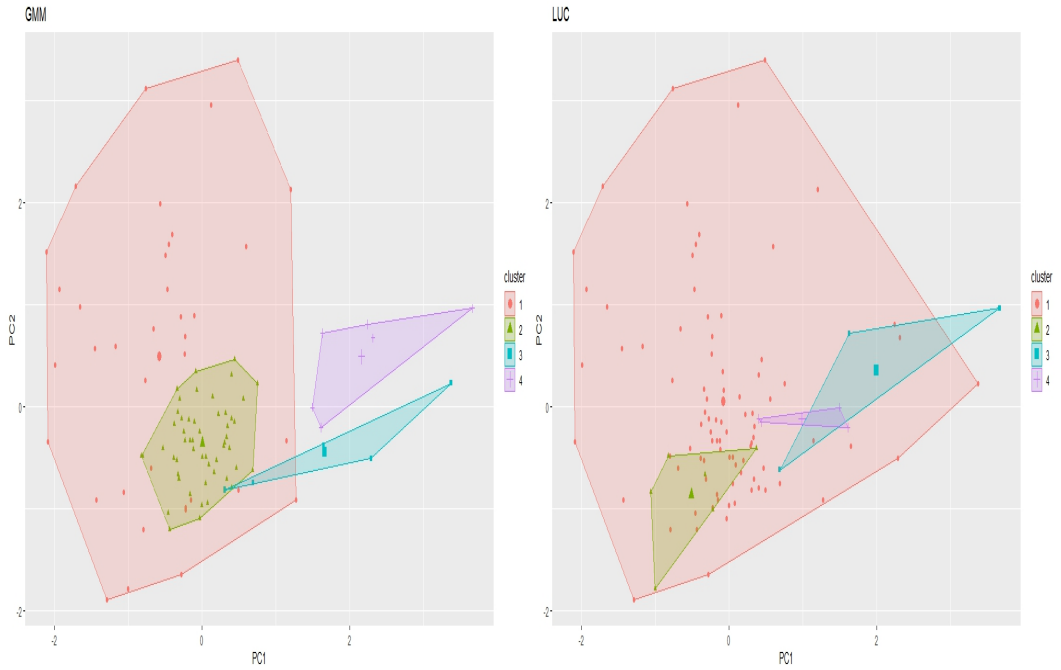

Figure 3.5.4. Assignments of units to clusters considering the two component of PCA using GMM (left) and LUC (right) on the BoAs corpus.

Upon closer examination, particularly focusing on the partition in four clusters obtained using the three new methods, it becomes apparent that within each cluster, there exist sub-clusters of documents that share the same topic. This observation suggests that despite the overall similarity not being optimal, the methods still manage to generate interpretable results. As illustrated in Figure 3.5.5, for instance considering the partition with four clusters, it becomes evident that the prevalence of topics within clusters is more or less homogeneous for the NEFRC-P method, the situation is somewhat less homogeneous for the HAC-P method, while in the other case it is heterogeneous just for three clusters. It is noteworthy that for the LUC method, topic 10 ("Functional Analysis") stands out as it is predominantly discussed in cluster 4. For cluster 2 and 3, it is possible to observe that they share more less the same topics. To conclude, the labels associated to the other topics are: 1 ("Undefined"), 3 ("Undefined"), 4 ("Computational Statistics"), 5 ("Bayesian"), 6 ("Regression"), 7("High-Dimensional"), 8 ("Asymptotics"), 9("Spatial data"), 11("Causal Inference") and 12 ("Clustering").

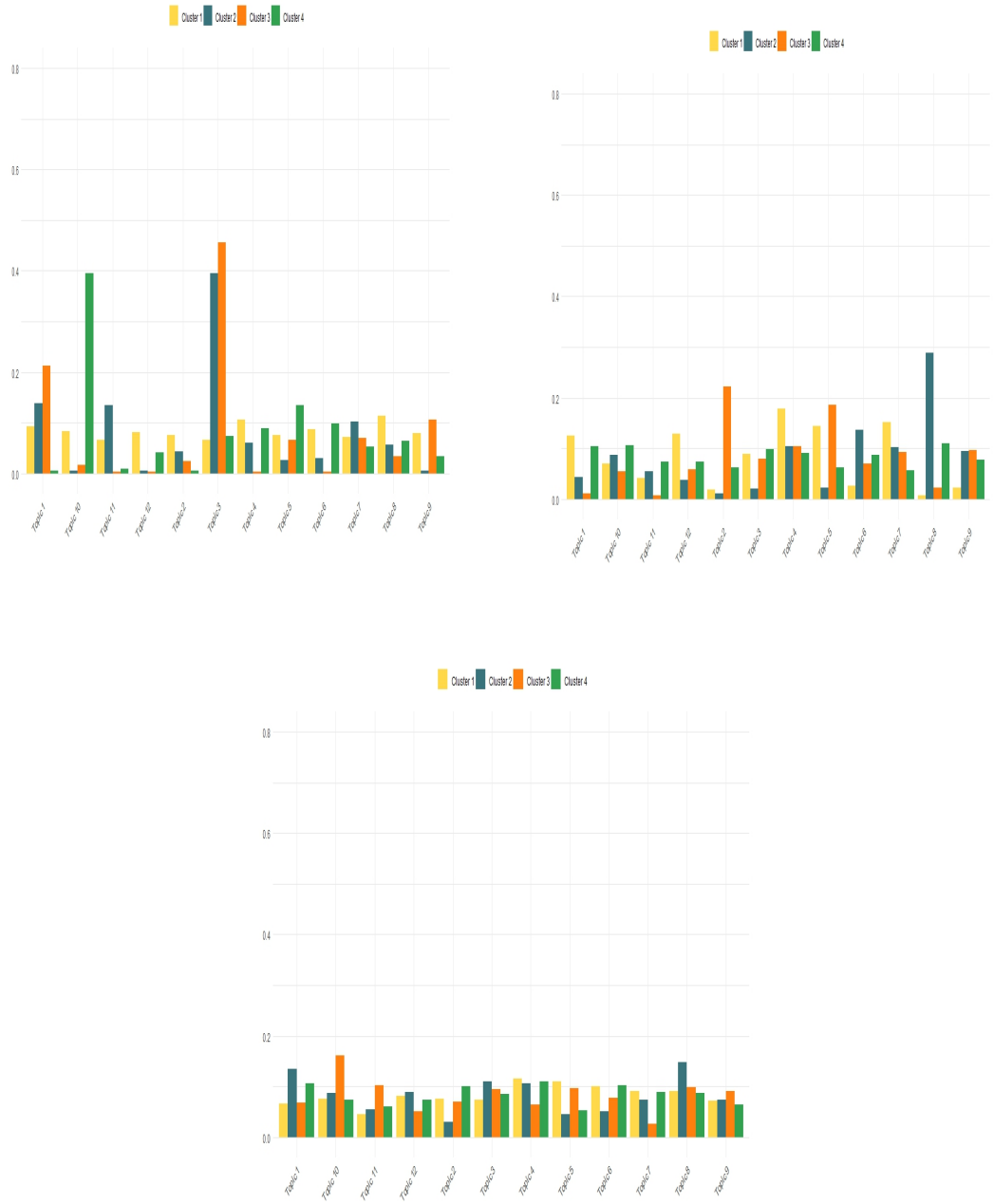

Figure 3.5.5. Significance degree of the 12 topics (3.1) inside the partition with 4 clusters obtained using the LUC method (top left), HAC-P (top right) and NEFRC-P (bottom center) referring to the BoAs corpus.

Moving forward to the other corpus, despite employing the same number of Bootstrap replicates, it is evident from Figure 3.5.6 that the degree of sparsity in the p-value matrix is lower (approximately 18%). Additionally, it seems that this

time, relationships between documents have been more accurately captured by the p-values. Lastly, I want to emphasize that even though the access to the information about the labels of documents is possessed in this scenario, the primary aim remains clustering rather than classification, due to the fact they are distinct tasks with different goals.

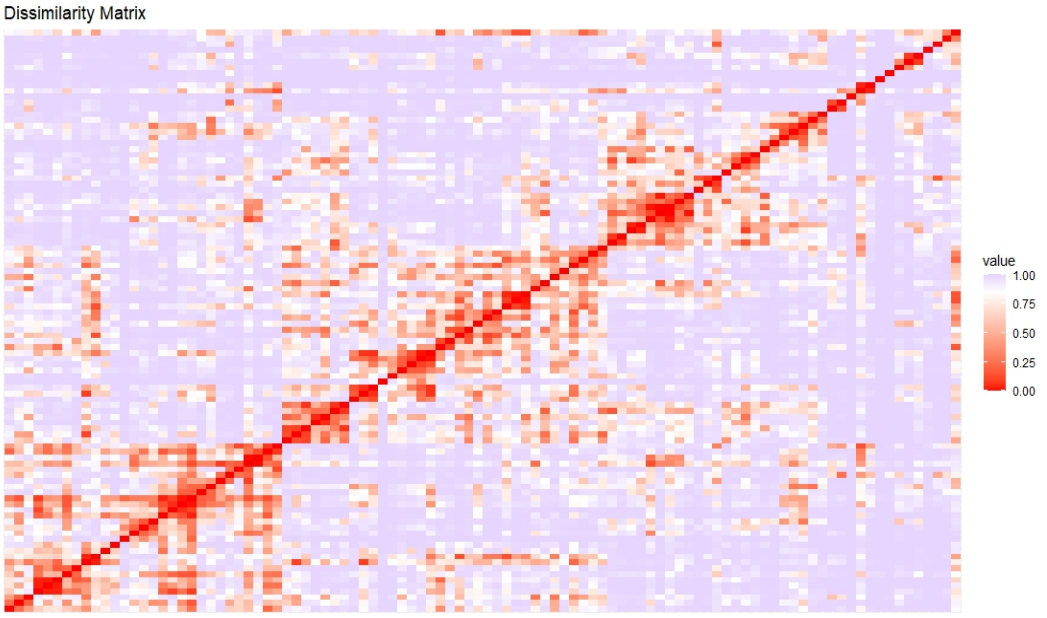

Figure 3.5.6. Dissimilarity matrix obtained using $D(\mathbf{d}_i, \mathbf{d}_{i^*}) = 1 - \widehat{p}_{ii^*}$ on the 20 Newsgroups corpus.

From Table 3.5.2, it can be observed the behavior of different models for the partition with four clusters. From a classification perspective, it is evident that none of the chosen models perform exceptionally well. Considering all indices, the HAC-C method, the KM-C method, and the SC method emerge as the most effective.

Index	LUC	HAC-P	HAC-C	KM-E	KM-C	FKM	BKM	GMM	SC	NEFRC-P
Entropy	0.335	0.329	**0.272**	0.338	0.337	0.356	0.336	0.327	0.3126	0.330
$F-measure$	0.117	0.201	0.241	0.128	0.242	0.206	0.168	0.243	0.2458	**0.260**
Cosine	0.365	0.3436	0.348	0.388	**0.437**	0.397	0.406	0.405	0.422	0.3710

Table 3.5.2. Entropy score, $F-measure$ and Cosine similarity for all the different methods applied to the reduced 20 Newsgroups with 4 labels.

From an interpretative perspective, Figure 3.5.7 illustrates the following situation. For the NEFRC-P method, no particular topics dominate any cluster, as the proportions within each topic are fairly similar. In contrast, the other two clustering methods reveal a more distinct prevalence of specific topics within clusters. Specifically, for the LUC method, Cluster 1 is associated with Topic 3 ("Electronics-Know How"), Cluster 2 with Topic 5 ("Medicine"), Cluster 3 with Topic 1 ("Crypto-Government"), and Cluster 4 with Topic 10 ("Space"). For the other method, the distinctions are less pronounced, but it is still possible to identify a connection between Cluster 3 and Topic 10, as well as Cluster 1 and Topic 3.

Figure 3.5.7. Significance degree of the 10 topics (3.1) inside the partition with 4 clusters obtained using the LUC method (top left), HAC-P (top right) and NEFRC-P (bottom center) referred to the 20 Newsgroups corpus.

The labels associated to other topics are: 2 ("Undefined"), 4 ("Electronics-Chips"), 6 ("Electronics-NNTP"), 7 ("Electronics-Internet Privacy"), 8 ("Undefined") and 9 ("Space-Company").

In conclusion, based on the results of these two small experiments, several observations can be made. Firstly, despite the higher computational cost, it is clear that the newly defined clustering methods tend to perform on average as well as other methods, without exhibiting exceptional behavior. Secondly, these new methods rely on the estimation of the dissimilarity matrix, which depends on the number of Bootstrap replicates. Therefore, the main disadvantage is the extensive computational time required for the computation of all the p-values, making the problem not scalable with basic computers. Lastly, from an interpretive point of view, the LUC method appears to be the most effective at capturing documents that share the same topic among the newly developed clustering schemes.

To make this method scalable and efficient, progress must be made in reducing computational time, allowing for more Bootstrap replicates. Methodologically, developing a new method for selecting centroids could be beneficial. Future work should focus on addressing these two main limitations.

Chapter 4

Application

In Chapter 3 a comparison between the newly developed clustering techniques and existing ones was conducted in order to understand the behaviour of the different clustering techniques. In this Chapter, the focus shifts to a real case scenario, specifically the BoAs dataset, with several questions in mind:

- **What are the main topics of interest for three different years?**
- **How have the topics within the papers presented at the conference evolved over thirteen years?**
- **Is it possible to identify groups of similar documents within each corpus? Is it possible to interpret those clusters?**

To address these questions, I have decided to focus on the years 2008, 2014, and 2020. An immediate observation is the significant increase in the number of papers submitted and presented at the conference over time: from just 92 papers in 2008 to 670 in 2014, and 692 in 2020. This is linked to the growing popularity of the conference.

Following the preprocessing step which directly employed stemming (as opposed to lemmatization used previously), the vocabularies of the corpora were reduced to 354 words with 89% sparsity for 2008, 763 words with 93% sparsity for 2014, and 895 words with 92% sparsity for 2020. It is important to note that each corpus consists of abstracts, which are generally relatively short texts.

As in the previous analysis, determining the optimal number of topics for each corpus is crucial. From Figure 4.0.1, it is evident that the optimal number of topics for 2008 is approximately 8, for 2014 it is in the range $18 - 22$, and for 2020 it is approximately between the same range. For the last two years, I have tried different combinations of topics, with the optimal number emerging as 20 in both cases from an interpretative point of view.

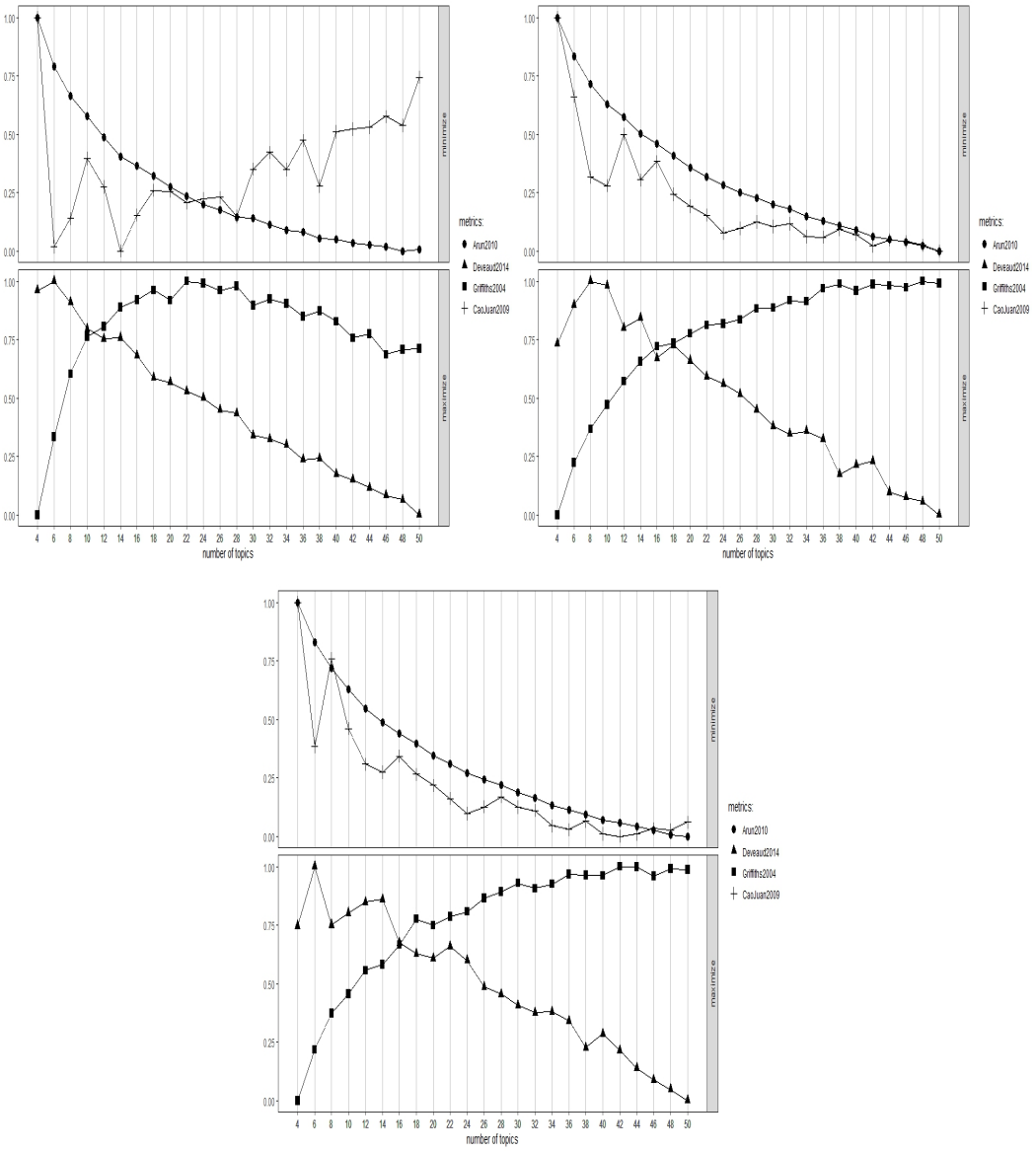

Figure 4.0.1. Coherence scores for the year 2008 (top left), 2014 (top right) and 2020 (bottom center).

As a side remark, it is noteworthy that the assumption of independence among topics from different years was made for simplicity. From a practical standpoint, it is evident that the topics discussed in one year can influence those in subsequent years. To mitigate this effect, I have chosen distant years for analysis. However, in general, a generalized form of LDA, as discussed in [8], can be employed. Lastly, due to the nature of the corpora, it is likely that some topics may overlap.

Once the optimal number of topics has been determined, it is possible to give an answer to the first postulated question. A significant difference is the substantial increase in the number of documents (and subsequently topics) between 2008 and the other two years. As shown in Table 4.0.1, the topics in 2014 and 2020 are very similar, indicating that the conference consistently includes papers on these topics. However, it is challenging to directly compare them with the topics from 2008. The topics in 2008 can be seen as laying the foundation for the conference's development, as many of the topics discussed in 2008 continue to be relevant in subsequent years.

2008	**2014**	**2020**
Asymptotics	*Applied Statistical Analysis*	*Asymptotics*
Bootstrap	*Asymptotics*	*Bayesian Statistics*
Clustering	*Bayesian Statistics*	*Clustering*
Fuzzy RV	*Clustering*	*Causal Inference*
Neural Networks	*Dependency*	*Dependency*
Regression Model	*Financial Market*	*Extreme Value Theory*
Sample Survey	*Functional Analysis*	*Financial Market*
Time Series	*High-Dimensional*	*Functional Analysis*
	Hypothesis Test	*High-Dimensional*
	Mixture Model	*Hypothesis Test*
	MLE	*Machine learning*
	Multivariate Statistics	*Microbiome*
	Optimization	*Network Analysis*
	Principal Component Analysis	*Optimization*
	Regression Model	*Regression Model*
	Robust Methods	*Stochastic Processes*
	Stochastic Processes	*Survey Analysis*
	Survival Analysis	*Survival Analysis*
	Time Series	*Time Series*
	Variable-Selection	*Variable-Selection*

Table 4.0.1. Topics' labels for the three different years

The interesting part is that each year also features unique topics: for example, in 2008 there were papers on fuzzy random variables while in 2014 there were papers on robust statistical methods and in 2020 there were more papers on stochastic processes. Table 4.0.2 shows the papers most similar to the specific topics for each year.

Title	**Topic**	**Year**
A possible extension of upper and lower probabilities to the case of fuzzy random variables	*Fuzzy RV*	2008
General framework for the rotation of units in repeated survey sampling	*Sample Survey*	2008
Robust estimation of multivariate scatter with fixed center	*Robust Methods*	2014
E553: Mixtures of skewed distributions with hypercube contours	*Mixture Models*	2014
E0544: A Bernstein-von Mises theorem for stochastic PDEs	*Stochastic Processes*	2020
E1061: Climate extreme event attribution using multivariate peaks-over-thresholds modeling and counterfactual theory	*Extreme Value Theory*	2020

Table 4.0.2. Documents more similar to the specific topic for each year.

Moving to the last question, although it would have been interesting to use the three new clustering methods developed during this thesis, agglomerative hierarchical clustering with cosine dissimilarity (computed as the complement to 1 of the cosine similarity) was employed instead on the normalized version of the TF-IDF matrix. This choice was made for two main reasons: first, agglomerative hierarchical clustering is widely used in this context and proved to be one of the most effective methods in the two previous small examples; second, the computational time required for the three new clustering methods is too high for this volume of documents.

Figure 4.0.2 shows overall similarity scores for different numbers of clusters in the three years. According to the plot, the optimal number of clusters is 5 for 2008, while for both 2014 and 2020, the optimal number of clusters is 10, which is relatively high. As in the previous application, it is noteworthy that there is a monotonic relationship between the number of clusters and the overall similarity score.

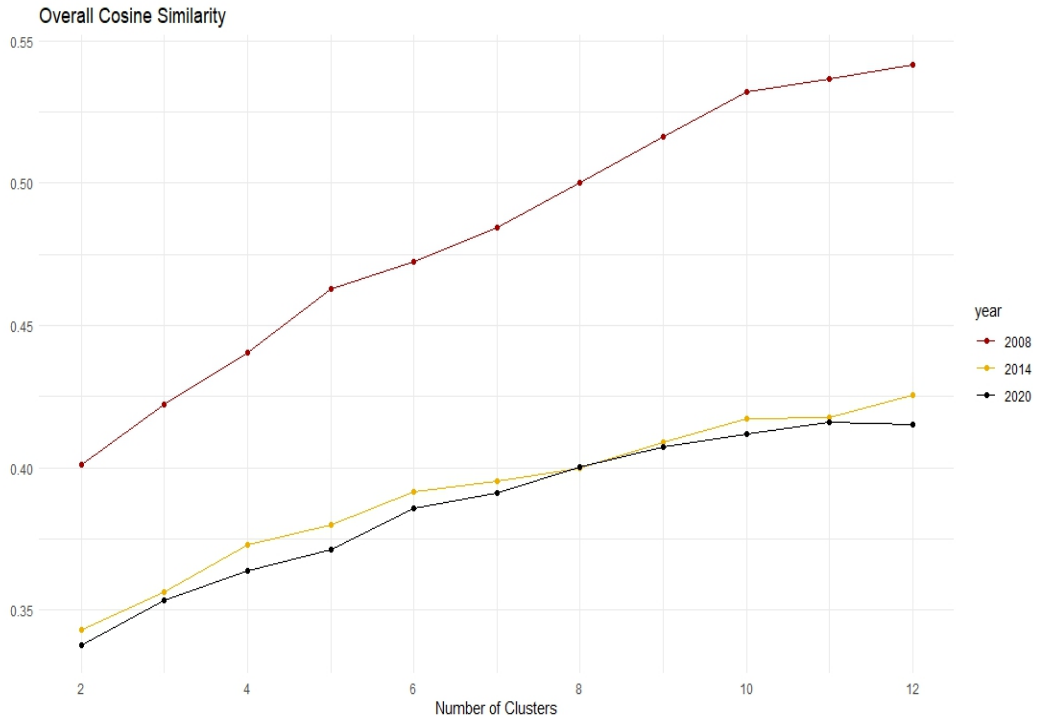

Figure 4.0.2. Overall Cosine Similarity for the three different years using HAC-C.

Moving forward to the analysis of the clustering partition for the year 2008, Figure 4.0.3 and Figure 4.0.4 show the most frequent words within clusters and the prevalence of topics within clusters. The most frequent words in each cluster can be interpreted as representing the average document in that cluster (the centroid). From these plots, it is evident that "Time series" is predominantly discussed in clusters 4 and 5, while "Dependency" and "Fuzzy RV" are related to cluster 1. Additionally, it appears that cluster 3 is associated with "Clustering" and cluster 2 with "Regression Model". It is noteworthy to mention that cluster 2 is the one with the greatest size and it tends to incorporate documents which refer to different documents, so it is not a "specific" cluster.

Table 4.0.3 provides the papers most similar to each cluster centroid according to the cosine similarity. In particular, it can be claimed that it gives the same representation as the previous plot. As a side remark, note that the cosine similarity between a document and its corresponding cluster centroid tends to be higher when the cluster size is smaller.

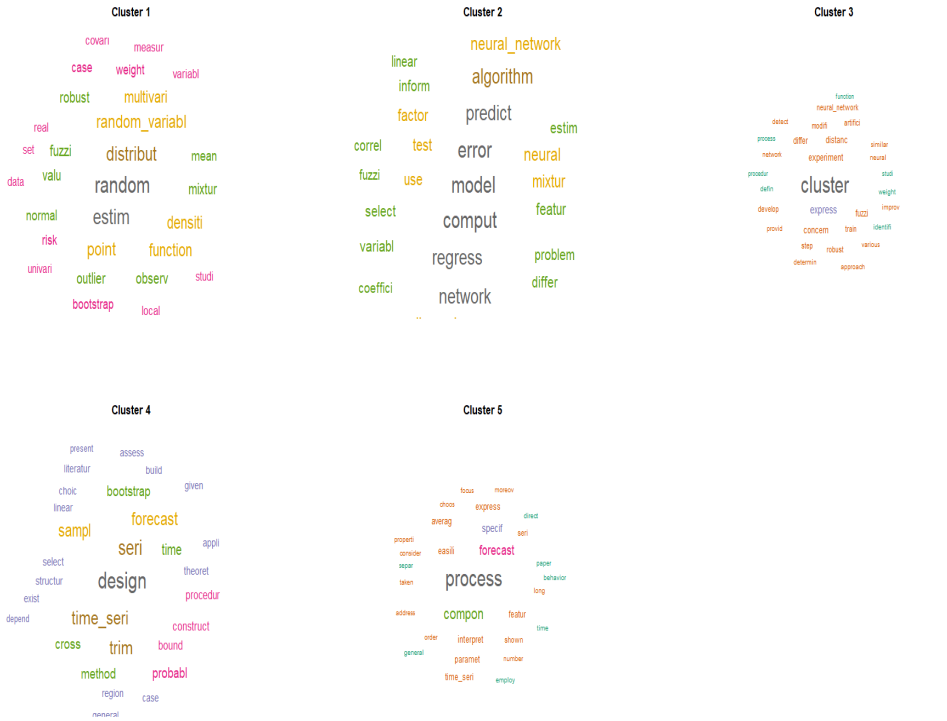

Figure 4.0.3. Wordclouds of the most frequent words per cluster for 2008

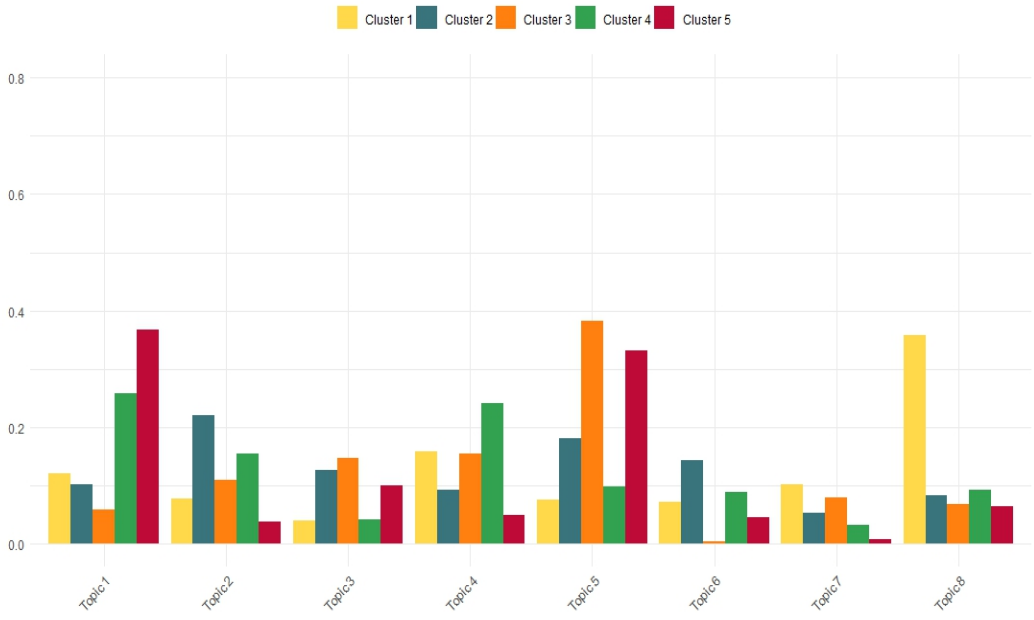

Figure 4.0.4. Significance degree of topics inside clusters for 2008.

Title	Similarity	Cluster	Cluster size
Fuzzy techniques in the analysis of distributions of real random variables	0.474	1	28
The estimation of prediction error for neural networks: a simulation study	0.485	2	46
Probabilistic noise clustering as M-estimators	0.774	3	3
Variable selection for time series forecasting using the group-wise LARS algorithm	0.5043	4	12
Aggregation of vector ARMA processes: some further results	0.8286	5	3

Table 4.0.3. Documents with the highest cosine similarity with respect to the cluster centroid for 2008.

Starting from 2014, a conjoint analysis of Figure 4.0.5 and Figure 4.0.6 allows for various observations. Firstly, it is evident that there are many "specific" clusters each linked to one or at most two topics, except for cluster 1 (the largest cluster), which appears to be more of a noise cluster rather than being topic-specific. In more detail:

1. Cluster 2 is linked with "Survival Analysis";
2. Cluster 3 is a mix between "Asymptotics" and "Stochastic Processes";
3. Cluster 4 focuses on "Robust Methods";
4. Cluster 5 primarily discusses "Causal Inference";
5. Cluster 6 is a mix between "Time Series" and "Clustering";
6. Cluster 7 addresses "Hypothesis Test";
7. Cluster 8 discusses "Dependency Structure";
8. Cluster 9 primarily addresses the combination of "Causal Inference" and "Optimization";
9. Cluster 9 is linked to "Financial Market".

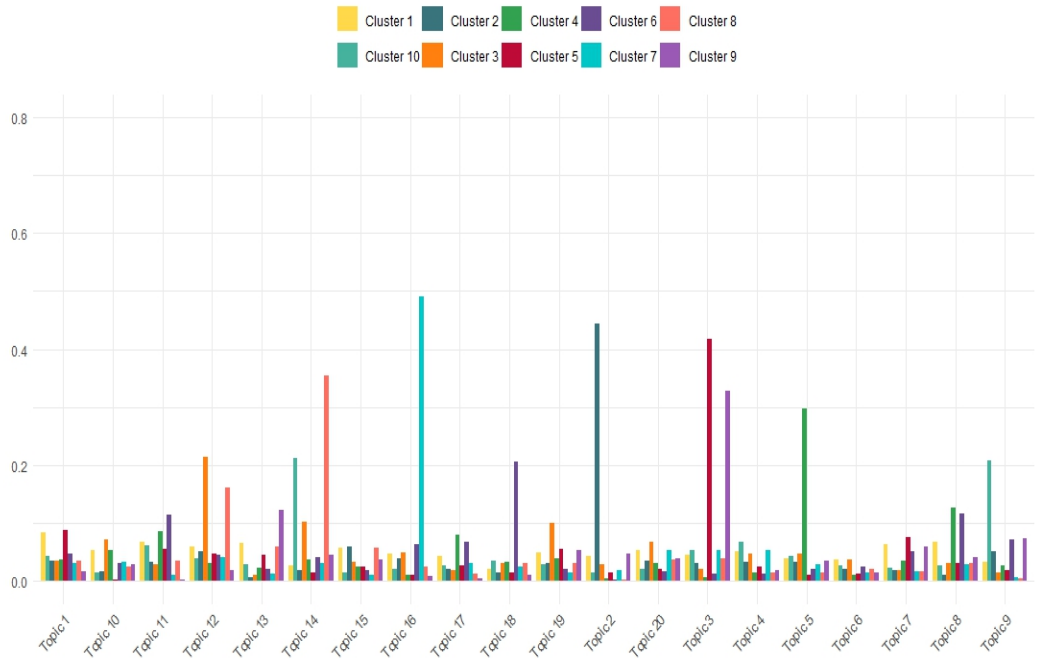

Figure 4.0.5. Topics prevalence inside cluster for 2014.

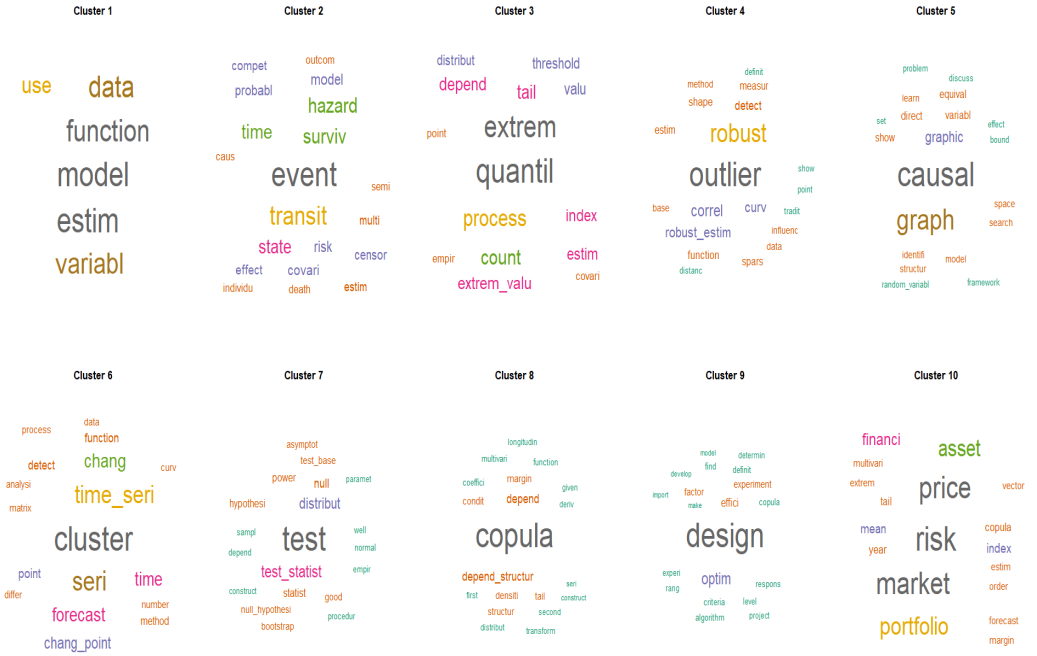

Figure 4.0.6. Wordclouds of the most frequent words inside clusters for 2014.

Other topics are discussed to some extent across various clusters, making a specific assignment difficult. Table 4.0.4 lists the document most similar to the centroid in each cluster, corroborating the observations mentioned above.

Title	Similarity	Cluster	Cluster size
E903: Hierarchical Bayesian LASSO for a negative binomial regression	0.3897	1	431
E055: An overview of multistate modeling with R	0.601	2	33
E045: Estimation of extreme conditional quantiles through power transformation	0.589	3	51
E142: Subspace search for outlier detection and description	0.605	4	27
E766: Causal models with hidden variables	0.7309	5	16
E841: Fuzzy probabilistic-distance clustering of time and numerical series modeled by penalized spline	0.563	6	34
E594: Permutation and randomization tests of parameters	0.633	7	25
E1075: Simultaneous inference in structured additive conditional copula regression models	0.692	8	15
E645: Algorithms for factorial designs generation	0.835	9	10
E745: Portfolio optimisation under switching dependence	0.541	10	28

Table 4.0.4. Documents with the highest cosine similarity with respect to the cluster centroid for 2014.

For the year 2020, following the same reasoning applied in the previous two analyses, observing Figure 4.0.7 and Figure 4.0.8 it can be asserted that all clusters are linked to specific topics, with the exception of the first cluster, which encompasses a variety of topics. A more detailed analysis reveals that:

1. Cluster 2 is associated with "Causal Inference";
2. Cluster 3 is a mix of "Survival Analysis" and "Variable Selection";
3. Cluster 4 pertains to "Hypothesis Testing";
4. Cluster 5 discusses "Network Analysis";
5. Cluster 6 is a combination of "Time Series" and "Asymptotics";
6. Cluster 7 addresses "Extreme Value Theory" and "Regression Models";
7. Cluster 8 primarily discusses "Clustering" and "Bayesian Statistics" within the context of Mixture Models;

8. Cluster 9 is focused on the "Microbiome," a very specific topic;

9. Cluster 10 refers to "Dependency".

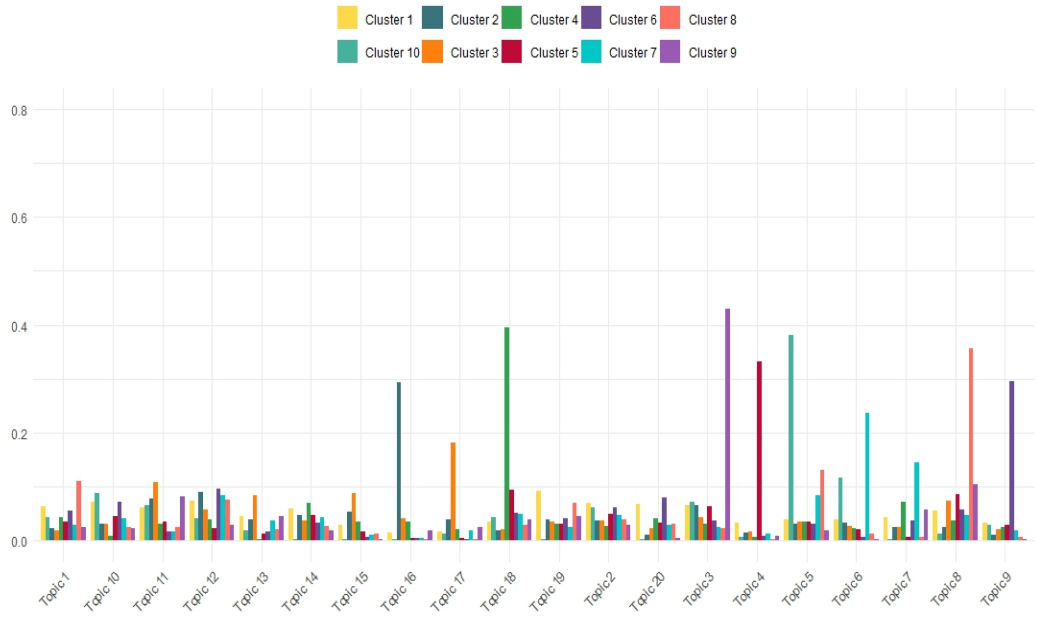

Figure 4.0.7. Topics prevalence inside cluster for 2020.

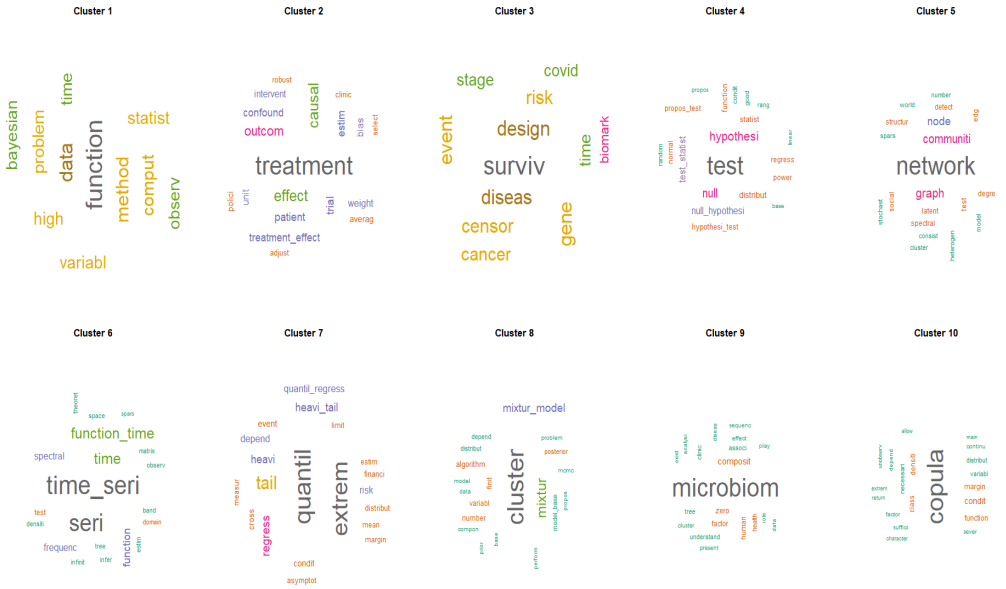

Figure 4.0.8. Wordclouds of the most frequent words inside clusters for 2020.

Table 4.0.5 shows the document most similar to each cluster according to cosine similarity. In conclusion, I can definitively state that the core topics and clusters have not changed significantly between 2014 and 2020, although in both years there are some documents that are difficult to cluster effectively. In my opinion, this is because these documents contain an equal proportion of several topics, which is consistent with the nature of statistical papers. Consider this thesis, where Topic Modeling, Hypothesis Testing, Bootstrap Methods, and Clustering have been discussed.

Title	Similarity	Cluster	Cluster size
E0855: Persistence via exact excursion time distributions	0.373	1	411
E0741: Double machine learning for (weighted) dynamic treatment effects	0.574	2	63
E0509: A nonparametric survival estimation method for dependent competing risk: An application in relative survival analysis	0.4311	3	83
E0883: Omnibus test for normality based on the Edgeworth expansion	0.5909	4	27
E0256: Community detection for hypergraph networks via regularized tensor power iteration	0.6292	5	28
E0835: Sparse PLS-DA: Clustering time series for art conservation	0.7558	6	22
E1015: On second-order automatic bias reduction for extreme expectile estimation	0.6324	7	30
E0413: Bayesian clustering of high-dimensional data	0.765	8	14
E0718: Compositional mediation models: Application to microbiome data	0.807	9	9
E0849: Grid-uniform copulas and rectangle exchanges: Model and Bayesian inference for a rich class of copula functionse	0.757	10	10

Table 4.0.5. Documents with the highest cosine similarity with respect to the cluster centroid for 2020.

Figure 4.0.9 illustrates the assignment of documents to different clusters in the two-dimensional space obtained using UMAP reduction. Notably, the assignment of documents for 2014 and 2020 is quite clear, as is the noisy nature of cluster number 1. However, the situation for 2008 appears much less clear.

Figure 4.0.9. Representation of the documents after a UMAP reduction for the three different years.

Chapter 5

Conclusions and final remarks

5.1 Future work and possible extensions

During this thesis, various techniques were explored and discussed in detail, both from theoretical and practical perspectives. Moreover, a novel approach to document clustering has been explored, leveraging the p-value associated with the homogeneity of topic distributions within documents. The foundational idea, inspired by [35], is intriguing. It is important to mention that the impact of changing the test statistic from the Kullback-Leibler divergence to the Bhattacharyya distance on the estimated p-value has been studied obtaining similar results in a reduced setup. Then, the estimated p-value has been employed as a similarity measure for subsequent analysis, attempting to develop a test-based clustering procedure as well as two other techniques which relies on agglomerative hierarchical clustering and fuzzy relational clustering.

Overall, the concept of comparing documents based on their topics offers a fresh perspective, moving away from traditional methodologies in the field. Additionally, this approach provides improved explanations and interpretations of the clustering results.

The results obtained in Chapter 3 reveal that these new techniques exhibit behavior similar to well-known methods in the literature (presented in Chapter 2), although they were not the optimal solution for the problem in any of the considered

examples. Additionally, the relatively small size of the corpora raises questions about the reliability of the results. From an interpretive standpoint, they enhance the understanding of how documents are clustered based on their topics. Conversely, it has also been demonstrated that Spectral Clustering and agglomerative hierarchical clustering perform well in this particular field, as highlighted in the literature.

There are several limitations to these new techniques:

1. **computational time** - the computational time required to compute the Bootstrap replicates for the approximation of the p-value. This can dramatically impact all subsequent clustering analyses. The reliance of these methods on LDA significantly slows down computations, suggesting it might be beneficial to explore other topic modeling techniques and redefine the strategy used. In order to be able to prove the efficiency of the methods, this would be the main issue to solve;

2. **reliance on BOW mechanism** - according to me, a minor issue is the reliance on a BOW mechanism, which may fails to capture the linkage and semantics between words. To mitigate this problem, although it cannot be fully solved, I included bigrams and trigrams in the preprocessing step;

3. **centroid definition** - it is important to note that in both the procedure to estimate the p-value and the LUC method, the centroid is obtained merely by summing the corresponding rows in the DTM. It is crucial to develop a new method for creating the cluster centroid;

4. **static procedure** - time is not directly considered in the analysis, but topics may change over time, making it overly simplistic to ignore temporal factors. Therefore, the procedure for estimating the p-value (affecting the clustering procedure) should be modified to account for time, for instance by employing the dynamic version of LDA [8].

Lastly, it is crucial to emphasize that a new internal cluster validity index needs to be defined for document clustering, as existing indices are not perfectly suitable

for selecting the number of clusters in this particular domain. This remains an open problem in the field, as evidenced by the publication of [48].

Moving forward, in Chapter 4, a real case scenario was discussed, applying agglomerative hierarchical clustering using cosine dissimilarity. The study focused on how topics evolved in the CMStatistics conference and defined "folders" of similar documents according to those topics. Notably, 2008 laid the foundations for the conference, and the main topics remained consistent in 2014 and 2020

In conclusion, there is significant potential for further development of these new methods, as they are not yet suitable for real-life applications. In particular, a new approach must be found to reduce computational costs without substantially altering the original idea.

5.2 AI Usage

Throughout the course of this thesis, the use of chatbots and, more generally, large language models has been minimal. Specifically, I have only employed them [53, 18] to make occasional corrections to the English language.

Appendix A

Code Appendix

- **Matrix of p-values.**

```
p_value_matrix = function(ind, m, dtm, k, B, theta){
  #' @param ind description: indices to consider in the document-term
      matrix
  #' @param m description: empty matrix of size M x M
  #' @param k description: number of topics for LDA
  #' @param B description: number of Bootstrap replicates
  #' @param theta description: document topic matrix obtained fitting
      LDA

  num_cores = detectCores(logical = T)-1
  plan(multisession, workers = num_cores)
  message("Number of parallel workers: ", nbrOfWorkers())
  # Start the parallelized session
  iter = 1
  for (i in seq_len(nrow(ind))) {
    cat("We are at the iteration number", iter, "over", nrow(ind), "\n")
    m[ind[i,1], ind[i,2]] = boottest_similarity(dtm = dtm, index1 =
        ind[i, 1], index2 = ind[i,2], k = k,  B = B, theta =
        theta)$pvalue
    # compute the Bootstrap pvalue
    m[ind[i, 2], ind[i,1]] = m[ind[i,1], ind[i,2]] # assign the
        computed value to the element in  the lower triangular matrix
    iter = iter + 1
  }
  return(m)
}
```

- **Test based clustering method.**

```
clustering_method = function(mat, dtm, alpha, B, k){
  #' @param mat description: starting p-values matrix
  #' @param dtm description: initially document term matrix
  #' @param alpha description: significance level
  #' @param B description: Bootstrap replicates
  #' @param K description: number of topics

  M = ncol(mat)
  partizione = lapply(1:M, function(x) list(x)) # save the partition
      for each iteration
  colnames(mat) = paste("Cluster", sep = " ", seq_len(M))
  storage = matrix(0, nrow = 1, ncol = M) # store clusters
  colnames(storage) = colnames(mat)
  p_values = numeric(M) # save merging height
  au = colnames(mat)
  cluster = 1
  iter = 1
  partizione[[iter]] = storage  # save the trivial partition

  while (any(mat[upper.tri(mat) == T] >= alpha) &  (iter < M-1)) {
  # while there exists a p_value greater than alpha
    iter = iter + 1
    cat("Iteration number", iter, "/", M, "\n")
    inde = which(upper.tri(mat) == T & mat == max(mat), arr.ind =
        T)[1,] # find the documents corresponding to the highest pvalue
    p_values[iter] = max(mat) # save this value

    # Storage procedure:
    # - find the names of the clusters;
    # - merge them in the storage value according to the rule below
    r = au[inde[1]]
    r2 = au[inde[2]]
    if (storage[1, r] == 0 & storage[1, r2] == 0){
      storage[1, r] = cluster
      storage[1, r2] = cluster
      cluster = cluster + 1}
    else if (storage[1, r] > 0 & storage[1, r2] == 0) {
      storage[1, r2] = storage[1, r]}
    else if (storage[1, r] ==  0 & storage[1, r2] > 0) {
      storage[1, r] = storage[1, r2] }
    else if (storage[1, r] > 0 & storage[1, r2] > 0) {
      storage[storage == storage[1, r2]] = storage[1, r] }

```

```
    partizione[[iter]] = storage # save the obtained partition

    if (iter == M-1) {break}

    dtm[inde[1], ] = apply(dtm[c(inde[1], inde[2]), ], 2, sum) # merge
        the two documents in the Document Term matrix

    dtm = dtm[-c(inde[2]), ] # delete the second document from the
        Document Term matrix

    au = au[au != r2] # delete from the list of all the clusters the
        one that has been merged

    dtm = as.dfm(dtm)
    mat = mat[-c(inde[2]), -c(inde[2])] #delete from the matrix of
        pvalue the row and the column of the cluster that has been
        merged
    # We need to compute the pvalues between the new document and the
        remaining ones

    s = seq(from = 1, to = ncol(mat)) # define the sequence of all the
        remaining clusters excluding the newly formed

    s = s[s != inde[1]]

    index = data.frame(rep(inde[1], length(s)), s)
    # This part of code is needed due to the fact the function pvalue
        matrix relies on another function that work exclusively when
        index1 > index2
    for (i in 1:nrow(index)) {
      if (index[i, 1] > index[i, 2]) {
        aux = index[i, 1]
        index[i, 1] = index[i, 2]
        index[i, 2] = aux
      }
    }
    z = LDA(dtm, k = k, method = "Gibbs",
            control = list(alpha = 1/k, iter = 1000)) # fit LDA with
                the new Document Term matrix
    theta = z@gamma
    mat = p_value_matrix(ind = index, m = mat, dtm = dtm, k = k, B = B,
        theta = theta) # compute the pvalue between the newly formed
        cluster and the others
  }
```

```
  plan(sequential) # stop the parallelized procedure
  print("Resting my workers")
  results = list(clustering = storage, height = p_values, partition =
      partizione)
  return(results)
}
```

- **Overall similarity:**

```
Cosine_Similarity = function(theta, sizes, pred, values){
  #' @param theta description: original DTM
  #' @param sizes description: numebr of elements inside each cluster
  #' @param pred description: estimated membership belonging of a unit
      to a cluster
  #' @param values description: unique clusters.
  M = sum(sizes)
  k = length(sizes)
  sim = numeric(k)
  for (i in 1:k) {
    if(sizes[i] == 1){
      centroid = theta[pred == values[i], ]
      sim[i] = cosi(theta[pred == values[i], ], centroid)
    }
    else{
      centroid = apply(theta[pred == values[i], ], 2, mean)
      sim[i] = mean(apply(theta[pred == values[i], ], MARGIN = 1, cosi,
          centroid))
    }
  }
  ov = sum(sim *(sizes/M))
  return(ov)
}
```

Bibliography

[1] Dimo Angelov. "Top2vec: Distributed representations of topics". In: *arXiv preprint arXiv:2008.09470* (2020).

[2] David Arthur, Sergei Vassilvitskii, et al. "k-means++: The advantages of careful seeding". In: *Soda*. Vol. 7. 2007, pp. 1027–1035.

[3] Rekha Baghel and Renu Dhir. "A frequent concepts based document clustering algorithm". In: *International Journal of Computer Applications* 4.5 (2010), pp. 6–12.

[4] Jeffrey D Banfield and Adrian E Raftery. "Model-based Gaussian and non-Gaussian clustering". In: *Biometrics* (1993), pp. 803–821.

[5] Florian Beil, Martin Ester, and Xiaowei Xu. "Frequent term-based text clustering". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 436–442.

[6] Anil Bhattacharyya. "On a measure of divergence between two statistical populations defined by their probability distribution". In: *Bulletin of the Calcutta Mathematical Society* 35 (1943), pp. 99–110.

[7] David Blei and John Lafferty. "Correlated topic models". In: *Advances in neural information processing systems* 18 (2006), p. 147.

[8] David M Blei and John D Lafferty. "Dynamic topic models". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 113–120.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[10] Edoardo Bompiani et al. "High-Performance Computing with TeraStat". In: *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. 2020, pp. 499–506. DOI: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00092.

[11] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. "Density-based clustering based on hierarchical density estimates". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2013, pp. 160–172.

[12] Chun-Ling Chen, Frank SC Tseng, and Tyne Liang. "An integration of WordNet and fuzzy association rule mining for multi-label document clustering". In: *Data & Knowledge Engineering* 69.11 (2010), pp. 1208–1226.

[13] Irene Cozzolino et al. "Fuzzy spectral clustering methods for textual data". In: (2023).

[14] Irene Cozzolino and Maria Brigida Ferraro. "Document clustering". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 14.6 (2022), e1588.

[15] Radu G Creţulescu et al. "DBSCAN algorithm for document clustering". In: *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences* 9.1 (2019), pp. 58–66.

[16] Douglass R Cutting et al. "Scatter/gather: A cluster-based approach to browsing large document collections". In: *ACM SIGIR Forum*. Vol. 51. 2. ACM New York, NY, USA. 2017, pp. 148–159.

[17] Rajesh N Davé and Sumit Sen. "Robust fuzzy clustering of relational data". In: *IEEE transactions on Fuzzy Systems* 10.6 (2002), pp. 713–727.

[18] DeepL SE. *DeepL Translator*. https://www.deepl.com/translator. 2017.

[19] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[20] Matthew J Denny and Arthur Spirling. "Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it". In: *Political Analysis* 26.2 (2018), pp. 168–189.

[21] Chris HQ Ding et al. "A min-max cut algorithm for graph partitioning and data clustering". In: *Proceedings 2001 IEEE international conference on data mining*. IEEE. 2001, pp. 107–114.

[22] B Efron. "The 1977 RIETZ lecture". In: *The annals of Statistics* 7.1 (1979), pp. 1–26.

[23] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[24] Chris Fraley and Adrian E Raftery. "Model-based clustering, discriminant analysis, and density estimation". In: *Journal of the American statistical Association* 97.458 (2002), pp. 611–631.

[25] Gil González-Rodrıguez et al. "Multi-sample test-based clustering for fuzzy random variables". In: *International Journal of Approximate Reasoning* 50.5 (2009), pp. 721–731.

[26] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[27] Anna Huang et al. "Similarity measures for text document clustering". In: *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. Vol. 4. 2008, pp. 9–56.

[28] Reza Nurul Gayatri Indah et al. "DBSCAN algorithm: twitter text clustering of trend topic pilkada pekanbaru". In: *Journal of physics: conference series*. Vol. 1363. 1. IOP Publishing. 2019, p. 012001.

[29] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[30] Pankaj Jajoo. "Document clustering". In: *IIT Kharagpur, Thesis* (2008).

[31] Fred Jelinek et al. "Perplexity—a measure of the difficulty of speech recognition tasks". In: *The Journal of the Acoustical Society of America* 62.S1 (1977), S63–S63.

[32] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[33] Aurangzeb Khan et al. "A review of machine learning algorithms for text-documents classification". In: *Journal of advances in information technology* 1.1 (2010), pp. 4–20.

[34] Daphne Koller and Mehran Sahami. "Hierarchically classifying documents using very few words". In: *ICML*. Vol. 97. 1997, pp. 170–178.

[35] Louisa Kontoghiorghes and Ana Colubi. "Testing the Homogeneity of Topic Distribution Between Documents of a Corpus". In: *International Conference on Soft Methods in Probability and Statistics*. Springer. 2022, pp. 248–254.

[36] Gerald J Kowalski. *Information retrieval systems: theory and implementation*. Vol. 1. springer, 2007.

[37] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[38] Michael Steinbach George Karypis Vipin Kumar, M Steinbach, and G Karypis. "A comparison of document clustering techniques". In: *Department of Computer Science and Engineering, University of Minnesota* (2000).

[39] Bjornar Larsen and Chinatsu Aone. "Fast and effective text mining using linear-time document clustering". In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. 1999, pp. 16–22.

[40] Changzhou Li et al. "LDA meets Word2Vec: a novel model for academic abstract clustering". In: *Companion proceedings of the the web conference 2018*. 2018, pp. 1699–1706.

[41] Huma Lodhi et al. "Text classification using string kernels". In: *Journal of machine learning research* 2.Feb (2002), pp. 419–444.

[42] Yinglong Ma, Yao Wang, and Beihong Jin. "A three-phase approach to document clustering based on topic significance degree". In: *Expert Systems with Applications* 41.18 (2014), pp. 8203–8210.

[43] Elizabeth Ann Maharaj. "A significance test for classifying ARMA models". In: *Journal of Statistical Computation and Simulation* 54.4 (1996), pp. 305–331.

[44] Bryan FJ Manly. *Randomization, bootstrap and Monte Carlo methods in biology*. chapman and hall/CRC, 2018.

[45] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[46] Duoqian Miao et al. "Rough set based hybrid algorithm for text classification". In: *Expert Systems with Applications* 36.5 (2009), pp. 9168–9174.

[47] David Mimno et al. "Optimizing semantic coherence in topic models". In: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, pp. 262–272.

[48] Michelangelo Misuraca, Maria Spano, and Simona Balbi. "BMS: An improved Dunn index for Document Clustering validation". In: *Communications in statistics-theory and methods* 48.20 (2019), pp. 5036–5049.

[49] A Murua et al. "Model based document classification and clustering". In: *Manuscript in preparation* (2001).

[50] David Newman et al. "Evaluating topic models for digital libraries". In: *Proceedings of the 10th annual joint conference on Digital libraries*. 2010, pp. 215–224.

[51] Andrew Ng, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems* 14 (2001).

[52] Aytug Onan, Hasan Bulut, and Serdar Korukoglu. "An improved ant algorithm with LDA-based representation for text document clustering". In: *Journal of Information Science* 43.2 (2017), pp. 275–292.

[53] OpenAI. *ChatGPT: A Large-Scale Generative Model for Open-Domain Chat*. https://github.com/openai/gpt-3. 2021.

[54] Nikhil R Pal and James C Bezdek. "On cluster validity for the fuzzy c-means model". In: *IEEE Transactions on Fuzzy systems* 3.3 (1995), pp. 370–379.

[55] Lance Parsons, Ehtesham Haque, and Huan Liu. "Subspace clustering for high dimensional data: a review". In: *Acm sigkdd explorations newsletter* 6.1 (2004), pp. 90–105.

[56] Martin F Porter. "An algorithm for suffix stripping". In: *Program* 14.3 (1980), pp. 130–137.

[57] Martin F Porter. *Snowball: A language for stemming algorithms*. 2001.

[58] Cornelis J van Rijsbergen. "Towards an information logic". In: *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*. 1989, pp. 77–86.

[59] Gerard Salton. "Introduction to modern information retrieval". In: *McGraw-Hill* (1983).

[60] Lei Shi et al. "Rough set based decision tree ensemble algorithm for text classification". In: *Journal of Computational Information Systems* 6.1 (2010), pp. 89–95.

[61] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17 (2007), pp. 395–416.

[62] Kohei Watanabe and Phan Xuan-Hieu. *seededlda: Seeded Sequential LDA for Topic Modeling*. https://github.com/koheiw/seededlda, https://koheiw.github.io/seededlda/. 2023.

[63] Lei Wu, Steven CH Hoi, and Nenghai Yu. "Semantics-preserving bag-of-words models and applications". In: *IEEE Transactions on Image Processing* 19.7 (2010), pp. 1908–1920.

[64] Chyi-Kwei Yau et al. "Clustering scientific documents with topic modeling". In: *Scientometrics* 100 (2014), pp. 767–786.

[65] Jianhua Yin and Jianyong Wang. "A dirichlet multinomial mixture model-based approach for short text clustering". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 233–242.

[66] Ying Zhao and George Karypis. "Comparison of agglomerative and partitional document clustering algorithms". In: (2002).

[67] Shi Zhong and Joydeep Ghosh. "Generative model-based document clustering: a comparative study". In: *Knowledge and Information Systems* 8 (2005), pp. 374–384.