SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Department of Statistical and Decision Sciences
PhD in Methodological Statistics

Thesis For The Degree Of Doctor Of Philosophy

Fuzzy spectral clustering methods for textual data

Supervisor
Prof. Maria Brigida Ferraro

Candidate
Irene Cozzolino

Academic Year MMXIX-MMXXII (XXXV cycle)

Abstract

Nowadays, the development of advanced information technologies has determined an increase in the production of textual data. This inevitable growth accentuates the need to advance in the identification of new methods and tools able to efficiently analyse such kind of data. Against this background, unsupervised classification techniques can play a key role in this process since most of this data is not classified. Document clustering, which is used for identifying a partition of clusters in a corpus of documents, has proven to perform efficiently in the analyses of textual documents and it has been extensively applied in different fields, from topic modelling to information retrieval tasks. Recently, spectral clustering methods have gained success in the field of text classification. These methods have gained popularity due to their solid theoretical foundations which do not require any specific assumption on the global structure of the data. However, even though they prove to perform well in text classification problems, little has been done in the field of clustering. Moreover, depending on the type of documents analysed, it might be often the case that textual documents do not contain only information related to a single topic: indeed, there might be an overlap of contents characterizing different knowledge domains. Consequently, documents may contain information that is relevant to different areas of interest to some degree.

The first part of this work critically analyses the main clustering algorithms used for text data, involving also the mathematical representation of documents and the pre-processing phase. Then, three novel fuzzy versions of spectral clustering algorithms for text data are introduced. The first one exploits the use of fuzzy K-medoids instead of K-means. The second one derives directly from the first one but is used in combination with *Kernel and Set Similarity* (KS^2M), which takes into account the Jaccard index. Finally, in the third one, in order to enhance the clustering performance, a new similarity measure $\mathbf{S}^*$ is proposed. This last one exploits the inherent sequential nature of text data by means of a weighted combination between the Spectrum string kernel function and a measure of set similarity.

The second part of the thesis focuses on spectral bi-clustering algorithms for text mining tasks, which represent an interesting and partially unexplored field of research. In particular, two novel versions of fuzzy spectral bi-clustering algorithms are introduced. The two algorithms differ from each other for the approach followed in the identification of the document and the word partitions. Indeed, the first one follows a simultaneous approach while the second one a sequential approach. This difference leads also to a diversification in the choice of the number of clusters. The adequacy of all the proposed fuzzy (bi-)clustering methods is evaluated by experiments performed on both real and benchmark data sets.

Contents

List of Figures

List of Tables

Chapter 1

General introduction

The main pillar of this thesis relies on unsupervised classification algorithms for document data sets based on fuzzy spectral clustering methods.

The first part of the thesis revolves around a novel fuzzy spectral clustering algorithm, which exploits the employment of fuzzy K-medoids.
Though spectral clustering has many advantages, yet one of the drawbacks is that the clustering results are based on a crisp assignment of the data points to the corresponding clusters. In the field of document classification, this can lead to unrealistic results since the documents with intermediate characteristics between two or more clusters are forced to belong to exactly one cluster. We have solved this problem by introducing a modified fuzzy spectral clustering algorithm, which is based on two main steps. The first step is to build the Laplacian matrix of the graph-based representation of the collection of documents. The second step is to obtain clustering results by employing fuzzy K-medoids instead of the standard K-means algorithm.

Initially, the new fuzzy spectral clustering algorithm is used in combination with *Kernel and Set Similarity* (KS^2M), which represents a known similarity measure for text documents. This leads to the development of a second version of fuzzy spectral clustering algorithm.
However, in order to improve the accuracy of the clustering results, a novel similarity measure for text data, $\mathbf{S}^*$, is also introduced.
The new proposed similarity measure relies on both overlapping coefficient, used as a measure of set similarity, and string kernel function, used as a measure of sequence similarity. Indeed, text data are characterized by an inherent sequential nature that should be properly captured, since each document in the collection can be seen as an ordered sequence of items.
The proposed metric is then used to build the matrix of input of the novel fuzzy spectral clustering algorithm, giving rise to a third fuzzy spectral clustering al-

gorithm.

The second part of the thesis focuses on spectral bi-clustering methods applied to text data, representing a great exploratory tool for highly complex data. Bi-clustering algorithms are a branch of clustering techniques whose aim is to identify clusters of data points that are related to subset of features and, on the other hand, identifying clusters of features that share similar characteristics on subsets of data points.
Against this background, two novel fuzzy spectral bi-clustering algorithms for text data, resulting from an extension of Dhillon's and Kluger's state-of-art methods, are introduced. The contribution of the new proposed methods consists in allowing for an overlapping between word clusters and document clusters. Moreover, the novel bi-clustering algorithms prove to improve the accuracy of the clustering results compared to the corresponding crisp counterparts.
The main difference between the two proposed methods is in their implementation. The first method is based on a simultaneous approach to bi-clusters while the second one follows a sequential approach.

The main contribution of this thesis consists in pursuing the goal to improve the understanding and the interpretability of clustering results in the field of unsupervised classification of document data sets. Our work can be seen as an attempt to improve the classification of text documents through the employment of a fuzzy approach to clustering/bi-clustering, resulting in a more realistic assignment of data points to clusters. Moreover, we also attempted to improve the accuracy of the clustering results.

The remaining part of the thesis is organized as follows: in Chapter 2 the main document clustering approaches available in literature, for each class of clustering algorithms (prototype-based, graph-based, hierarchical and model-based), are analysed. In Chapter 3 the novel one-mode fuzzy spectral clustering algorithms for document data sets, used in combination with, respectively, *Kernel and Set Similarity* (KS^2M) and $\mathbf{S}^*$, are introduced and described. Chapter 4 analyses the spectral bi-clustering setting and introduces the two novel fuzzy spectral bi-clustering algorithms. Finally, conclusions and open problems come out in Chapter 5.

Chapter 2

Literature overview on document clustering techniques

In this chapter the main document clustering approaches for each class of clustering algorithms (prototype-based, graph-based, hierarchical and model-based) are analysed.
In Section 2.2, a critical review on the main steps of the document clustering process is carried out: special attention is given to the mathematical representation of documents, taking into consideration the pre-processing phase and the different term-weighting schemes used in the construction of the Vector Space Model.
Then, in Section 2.3, the main characteristics of the most used clustering algorithms for text data for every of the aforementioned categories are critically analysed: spherical K-means for prototype-based methods, spectral clustering for graph-based methods, divisive and agglomerative algorithms with different criterion functions for hierarchical methods and GMM for model-based methodos.
Furthermore, starting from the above proposals, more advanced methods are considered. For further details refer to Cozzolino and Ferraro (2022).

2.1 Introduction

Text clustering consists in the application of cluster analysis to text data. Given the high level of granularity in text data, clustering techniques prove to be very useful in this field. In particular, document clustering refers to the application of cluster analysis at document level and is used to partition a collection of text documents into homogeneous groups according to their similarity.
It was at first used in information retrieval (IR) systems for enhancing the precision and recall (Van Rijsbergen, Harper, & Porter, 1981). Nowadays, due to the increasing number of text data, document clustering is used for different ap-

plications: document structuring (such as the organization of large electronic archives or the classification of documents in taxonomies), topic extraction, web mining and search optimization (in this setting, clustering methods are useful for improving the performance of web browsers since the user sentences are initially compared with the content of the clusters instead of the documents).

In document clustering each document in the collection (*corpus*) is converted into a vector in a multidimensional space and clustering aims at identifying a partition of documents based on the inherent structure of the newly-formed space. More specifically, traditional document clustering algorithms rely on the bag-of-word (BOW) representation, where the order of words within each document and the order of files in the collection is not statistically significant. The main drawback of the BOW approach is that the semantic between words is not taken into consideration: those terms that are semantically connected, such as hyper/hyponyms or synonyms, are not taken into account. For instance, words such as *company*, *firm*, and *enterprise* are considered different terms even though they share approximately the same meaning and can be used indiscriminately within a text.

In this setting, the identification of a measure to establish the similarity between two feature vectors plays a key role in document clustering techniques. Several similarity and distance measures have been proposed in literature, examples include the Jaccard correlation coefficient, the Kullback-Leibler divergence and the cosine similarity. The first one is a measure of similarity between two sets and it is defined as the size of their intersection over the size of their union. It is useful when the replication of the same word in two distinct documents does not influence their similarity. The second one is used to compare two probability distributions, P and Q, where the former one is considered to be the target probability distribution. Kullback-Leibler divergence measures the expected loss of information when using Q instead of P; it refers to a probabilistic approach to text mining. For P and Q being two discrete probability distributions defined over the space $\mathcal{X}$, the Kullback-Leibler divergence is defined as:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \cdot ln \left(\frac{P(x)}{Q(x)} \right). \tag{2.1}$$

Finally, the last one is defined as the cosine of the angle formed by two vectors and it is useful when the documents exhibit words written in the same way but having different meanings (in this context the repetition of such a word can halter the similarity between two documents). An overview of the most widely used similarity measures is provided in Huang (2008).

It worth emphasising that starting from a corpus and arriving to a partition of documents does not consist in a single step. It involves numerous operations that in general include pre-processing, document representation by means of numerical vectors and clustering.
The very last step consists in applying cluster analysis to the mathematical representation of documents. Algorithms for document clustering, where the semantic is not considered, can be divided into partitional, graph-based, hierarchical and model-based. A detailed description of these methods is provided.

Some detailed document clustering reviews are addressed in Shah and Mahajan (2012) and Premalatha and Natarajan (2010) which consist of describing the general document clustering process and its challenges, focusing mainly on extensions of K-means applied in the context of document clustering and the conventional hierarchical clustering algorithms. Furthermore, in addition to the aforementioned works, Bisht and Paul (2013) analyse also the frequent itemset based clustering approach which consists in a set of techniques that do not require the Vector Space Model representation of the corpus.
In this work, we go beyond K-means and conventional hierarchical clustering, reviewing the most common document clustering methodologies while considering, within a certain extent, the main classes of algorithms.
For semantic document clustering techniques refer to Fahad and Yafooz (2017) for a detailed review.

It is worth noticing that, in this work, we decided to focus on unsupervised classification techniques. However, there are also many supervised text categorization proposals in literature (for a detailed survey on the main text classification algorithms see, for instance, Aggarwal and Zhai (2012)).
In a supervised framework, Support Vector Machines (SVM) (Cortes and Vapnik (1995), Vapnik (1999)) have gained considerable attention due to their good performance in text categorization tasks: for instance, in Joachims (1998) the authors highlight both how SVM are able to capture the intrinsic structure of a text (high dimensionality, sparsity, few irrelevant features) and also their robustness, outperforming in this regard other existing methods.

2.2 Document representation

Since most clustering methods require numerical features, it is necessary to transform the corpus of documents into a mathematical object that can be passed as input to clustering algorithms. The representation of a set of documents into numerical attributes is called Vector Space Model (VSM) and will be analysed in Section 2.2.2. Nevertheless, the construction of a VSM requires a pre-processing step that takes place directly on the documents written in natural language. The pre-processing phase aims at removing the noise from text data (e.g. the non-meaningful terms) and hence reducing the dimensions of the feature-space.

2.2.1 Pre-processing

Pre-processing plays a key part in document clustering techniques since it is the very first phase of the entire process. The main steps of pre-processing are: tokenization, filtering, pruning, stemming & lemmatization.

- **Tokenization**: This step separates each stream of text data into smaller elements called tokens. Tokens can be of different dimensions: unigram, bigram, ..., n-gram. Word (n-gram) tokenization is the most commonly used one, assuming the white-space as a delimiter. In Webster and Kit (1992) a detailed description of tokenization as the very first step in text mining applications is provided. The work focuses on the description of the main approaches to tokenization which are, respectively, the lexicography approach (with the consequent definition of what is considered to be a token) and the mechanical approach employing, among the others, dictionary-based techniques. Furthermore, insights on how to identify compounds tokens in English and how to handle the ambiguity of terms are also provided. Finally, it also discusses on the complexity of tokenization in languages such as Chinese characterized by the absence of words.

- **Filtering**: In this step special characters, punctuation marks and stopwords are removed. Stopwords are those words which do not convey any semantic meaning to the comprehension of the documents, such as pronouns, conjunctions, articles or adverbs. Each language has its specific list of stopwords. Removing stopwords has the effect to reduce the dimension of the term-space. The standard method for stopword removal consists in comparing each single term appearing in the corpus with a sequence of already recognized stopwords (Jivani et al., 2011). In addition to the classic stop list, it is possible also to use supervised-learning approaches to perform automatic feature selection, such as the Mutual Information (MI) (Shannon

(2001), Cover (1999)) method. Indeed, this method is based on calculating the mutual information between a specific word and a document category (e.g., positive, negative). Mutual information between two random variables calculates the amount of information the first variable shares with the other one; it is interpreted as the reduction of uncertainty of one random variable given the other. The intuition behind this approach consists in comparing the joint probability of observing the term and the category with the probabilities to observe the category and the term independently. In other words, MI quantifies the amount of information the term provides about a given class. If the MI value is low, then the term is characterized by a low discriminating strength and consequently it can be deleted from the collection (Jivani et al., 2011; Sharma & Cse, 2012). Another more recent approach is the so-called Term Based Random Sampling (TBRS) (Lo, He, & Ounis, 2005), based on the Kullback-Leibler divergence to assess the importance of each word. The collection is randomly divided into different subsets of documents. Each term is randomly selected from each chunk and its informative power is evaluated through the Kullback-Leibler divergence. The idea behind this approach consists in measuring the divergence of the distribution of a given term in the collection from the distribution of the same term within the sampled set of documents. Indeed, the objective is to find the terms that better complement the initially chosen subset of documents according to their overall distribution in the collection. As for the previous method, it is possible to automatically derive a suitable list of stopwords containing the least informative terms.

- **Pruning**: It is the process of removing those words having a very low or very high number of occurrences in the corpus. In this regard, it is common to employ a specific threshold that should be appropriately identified. In other words, it consists in deleting those stopwords specific of the considered corpus according to their frequencies: indeed, those terms characterized by very high frequencies are considered to be too common, while those with very low frequencies are too rare. For this purpose, it is necessary to properly identify an upper and a lower threshold. An application of this technique has been performed by Lenz and Winker (2020), by removing from the collection all the words that appeared in more than 65% and less than 0.05% of documents. In many cases, the thresholds should be determined empirically, namely until all corpus-specific stopwords are removed.

- **Stemming & Lemmatization**: Stemming (see, e.g., Krovetz, 2000) refers to the approach used to identify the root of each word by removing suffixes and prefixes. Porter stemming algorithm (Porter, 1980, 2001), is one of

the most famous stemming technique used for text mining applications. Lemmatization (see, e.g., Korenius, Laurikkala, Järvelin, & Juhola, 2004 and Balakrishnan & Lloyd-Yemoh, 2014) is a more complex approach: it consists in finding the base/dictionary form (lemma) of each word in the document. In order to identify the lemma is first necessary to establish the corresponding part of speech of the term. For this purpose, lemmatization algorithms usually rely on external dictionaries.

For a detailed review on pre-processing techniques for text mining applications see Vijayarani, Ilamathi, Nithya, et al. (2015).

2.2.2 Vector Space Model

Vector Space Model (VSM) is the statistical model used to determine the relevance between the documents in the collection and the words within each document. In the VSM, initially proposed by Salton (Salton, 1971), the documents are encoded by a set of multidimensional features spanned by the term vectors representing the vocabulary (obtained as the remaining list of unique words after performing the pre-processing). Thus, under the VSM a corpus of N documents with T unique terms is converted into an $N \times T$ matrix, where each single file in the collection is represented as a T-dimensional features vector. The $N \times T$ matrix is also known as Document Term Matrix, expressed in symbols as **DTM**. Sometimes, even if more rarely, the transposed of the **DTM**, the Term Document Matrix, is also considered as the mathematical representation of the collection. In the remaining part of the work we consider as VSM the **DTM**.

Each entry of the **DTM** represents an individual term weight associated to the corresponding document. Many term weighting schemes have been proposed in literature. One well known method is the binary weighting scheme, where each entry of the **DTM** can assume only the values 1 or 0 representing, respectively, the presence and the absence of a word in the current document. Another commonly used weighting scheme relies on word frequencies (TF weighting scheme), counting the terms occurrences within each document. Among the competitors, the most popularly used one is the Term Frequency - Inverse Document Frequency (TF-IDF) weighting scheme (Salton & McGill, 1983): if a word of the vocabulary appears with a high frequency in the current document, but rarely in the whole corpus, then the TF-IDF scheme assigns a high weight to the term. The words characterized by a high TF-IDF score are highly informative and can be useful in discriminating between the documents in the overall collection.
Considering a set of N documents with a T-sized vocabulary, the TF-IDF statistic

for the i-th document and the j-th term is calculated as follows:

$$w_{ij} = tf_{ij} \times log\left(\frac{N}{df_j}\right) \quad i = 1, \ldots, N \quad j = 1, \ldots, T, \tag{2.2}$$

where tf_{ij} represents the relative frequency of term j in document i; df_j is the number of documents containing the j-th word and N is the size of the corpus.

As reported in Salton and Buckley (1988), many variants of the TF-IDF measure have been proposed: depending on the type of data set used, they can return better results respect to TF-IDF.
A common extension of the TF-IDF measure consists in scaling sub-linearly the term frequency factor as $log(tf_{ij} + 1)$, in order to reduce the importance given to frequent terms by flattening the weight. As highlighted in Nguyen (2013), this proves to be beneficial when the term frequencies follow a power law with respect to the rank.
Okapi BM25 (Robertson & Zaragoza, 2009), more commonly known as BM25, is also a standard term weighting methodology used to establish the importance of a given term within the current document. The BM25 formula for a term weight is itself based on the TF-IDF measure but with variations in the way the components are calculated. The weight in the BM25 scheme for the j-th term and the i-th document is calculated as follows:

$$w_{ij} = IDF_j \times \frac{tf_{ij} \times (k_1 + 1)}{tf_{ij} + k_1 \times (1 - b + b \times \frac{|\mathbf{d}_i|}{avgLen})}, \tag{2.3}$$

$$IDF_j = ln\left(\frac{N - df_j + 0.5}{df_j + 0.5} + 1\right), \tag{2.4}$$

where IDF_j is the inverse document frequency of the j-th term in the vocabulary; $|\mathbf{d}_i|$ is the length of document i; $avgLen$ is the average document length in the collection. Then, k_1 and b are two free parameters that should be properly chosen. Following Manning, Raghavan, and Schutze (2008), k_1 is a non-negative parameter that controls the scaling of the TF component. If $k_1 = 0$, it returns the IDF_j; on the contrary, for high values of k_1, it returns the standard term frequencies (occurrences of the term in each document). The parameter b controls the scaling of the length of the documents and it varies in the interval $[0, 1]$; when it assumes a value equals to 0, then no normalization is performed.

In classification problems, where a train-test split of the data is carried out, k_1 and b should ideally be selected so to optimize the performance of the scheme on the test set. For this reason it is recommended to use optimization techniques. However, as reported in Manning et al. (2008), reasonable results have been ob-

tained by setting $k_1 \in [1.2, 2]$ and $b = 0.75$ in practical applications.

A detailed review of different term weighting schemes is provided in Lan, Tan, Su, and Lu (2008), where the authors have investigated the effectiveness of different supervised and unsupervised weighting schemes on two popular benchmark data corpus.

Other approaches consist in using as feature selection measures the following metrics: χ^2 (multiply tf_{ij} by a χ^2 function), information gain (multiply tf_{ij} by an information function), gain ratio (multiply tf_{ij} by a gain ratio), odds ratio (multiply tf_{ij} by an odds ratio) (Jones, 1972; Robertson, 2004).

2.3 Document clustering methods

The document clustering problem consists in partitioning the corpus of N documents, $\mathcal{C} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ into K clusters; each $\mathbf{d}_i \in \mathbb{R}^{\mathbb{T}}$ is an attribute vector in a T-dimensional space. The final objective of document clustering is to identify a small number K of homogeneous groups (clusters) by means of a certain dissimilarity measure calculated on the T observed features.
Clustering techniques are classified into two main approaches: hard and soft clustering.

Hard (crisp) clustering methods are characterized by computing the allocation of a document to a cluster: in other words, each document is forced to belong to only one cluster. This approach returns as output a partition of disjoint groups.

From a practical perspective, there exist documents that cannot be uniquely assigned to only one cluster since they show in-between characteristics among groups. The soft approach tries to solve this issue by calculating, for each document, a membership degree ranging in the interval $[0, 1]$, representing a measure of belonging to each cluster of the partition. Hence, each observation can be assigned to more clusters at the same time. Soft clustering methods divide into fuzzy, possibilistic and probabilistic.
For a more detailed review on soft clustering methods see Ferraro and Giordani (2020).

There are different types of clustering algorithms: prototype-based, graph-based, hierarchical and model-based.
Prototype-based algorithms identify a prototype for each group, and the obser-

vations are grouped around the prototypes. Within the most famous and extensively used prototype-based methods (crisp and soft, respectively) fall K-means (MacQueen, 1967) and Fuzzy K-Means (FKM) (Bezdek, 1981).
Despite K-means is considered one of the first 10 data mining algorithms (Wu et al., 2008), it is not excused from drawbacks. One of its main limitation consists in setting properly the initial prototypes since the method is sensible to the initialization phase (usually the centres are chosen uniformly at random from the data; consequently it is recommended to run the algorithm multiple times with different random seeds) and it may converge to non-optimum solution. Among the competitors, this problem has been addressed by Arthur and Vassilvitskii (2006), proposing a simple and fast alternative algorithm known as K-means ++. The method consists in randomly choosing the seeds but in such a way that the data are progressively weighted according to their squared distance from the closest center already chosen. Other attempts in this direction have been made by Nazeer and Sebastian (2009). Their algorithm consists in initially calculate the distance between each pair of data, then the first cluster is formed by considering the closest two data points. Successively, the other closest data points are added to the newly formed cluster until a certain threshold is reached. All the data points belonging to the first cluster are deleted from the initial set; the process continues until forming K initial clusters. The seeds are generated by averaging over all the vectors in each cluster. Babu and Murty (1993) propose a hybrid approach that consists in combining the genetic algorithms, for the initial seeds selection, and K-means. For a detailed review see Jain, Murty, and Flynn (1999).

Graph-based algorithms treat observations as nodes of a graph, and the distance between the two data points is used to weight the edge linking the two nodes. Hence, observations can be visualized as a graph and a connected subgraph identifies a cluster. Spectral clustering methods (A. Y. Ng, Jordan, & Weiss, 2002) are representative of graph-based class. These methods rely on the use of an affinity matrix, determining a connection between kernel methods and spectral clustering (see Dhillon, Guan, and Kulis (2004) for a discussion on the relationship between kernel methods and spectral clustering). Some of the most common kernel functions are: Gaussian and Fisher kernels, radial basis function kernel and polynomial kernel. In Section 2.3.2 specific kernel functions, used to define affinities between documents, are briefly analysed.

Hierarchical algorithms aim at identifying a hierarchical set of partitions. The graphical representation of hierarchies can be visualized taking advantage of specific tree-like structures by means of the so-called *dendograms*. A representative algorithm for this category is the Agglomerative hierarchical clustering (AHC) (Tan, Steinbach, & Kumar, 2006). For an exhaustive review of hierarchical meth-

ods see Rencher (2005).

Finally, model-based clustering algorithms are based on the assumption that the data follow a mixture of parametric probability models (mixture components). These methods calculate the posterior probability that each object belongs to one of the mixture components. In this framework, the most common one is the Gaussian mixture model (Fraley & Raftery, 1998). Successively, several extensions employing other probability distributions have been developed. For a more detailed review on the model-based approach refer to McLachlan, Lee, and Rathnayake (2019).

In the following sections the main clustering approaches for text data, for each of the aforementioned categories, are described.

2.3.1 Prototype-based methods

Compared to other competitors, such as hierarchical methods, prototype-based techniques are usually more suitable for large document data sets since the final results are more easily interpretable. However, these methods present the drawback to properly select the input parameters; among the others, the most important is the one representing the number of clusters in the partition, K. A non-suitable choice of this parameter might determine a poor accuracy.

The Euclidean distance is commonly adopted for many prototype-based clustering algorithms, including K-means. However, it is not suitable for text data since long documents, characterized by high term weights, are over-represented (Hornik, Feinerer, Kober, & Buchta, 2012). To weaken the consequences arising from different document lengths, Dhillon and Modha (2001) suggest to employ the cosine distance rather than the Euclidean one, coming up with the spherical K-means clustering algorithm.

The cosine distance between two generic vectors, $\mathbf{x}$ and $\mathbf{y}$, is expressed as follows:

$$d(\mathbf{x}, \mathbf{y}) = 1 - cos(\mathbf{x}, \mathbf{y}) = 1 - \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{||\mathbf{x}||||\mathbf{y}||},$$

where $cos(\mathbf{x}, \mathbf{y})$ is the corresponding cosine similarity, quantified as the cosine of the angle formed by the two vectors.

Within this framework it is worth noticing that the cosine similarity is widely applied in document clustering and it returns better results compared to the existing competitors (e.g. Euclidean distance). For instance in Zhao and Karypis

(2004), the authors study several objective functions for prototype-based document clustering over 15 different data sets, finding as optimal criterion functions the ones based on the cosine distance.

Spherical K-means is directly applied on the VSM representation of the collection. It consists in partitioning the N documents into K distinctive groups by minimizing the loss function $\Phi(\mathbf{U}, \mathbf{H})$:

$$\Phi(\mathbf{U}, \mathbf{H}) = \sum_{i=1}^{N} \sum_{g=1}^{K} u_{ig}(1 - cos(\mathbf{d}_i, \mathbf{h}_g)), \tag{2.5}$$

over all binary allocation matrix $\mathbf{U}$ and prototype matrix $\mathbf{H}$.
The generic entry of $\mathbf{U}$, u_{ig}, denotes the assignment of object i to cluster g such that $\sum_{g=1}^{K} u_{ig} = 1$ for all i:

$$u_{ig} = \begin{cases} 1 & \text{if } i \text{ is allocated to cluster } g, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

$\Phi(\mathbf{U}, \mathbf{H})$ is minimized if and only if

$$\mathbf{h}_g = \sum_{i=1}^{N} u_{ig} \frac{\mathbf{d}_i}{||\mathbf{d}_i||}. \tag{2.7}$$

A fuzzy extension of the objective function for spherical K-means can be easily set up by employing the membership degree matrix instead of the allocation one. Against this framework, each membership degree, u_{ig}, takes value in the interval $[0, 1]$ allowing the observations to be assigned to multiple clusters simultaneously. In Equation (2.5) u_{ig} is replaced by u_{ig}^m, for $m > 1$, which is the fuzziness parameter, usually chosen in the interval $[1.5, 2]$ (Pal & Bezdek, 1995).

In document clustering the feature vectors are usually highly sparse. Spherical K-means, through the employment of the cosine distance, can adequately capture the sparsity of the input data but the computational time of the algorithm increases as the parameter K assumes higher values. Recently, Knittel, Koch, and Ertl (2021) develop an extension of spherical K-means improving the scalability of the algorithm with respect to the parameter K by introducing a new indexing structure. The method proves to be faster than the standard version when considering sparse input vectors.

There are other common prototype-based document clustering techniques derived directly from K-means. For instance, Krishna and Murty (1999) presented Genetic K-means Algorithm (GKA) for clustering textual documents by identi-

fying a globally optimal partition. It consists in the hybridization of K-means with genetic algorithms, which are stochastic optimization algorithms.
Other commonly used methods rely on the Particle Swarm Optimization (PSO) algorithm (Eberhart & Kennedy, 1995) based, as the name suggests, on a stochastic optimization technique used to improve the problem of the initialization. In Cui, Potok, and Palathingal (2005) a new document clustering algorithm relying on PSO is discussed. It aims at discovering valuable centroids in order to minimize within-cluster distance and maximize between-cluster distance. Differently from the K-means algorithm (which is able to identify a localized optimal solution), the PSO clustering algorithm carries out the search in the whole global space, avoiding the possibility of finding sub-optimal solutions. The authors tested the validity of their approach by applying K-means, PSO and hybrid PSO on several textual data sets. The experiments highlight that more compact clustering results are generated by means of the hybrid PSO algorithm rather than the K-means.

2.3.2 Graph-based methods

Graph partitioning methods convert the data clustering problem into a graph partitioning problem (Ding, He, Zha, Gu, & Simon, 2001). In this regard, spectral methods (i.e., the methods relying on the eigenvalues decomposition of the graph matrix) are commonly used to identify the partition of the graph (Guattery & Miller, 1994).
Concerning clustering techniques, spectral clustering algorithm has been extensively used when analysing text data: for instance, an extension of this methodology is addressed in Janani and Vijayarani (2019) where the authors propose a novel spectral clustering algorithm with PSO (called Spectral Clustering PSO), in order to deal with the problem of high dimensionality and with the sub-optimal solutions that might be induced by K-means since its dependence on the initialization phase; in Kumar and Daumé (2011) spectral clustering is proposed in combination with co-training algorithm in order to manage the multi-views of the corpus coming from different sources; also, the work by Bao, Tang, Li, Zhang, and Ye (2008) describes a novel negative matrix factorization to the affinity matrix for document clustering.
This class of methods arises its popularity also because of its flexibility, which allows to identify clusters independently of their shape.

Starting from the initial data set, the basic idea behind spectral clustering methods consists in building a weighted graph: the nodes of the graph represent the documents in the collection, while each edge is weighted with the similarity between the linked nodes.

In particular, the clustering procedure consists in splitting the graph in a given number of clusters so that nodes highly connected belong to the same group.

Spectral clustering relies on the eigen-decomposition of the Laplacian matrix, **L**:

$$\mathbf{L} = \mathbf{D} - \mathbf{S}, \tag{2.8}$$

where **S** identifies the adjacency matrix and **D** the degree matrix, which is a diagonal matrix of dimensions $N \times N$ with the degrees of the nodes along the diagonal. It is common to consider the normalized version of the Laplacian matrix:

$$\mathbf{L}_{norm} = \mathbf{D}^{-\frac{1}{1}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{1}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}. \tag{2.9}$$

Given the number of clusters, K, spectral clustering consists in applying the K-means clustering algorithm on the first K eigenvectors of the normalized Laplacian matrix (the eigenvectors are commonly normalized before running the K-means).
Hence, the main idea consists in finding a low-dimensional embedding by eigen-decomposition where data are separated and can be easily clustered.

The key point in spectral clustering algorithms is the identification of an appropriate similarity measure in order to properly describe the structure of the data. In the document clustering domain, string kernel functions (Lodhi, Saunders, Shawe-Taylor, Cristianini, & Watkins, 2002) are usually adopted as similarity measures.

In this regard, string kernel functions quantify the entity of the similarity between documents by counting the number of matching substrings the documents have in common.
Formally, a substring is defined to be a sequence of l characters appearing one after the other in the text, even though not necessarily contiguously. Consider, for instance, the following 3 words: "car", "air" and "arctic". The only matching substring of length 2 shared by the three words is the sequence "a-r". As it is possible to notice, in the second word the two letters are not contiguous.

Generically, a string kernel function between two documents, $\mathbf{d}_i$ and $\mathbf{d}_q$, is given by:

$$k(\mathbf{d}_i, \mathbf{d}_q) = \sum_{\gamma \in A^*} \text{num}_\gamma(\mathbf{d}_i)\text{num}_\gamma(\mathbf{d}_q)\lambda_\gamma, \tag{2.10}$$

where A^* is the set of all strings of length l, *num* counts how many times the substrings in A^* appear in the documents $\mathbf{d}_i$ and $\mathbf{d}_q$, and λ_γ is a decay factor

associated to γ representing the weight of each matching substring in the text. The decay factor can assume different values or it can be held constant for all the matching substrings.
Different string kernels can be found in literature: *Spectrum* kernel, *Exponential* kernel and *Boundrange* kernel are some of the most commonly used functions. The first one considers only those matching substrings composed by exactly l characters. In this case, a constant value of the decay factor, λ, is used for each matching substring. The *Exponential* kernel, also known as Exponential Decay kernel, is characterized by the reduction of the decay-factor when the matching substrings get shorter. Finally, *Boundrange* kernel takes into consideration only matching substrings whose length is lower or equal to l and, depending on their sizes, it attributes to each substring a different weight.
A detailed presentation of string kernel functions can be found in Lodhi et al. (2002) and Karatzoglou and Feinerer (2007).

Some of the drawbacks of spectral clustering consist in the selection of an adequate similarity measure and the computational time which increases with the complexity of the graph.

2.3.3 Hierarchical methods

Divisive and agglomerative clustering algorithms can also be applied for text documents classification.
The former one performs successive bisections on the clusters following an iterative approach (Steinbach, Karypis, & Kumar, 2000): all the documents initially belong to a single cluster, then the approach proceeds by performing further subsequent bisections according to a certain objective function. The process continues until having N single clusters, each containing a single document.
In the agglomerative case, each observation initially represents a cluster (singleton). Then, the distance matrix (according to the employed metric) between all the singletons is build: those observations having the lowest value of the considered distance measure are merged together into a cluster. After, a new distance matrix is constructed considering all the pairwise distances between the singletons together with the newly formed cluster. This process continues until only one cluster containing all the observations is obtained (Sneath & Sokal, 1973).
A detailed review on hierarchical methods for text data is present in Zhao, Karypis, and Fayyad (2005).

With respect to divisive methods, in the study proposed by Zhao and Karypis (2004) some of the most commonly used objective functions for divisive document

clustering are analysed.
For instance, the I_1 criterion function maximizes the sum of the average of the pairwise cosine similarities calculated between the documents belonging to the same group, each one weighted with its corresponding size (Puzicha, Hofmann, & Buhmann, 2000). It is expressed as follows:

$$I_1 = \sum_{g=1}^{K} n_g \left(\frac{1}{n_g^2} \sum_{\mathbf{d}_i, \mathbf{d}_q \in C_g} cos\left(\mathbf{d}_i, \mathbf{d}_q\right) \right), \tag{2.11}$$

where C_g represents the g-th cluster of dimension n_g.

On the contrary, the E_1 criterion function executes the clustering by minimizing the cosine similarity between the centroid of each group and the centroid of the overall collection (Hart, Stork, & Duda, 2000):

$$E_1 = \sum_{g=1}^{K} n_g \ cos\left(\mathbf{h}_g, \mathbf{h}\right). \tag{2.12}$$

The vector $\mathbf{h}$ represents the centroid of the corpus and it is expressed as $\mathbf{h} = \frac{\sum_{i=1}^{N} \mathbf{d}_i}{N}$.
Another criterion function, H_1, is obtained as ratio of I_1 and E_1.

With reference to agglomerative algorithms, several approaches for computing the similarity between two groups have been developed. The most common ones for text data refer to the well-known single-linkage, complete-linkage and average-linkage schemes where the Euclidean distance is replaced by the cosine one.
The first quantifies the level of similarity of two generic clusters by calculating the maximum of the cosine distance, cos_{dist}, between the documents for each of the two clusters

$$\Phi_{single.link}(C_g, C_f) = \max_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} cos_{dist}\left(\mathbf{d}_i, \mathbf{d}_q\right). \tag{2.13}$$

On the contrary, the complete-linkage scheme selects the minimum between all the pairwise cosine distances calculated between all the documents in the two considered clusters

$$\Phi_{complete.link}(C_g, C_f) = \min_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} cos_{dist}\left(\mathbf{d}_i, \mathbf{d}_q\right). \tag{2.14}$$

Finally, the average-linkage scheme calculates the average of the pairwise cosine distances between all the observations in the two clusters.

$$\Phi_{average.link}(C_g, C_f) = \frac{1}{n_g \cdot n_f} \sum_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} cos_{dist}\left(\mathbf{d}_i, \mathbf{d}_q\right). \tag{2.15}$$

It worth noticing that the construction of the dendogram would be prohibitive for large document data sets, making these methods not suitable for analysing large collection of documents, despite their relatively ease of implementation, that do not require the knowledge of further input parameters.

2.3.4 Model-based methods

Model-based clustering methods rely on the assumption that the population is composed by a mixture of different sub-populations, each one following a certain probability distribution. Hence, a crucial point consists in identifying a mixture model that can well describe the structure of text data (sparsity, high-dimensionality). Once identified, the parameters of the model are usually estimated through the Expectation–Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), whose convergence is influenced by the initialization phase and there is no guarantee that a global optimum solution is reached. Moreover, as for prototype-based methods, it is necessary to select a priori an appropriate value for the number of mixture components K so to increase the clustering accuracy. However, model-based methods can be more representative of real case studies compared to other competitors: indeed, every document is associated with the posterior probabilities to belong to each of the K clusters, identifying automatically a soft partition.

The conditional marginal distribution of the mixture model is given by:

$$f(\mathbf{d}_i, \boldsymbol{\Psi}) = \sum_{g=1}^{K} \pi_g f_g(\mathbf{d}_i|\theta_g), \tag{2.16}$$

where $\boldsymbol{\Psi} = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ denotes the global vector of unknown parameters and $f_g(\mathbf{d}_i|\theta_g)$ are the component densities.
The parameter $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_g, \ldots, \pi_K\}$ represents the vector of prior probabilities for each mixture component (such that $\pi_g > 0 \ \forall g = 1, \ldots, K$ and $\sum_{g=1}^{K} \pi_g = 1$).
In clustering problems, the separation between clusters and the homogeneity within clusters are commonly guaranteed by taking the component densities to belong to the same parametric family $f_g(\cdot|\theta_g) = f(\cdot|\theta_g)$. The estimate of parameters is performed using the maximum likelihood approach. Since a closed-form solution is not available, the EM is adopted.

Once reached the convergence of the EM algorithm, it is possible to identify a soft partition of the documents by inspecting the posterior probabilities. The corresponding hard partition can be obtained by assigning each observation

to the corresponding cluster characterized by the highest posterior probability (maximum a posteriori rule, McLachlan et al., 2019):

$$\pi(g|\mathbf{d}_i) = \frac{\pi_g f(\mathbf{d}_i|\theta_g)}{\sum_{l=1}^{K} \pi_l f(\mathbf{d}_i|\theta_l)} \quad \forall g = 1, \ldots, K \;\; \forall i = 1, \ldots, N. \tag{2.17}$$

Also for text data, the most used model-based clustering method is the Gaussian Mixture Model (GMM) (Fraley & Raftery, 2002), where each density component follows a multivariate Gaussian distribution.

Roweis and Saul (2000) and Belkin and Niyogi (2001) have shown that image and text data are generated from a probability distribution lying on a submanifold, having lower dimensions, of the sorrounding space. Against this background, He, Cai, Shao, Bao, and Han (2010) and J. Liu, Cai, and He (2010) proposed to add, when analyzing the likelihood function of GMM, a Laplacian regularizer (Belkin, Niyogi, & Sindhwani, 2006) in order to model the underlying submanifold structure. The manifold is modelled by including in the likelihood function the structure of the graph through the nearest neighbor graph representation. Based on this idea, Laplacian regularized Gaussian mixture model (LapGMM) (He et al., 2010) and Locally consistent Gaussian mixture model (LCGMM) (J. Liu et al., 2010) have been introduced, improving the performance of GMM on text data.
In Nigam, McCallum, Thrun, and Mitchell (2000) a new algorithm based on the interaction between the EM and the naive Bayes classifier is proposed, considering both labelled and unlabelled documents. Another example of application of GMM for document clustering can be found in Lenz and Winker (2020), where the authors measure the spread of innovations, as reported in newspapers and journals, by introducing a new topic modelling algorithm: Paragraph Vector Topic Model (PVTM). PVTM employs DOC2VEC (Le & Mikolov, 2014), a text embedding technique that projects the collection of documents into a new semantic space where useful relationships between documents may be uncovered. Clustering via GMM is then applied in the new latent semantic space; successively clusters are interpreted and transformed into meaningful topics.

2.4 Concluding remarks

Most of the document clustering algorithms have been analyzed in this first chapter. We have examined the main approaches for each class of clustering algorithms: prototype-based, graph-based, hierarchical and model-based. First, a critical review on the main steps of the document clustering process has been carried out: special attention is given to the mathematical representation of doc-

uments, taking into consideration the pre-processing phase, and the different term-weighting schemes used in the construction of the VSM.
We have discussed the main characteristics of the most used clustering algorithms for text data for every of the aforementioned categories: spherical K-means for prototype-based methods, spectral clustering in combination with string kernel functions for graph-based methods, divisive and agglomerative algorithms with different criterion functions for hierarchical methods and GMM for model-based methods. Furthermore, starting from the above proposals, we have also considered more advanced methods such as, for instance, the ones based on GA and PSO.
Given the increasing amount of text data, document clustering methodologies became an essential tool in statistical analysis: exploring the latest works would provide a valuable direction for research in text clustering.

Chapter 3

A novel fuzzy spectral clustering approach for text data

The aim of the work described in this chapter consists in introducing new methods for unsupervised classification of document data sets based on spectral clustering.
More specifically, a novel fuzzy spectral clustering algorithm is presented (see, also, Cozzolino, Ferraro, and Winker (2021)). The new method is used in combination with Spectrum string kernel function and *Kernel and Set Similarity* (KS^2M), resulting in two novel fuzzy spectral clustering algorithms for text data. However, in order to overcome their drawbacks, a (third) novel fuzzy spectral clustering algorithm is also introduced.
Indeed, given the inherent sequential nature of text data, the proposed algorithm is characterized by the employment of a novel similarity measure, which is also described in this chapter.
The new metric exploits the ordered position of the characters within the text (represented as an ordered sequence of items), as well as the overall similarities between the documents in the whole corpus. The proposed similarity measure is used for the construction of the Laplacian matrix, which corresponds to the object of input for the new document clustering algorithm.
The validity of the proposed approaches has been tested on both benchmark and real data sets.

3.1 Introduction

Following the direction already outlined in Chapter 2, clustering techniques are usually directly applied to the VSM representation of the corpus.
As previously introduced, among the different clustering methods applied to the VSM, K-means (MacQueen, 1967) is probably the most popular one. Another widely used clustering algorithm for document data sets is the Hyper-Spherical

K-means (Rodrigues & Sacks, 2004), where the cosine distance, rather than the Euclidean one, is employed to better represent the structure of the data.

Despite the VSM is commonly used to represent the corpus in the field of document clustering, it has some drawbacks; the most important ones follow.

- The curse of dimensionality problem.
  Each document is composed by many words and in the document vector representation each word is considered as a dimension. Against this background, for large data collection, clustering algorithms can not always manage the dimensional space efficiently (differently from what happens in small data sets).
  Another related issue is represented by the scalability: indeed, many clustering techniques perform well on small data sets but are inefficient when dealing with large collection of data.

- The selection of an adequate term-weighting scheme is another drawback in document clustering techniques, as it has already been discussed in Chapter 2.

- The selection of adequate coefficients.
  In this context, three main classes of coefficients might be taken in consideration: distance coefficients, association coefficients and probabilistic coefficients.

  1. Distance coefficients: Euclidean distance, for instance, has been used very extensively in cluster analysis. However, one of its main limits when analysing text data concerns the fact that it can lead to consider two documents to be highly similar even if they do not have common words.

  2. Association coefficients: these coefficients take into consideration the number of terms shared by each different pair of documents. Consequently, normalization becomes an essential element to handle documents of different sizes.

  3. Probabilistic coefficients: the idea behind the use of these coefficients for the identification of clusters is that the documents in each cluster are characterized by a high probability of being jointly relevant to a query.

The employment of string kernel functions (Lodhi et al., 2002) can be seen as an alternative form of "quantification" of the texts.
String kernels have been initially employed in text classification together with Support Vector Machines.

In this regard, among the different competitors, graph-based methods combined with string kernel functions have attracted special consideration in text classification, proving to perform well on text data. However, as far as we are concerned, very little has been done in the field of unsupervised classification of document data sets.
The study of Karatzoglou and Feinerer (2007) represents one of the main works in this field, where the authors compare the standard K-means and spectral clustering on benchmark text data sets, showing how spectral clustering method outperforms the former one in terms of recall rate.

However, it is worth noticing that despite the effectiveness of spectral clustering together with string kernels on the categorization of documents, the works available in literature consider mainly the crisp (hard) approach to clustering, with the risk to be unrealistic when analysing true data sets characterized by documents sharing similar characteristics.
Moreover, when employing spectral clustering algorithms, it is fundamental to select an adequate similarity measure that can properly describe the structure of the data points. Given the particular sequential nature of text documents, it becomes necessary to further investigate on this point in order to improve the clustering results.

These considerations motivate us to introduce a novel fuzzy spectral clustering algorithm for text data: contrary to the crisp approach, where each observation (document) can be assigned to only one cluster, the fuzzy approach proves to easily identify important relationships between documents since each observation can be assigned to more than one cluster simultaneously. The proposed fuzzy extension is used in different settings: with the Spectrum string kernel function and with *Kernel and Set Similarity* (KS^2M), already available in literature.
Furthermore, the proposed algorithm is used in combination with a novel similarity measure, $\mathbf{S}^*$, which is able to better uncover both the sequential and non-sequential essence of text data, returning encouraging results.

3.2 Spectral clustering and kernels for text sequences

An introduction of the most important concepts behind spectral clustering and string kernel functions is discussed in this section.

3.2.1 Spectral clustering

According to the work of Von Luxburg (2007), spectral clustering methods have their foundation in spectral graphs theory. Within this framework, the main idea consists in building a graph from the sample of observations so that every data point is represented by a node and the edges connecting the nodes quantify the level of similarity between the considered observations. In other words, if the only form of information available is provided by the level of similarity between data points, the starting point for representing the data consists in building a similarity graph $G = (V, E)$, such that V is a finite non empty set and E is a binary correspondence on V. The symbol V represents the so-called *vertex set* of the graph and its members are the vertices. The symbol E identifies the *edge set* of the graph and its components are the edges.

Let consider N observations identified by the vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ in $\mathbb{R}^T$. Each vertex v_i in the graph represents a data point $\mathbf{x}_i$.
Then, the pairwise similarities between data points, $s_{iq} = s(\mathbf{x}_i, \mathbf{x}_q)$ for $i \neq q = 1, \ldots, N$ are calculated by some function, s, which is non-negative and symmetric. The associated similarity matrix is denoted by $\mathbf{S} = [s_{iq}]_{i,q=1,\ldots,N}$.
If the similarity s_{iq}, with $i \neq q$ for $i = 1, \ldots, N$, is positive or larger than a certain threshold, the corresponding vertices v_i and v_q are connected and the edge is weighted by s_{iq}.

Against this background, the clustering process consists in identifying a partition of the aforementioned graph by recognizing the existence of sub-graphs containing sets of observations characterized by a high level of similarity. In other words, the aim consists in finding a partition of the graph characterized by the fact that the edges connecting different groups are associated to low weights (meaning that the data points assigned to different clusters are dissimilar from each other). On the contrary, the edges connecting endpoints in the same group should be associated to high weights (meaning that the data points in the same cluster are similar to each other). In order to formalize these concepts, some basic graph notation is introduced.

Notation used for the graphs

The notation $G = (V, E)$ identifies an undirected graph characterized by the vertex set $V = \{v_1, ..., v_N\}$. Let assume that G is a weighted undirected graph, meaning that associated to each edge connecting two vertices v_i and v_q there is a non-negative weight denoted as $\omega_{iq} = s_{iq}$. The corresponding weighted adjacency matrix is given by $\mathbf{\Omega} = \mathbf{S} = [s_{iq}]_{i,q=1,...,N}$. If $\omega_{iq} = 0$, then v_i and v_q do not have an edge in common. From the definition of undirected graph follows that $\omega_{iq} = \omega_{qi}$.

The degree δ_i of a vertex v_i is defined as the sum of all the similarity values to all nodes in the graph, i.e:

$$\delta_i = \sum_{q=1}^{N} \omega_{iq}. \tag{3.1}$$

$\mathbf{D}$ is known as *degree matrix*: it is a diagonal matrix whose elements on the diagonal, $\delta_1, \ldots, \delta_N$, are the degrees of each node.

The complement of a given set of vertices, $A \subset V$, is denoted as $\overline{A}$. The symbol $\mathbf{1}_A = (f_1, \ldots, f_N)'$ identifies whether or not a vertex belongs to A: $f_i = 1$ if $v_i \in A$ and $f_i = 0$ otherwise.
Given two subsets of vertices, A and B, the corresponding "part" of the weighted adjacency matrix is given by:

$$\mathbf{\Omega}(A, B) = \sum_{i \in A; q \in B} \omega_{iq}. \tag{3.2}$$

To conclude, the sets $A_1, \ldots, A_K$ identify a partition of the graph if the two following conditions are satisfied: $A_1 \cup \cdots \cup A_K = V$ and $A_{g_1} \cap A_{g_2} = \emptyset$ for $g_1 \neq g_2$ and $g_1, g_2 = 1, \ldots, K$.

Weighted graphs: the different versions available in literature

Starting from a given a set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with pairwise similarities s_{iq}, $i \neq q$ for $i = 1, \ldots, N$, there are three main ways to build a weighted graph. Indeed, the key point consists in choosing how to measure the relationships between data points. A description of the main weighted graphs follows.

1. The ε-neighborhood graph: for this kind of graph, instead of the pairwise similarities between data points, the pairwise distances are taken into consideration. In particular, the data points whose pairwise distances are smaller than a certain threshold, denoted as ε, are connected. For this specific graph, weighting the edges do not add any additional information to the graph since all the connected data points are approximately of the same

scale. Consequently, this kind of graph is not a "standard" weighted graph but it can be mainly considered as an unweighted graph (Von Luxburg, 2007).

2. The k-nearest neighbor graph: the objective behind the construction of this graph is to connect two vertices, v_i with v_q, if within the k-nearest neighbors of v_i there is the vertex v_q. The drawback of this definition is that the outcome results in a direct graph, since the neighborhood relationship lacks the property of symmetry. However, two methods are available to transform a direct graph to an undirect one.
   The first method simply ignores the directions of the edges. In other words, two vertices v_i and v_q are connected if v_i is one of the k-nearest neighbors of v_q or, alternatively, if v_q is one of the k-nearest neighbors of v_i. This procedure leads to an undirect graph which is commonly called the k-nearest neighbor graph.
   The second method considers the mutual k-nearest neighbors of both vertices. Indeed, two vertices v_i and v_q are connected if v_i is one of the k-nearest neighbors of v_q and, at the same time, v_q should be one of the k-nearest neighbors of v_i. We refer to this graph as the mutual k-nearest neighbor graph.
   In both methods, the edges connecting the corresponding vertices are weighted by the corresponding pairwise similarities between endpoints.

3. The fully connected graph: this graph is characterized by a relatively easy construction compared to the ones analysed above. Indeed, all the points having positive similarities between each other are connected. All the edges connecting the endpoints are weighted by $\omega_{iq} = s_{iq}$.

All the typologies of graphs described above are employed in spectral clustering algorithms.

Laplacian matrix

The key element for applying spectral clustering is the identification of the Laplacian matrix. Spectral graph theory is the research field dedicated to the analysis and the investigation of this kind of matrices. In the work of Chung (1997) is possible to find a complete review of the most important methods together with their most important properties.

Let G be an undirected, weighted graph with weight matrix $\mathbf{\Omega} = \mathbf{S}$, where $\omega_{iq} = s_{iq} \geq 0$.
As already seen in Chapter 2, the unnormalized Laplacian matrix, $\mathbf{L}$, corresponding to the matrix representation of a weighted graph, is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{\Omega}. \tag{3.3}$$

The works of Oellermann and Schwenk (1991) and Mohar (1997) contain complete and detailed reviews of its main properties.

The following propositions, taken from the work of Von Luxburg (2007), summarize some important results of Laplacian matrices. For a complete review and for the proofs, refer to Von Luxburg (2007).

Preposition 1 *The matrix $\boldsymbol{L}$ satisfies the following properties:*

1. *$\boldsymbol{L}$ is symmetric and positive semi-definite.*
2. *The smallest eigenvalue of $\boldsymbol{L}$ is 0, the corresponding eigenvector is the constant one vector.*
3. *$\boldsymbol{L}$ has N non-negative, real-valued eigenvalues $0 = \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_N$.*

Preposition 2 *Let G be an undirected graph with non-negative weights. Then the multiplicity K of the eigenvalue 0 of $\boldsymbol{L}$ equals the number of connected components $A_1, \ldots, A_K$ in the graph.*

A connected component of an undirect graph, G, is defined as a connected subgraph having no connections between its vertices and the vertices of the rest of the graph. Thus, the matrix $\mathbf{L}$ has as many eigenvalues 0 as there are connected components (Von Luxburg, 2007); in easier words, the graph G has K connected components if its Laplacian matrix, $\mathbf{L}$, has K blocks.

Usually, the normalized version of the Laplacian matrix is taken into consideration. There are two different versions of normalized Laplacian matrix:

$$\mathbf{L}_{norm.sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{\Omega}\mathbf{D}^{-\frac{1}{2}}, \tag{3.4}$$

$$\mathbf{L}_{norm.rw} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{\Omega}. \tag{3.5}$$

The first matrix, $\mathbf{L}_{norm.sym}$, is a matrix characterized by the property of symmetry; instead the second one, $\mathbf{L}_{norm.rw}$, is associated to a random walk. In the following sections we will refer to $\mathbf{L}_{norm.sym}$. The work of Chung (1997) represents one of the main references for deepening the study of normalized Laplacian

matrices.

Similarly to Preposition 1 and Preposition 2, we have:

Preposition 3 *Normalized Laplacian matrix satisfies the following properties:*

1. *σ is an eigenvalue of $\boldsymbol{L}_{norm.rw}$ with eigenvector u if and only if σ is an eigenvalue of $\boldsymbol{L}_{norm.sym}$ with eigenvector $\omega = \boldsymbol{D}^{\frac{1}{2}}u$.*

2. *0 is an eigenvalue of $\boldsymbol{L}_{norm.rw}$ with constant one vector as eigenvector if and only if 0 is an eigenvalue of $\boldsymbol{L}_{norm.sym}$ with eigenvector $\boldsymbol{D}^{\frac{1}{2}}\mathbf{1}$.*

3. *$\boldsymbol{L}_{norm.sym}$ is semi-definite positive and has N non-negative, real-valued eigenvalues $0 = \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_N$.*

As previously seen for unnormalized Laplacian matrices, also for normalized Laplacians the multiplicity of the eigenvalue 0 is associated to the number of connected components, as it is stated in Preposition 4 that follows.

Preposition 4 *Let G be an undirected graph with non-negative weights. Then the multiplicity K of the eigenvalue 0 of $\boldsymbol{L}_{norm.sym}$ equals the number of connected components $A_1, \ldots, A_K$ in the graph.*

Spectral clustering

Spectral clustering has two main versions: one from Shi and Malik (2000) and the other one from Ng, Jordan, and Weiss (2001). The former one employs the eigenvectors of the normalized Laplacian $\mathbf{L}_{norm.rw}$ while, on the contrary, the latter one uses the matrix $\mathbf{L}_{norm.sym}$.
In this work we are going to focus our attention on the second version.

According to Ng et al. (2001), the spectral clustering algorithm with the normalized Laplacian $\mathbf{L}_{norm.sym}$ proceeds as follows.

1. Build the weighted similarity graph, G, according to one of the methods previously described. Consider $\boldsymbol{\Omega}$ to be the corresponding weighted adjacency matrix.

2. Calculate $\mathbf{L}_{norm.sym}$, as the normalized Laplacian matrix.

3. Compute the first K eigenvectors of $\mathbf{L}_{norm.sym}$.

4. Build a new matrix containing the eigenvectors of Step 3 as column vectors.

5. Normalize the rows of the matrix at Step 4 in order to have norm equals to 1.

6. In the end, identify K clusters of data points through the K-means clustering algorithm.

In the spectral clustering algorithm the main "stratagem" consists in changing the original representation of data points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^T$ into new points, say $\mathbf{y}_i$ with $i = 1, \ldots, N, \in \mathbb{R}^K$. This trick allows for a better identification of the clusters, since the new "coordinates" representing the original data points should better capture the cluster-properties in the data.

The identification of the appropriate number K of clusters is a common problem for every clustering algorithm. Against this background and in order to overcome this issue, a large variety of methods have been illustrated in literature: within-cluster and between-cluster similarity measures, ad-hoc internal and external cluster validity indexes, information-theoretic criteria and so on (Still & Bialek, 2004). The cluster validity indexes used in our work are fully described in the following sections.

Spectral clustering as a graph partitioning problem

The objective behind the clustering procedure consists in partitioning the data points into several clusters according to their similarities. Against this background, the problem of spectral clustering can also be seen as an approximation of a graph partitioning problem.

The easiest way to identify a partition in the graph consists in solving the *mincut* problem. Let G be a similarity graph with adjacency matrix $\mathbf{\Omega}$ and let K be the given number of clusters. The *mincut* method identifies the partition $A_1, \ldots, A_K$ determining the smallest "cut", i.e. the smallest "boundary" (or sum of edge weights, more generally):

$$cut(A_1, \ldots, A_K) = \frac{1}{2} \sum_{g=1}^{K} \mathbf{\Omega}(A_g, \overline{A}_g). \tag{3.6}$$

As reported in the work of Stoer and Wagner (1997), the *mincut* problem solution is relatively easy and efficient for $K = 2$. However, in practical examples, it is not exempt from drawbacks. Indeed, very often it leads to unbalanced partitions since it simply divides one single vertex from the remaining graph. In

many practical applications, this form of classification is unrealistic. A possible solution consists in choosing other objective functions having the "constraint" that the sets $A_1, \ldots, A_K$ contain a reasonable number of data points. In particular, the ratio cut (Hagen & Kahng, 1992) and the normalized cut (Shi & Malik, 2000), denoted as *RatioCut* and *Ncut* respectively, provide a step forward in the right direction.

The size of the subsets $A_1, \ldots, A_K$ is measured differently according to which objective function is used: *RatioCut* takes into consideration the number of vertices, $|A|$; on the other hand, the *Ncut* function considers the weights of the edges, $vol(A)$, corresponding to the sum of the degrees δ_i of all vertexes v_i in the sets $A_1, \ldots, A_K$.
Both objective functions identify the partition $A_1, \ldots, A_K$ minimizing:

$$RatioCut(A_1, \ldots, A_K) = \frac{1}{2} \sum_{g=1}^{K} \frac{\mathbf{\Omega}(A_g, \overline{A}_g)}{|A_g|}, \tag{3.7}$$

$$Ncut(A_1, \ldots, A_K) = \frac{1}{2} \sum_{g=1}^{K} \frac{\mathbf{\Omega}(A_g, \overline{A}_g)}{vol(A_g)}. \tag{3.8}$$

In particular, the two functions assume small values if the clusters A_g are "big enough". If all $|A_g|$ are coincident, then the minimum of $\sum_{g=1}^{K}(1/|A_g|)$ is obtained; on the other hand, if all $vol(A_g)$ coincide, then the minimum of $\sum_{g=1}^{K}(1/vol(A_g))$ is achieved.
RatioCut and *Ncut* functions lead to balanced partitions (expressed in terms of number of vertices or edge weights). However, the introduction of the balancing conditions determines an increase of the computational complexity of the *mincut* problem, becoming NP hard. Refer to Wagner and Wagner (1993) for further details.

With spectral clustering is possible to solve the relaxed versions of these problems: relaxing *Ncut* produces the normalized spectral clustering; on the other hand, the relaxation of *RatioCut* leads to the unnormalized spectral clustering.

3.2.2 String kernel functions

It appears clear from the previous section that for spectral clustering methods it is very important to select adequately a similarity metric, s, in order to properly build the Laplacian matrix.
Since we are dealing with text data, it is required some kind of similarity measures between strings. In natural language processing, strings are sequences of

alphabet characters and represent text in natural language.
Recently, given the arise of kernel-based methods for pattern analyses (Smola & Schölkopf, 1998; Shawe-Taylor, Cristianini, et al., 2004) and classification techniques like SVM, string kernel functions are usually adopted. Many string kernels with different specificities have been proposed in literature. Indeed, kernel functions naturally induce a measure of similarity. In text mining applications they evaluate the level of similarity between two documents by means of matching sequences of l characters they contain (where l is a free parameter that should be appropriately chosen). In other words, string kernel functions calculate the amount of matching substrings of length l shared by the documents: more similar two documents are, then more matching substrings of length l they share.

The works of Smola and Schölkopf (1998) and Shawe-Taylor et al. (2004) provide a complete overview of the theory beyond kernel methods.
Let $\mathcal{X}$ be an input space, a (positive semi-definite) kernel is defined as a function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, with $\mathcal{K}(x, y) = \mathcal{K}(y, x)$, such that for any $\{c_i\}_{i=1}^N$ and $\{x_i\}_{i=1}^N$:

$$\sum_{i=1}^{N} \sum_{q=1}^{N} \{c_i\} \{e_q\} \mathcal{K}(x_i, x_q) \geq 0. \tag{3.9}$$

Given a set of data points $\{x_i\}_{i=1}^N$, the $N \times N$ matrix $\boldsymbol{\mathcal{K}} = [k_{iq}] = [\mathcal{K}(x_i, x_q)]$ is denoted as *gram matrix*.

Mercer's theorem states that $\mathcal{K} = \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is a (positive-definite) kernel if an only if there is a feature space $\mathcal{F}$, provided with an inner product $\langle \cdot \ , \ \cdot \rangle$, and a map $\Phi_{map} : \mathcal{X} \longrightarrow \mathcal{F}$, satisfying for all $x, y \in \mathcal{X}$:

$$\mathcal{K}(x, y) = \langle \Phi_{map}(x) \ , \ \Phi_{map}(y) \rangle. \tag{3.10}$$

Even the VSM, which is commonly used in information retrieval, can be subject to a form of "kernelization", by simply considering the input space $\mathcal{C} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$, the feature space $\mathbb{R}^T$ and the feature map $\Phi_{map} = \mathbf{d}/||\mathbf{d}||$. The ad-hoc combination of these elements give rise to the Joachims' BOW kernel representation (Joachims, 1998):

$$\mathcal{K}(\mathbf{d}_1, \mathbf{d}_2) = \langle \Phi_{map}(\mathbf{d}_1) \ , \ \Phi_{map}(\mathbf{d}_2) \rangle = \frac{\langle \mathbf{d}_1 \ , \ \mathbf{d}_2 \rangle}{||\mathbf{d}_1|| \cdot ||\mathbf{d}_2||}, \tag{3.11}$$

which corresponds to the cosine measure:

$$cos(\mathbf{d}_1, \mathbf{d}_2) = \frac{\langle \mathbf{d}_1 \ , \ \mathbf{d}_2 \rangle}{||\mathbf{d}_1|| \cdot ||\mathbf{d}_2||}. \tag{3.12}$$

There are many kernel functions for strings with various applications. One of

the most natural ways to measure the similarity between two strings is to count how many substrings of fixed length l the two strings have in common. This corresponds to the Spectrum string kernel.
Getting back to the definition introduced in Chapter 2, given two strings x' and y', the Spectrum kernel is defined as:

$$\mathcal{K}_l(x', y') = \sum_{\gamma \in A^*} |x'|_\gamma \cdot |y'|_\gamma \cdot \lambda, \tag{3.13}$$

where λ is the constant factor used to weight the matching substrings of length l, whose default value is 1.1 (Karatzoglou, Smola, Hornik, & Karatzoglou, 2019).

3.3 A new proposal of fuzzy spectral clustering algorithm with string kernels

In the crisp version of spectral clustering, K-means algorithm is used to classify the documents according to the normalized eigenvectors of the normalized Laplacian matrix.
The novel fuzzy extension of spectral clustering proposed in this work can be developed in a straightforward way from the hard one by employing the fuzzy K-medoids, instead of the K-means, when analysing the eigenvectors of $\mathbf{L}_{norm.sym}$.

The fuzzy K-medoids consists in the following minimization problem:

$$\begin{aligned} \min_{\mathbf{U},\,\mathbf{H}} \quad & J = \sum_{i=1}^{N}\sum_{g=1}^{K} u_{ig}^m d^2(\mathbf{y}_i, \mathbf{h}_g), \\ \text{s.t.} \quad & u_{ig} \in [0,1] \quad \forall i = 1,\dots,N, \quad \forall g = 1,\dots,K, \\ & \sum_{g=1}^{K} u_{ig} = 1 \quad i = 1,\dots,N, \end{aligned} \tag{3.14}$$

with respect to the membership degree matrix, denoted as $\mathbf{U}$, and the medoids matrix, denoted as $\mathbf{H}$. The symbol d identifies the Euclidean distance.
Differently from the fuzzy K-means (Bezdek, 1981), in the fuzzy K-medoids the prototypes (medoids) of the clusters are directly observable within the sample of observations, $\{\mathbf{h}_g, g = 1,\dots,K\} \subseteq \{\mathbf{y}_i, i = 1,\dots,N\}$. On the contrary, in the former algorithm the prototypes (centroids) are calculated as weighted means and hence they do not qualify as observed objects.

The generic element u_{ig} is representative of the membership degree of the object i to the cluster g and ranges from 0 to 1, identifying different levels of

membership.

The parameter that controls the fuzziness of the partition is m: for m assuming high values all the memberships are set equal, while for m assuming values close to 1 the hard Partitioning Around Medoids (PAM) algorithm (Kaufman & Rousseeuw, 2009) is obtained. The default value for fuzzy algorithms is usually selected in the interval $(1, 2]$. However, for the fuzzy K-medoids algorithm, it is recommended to select m in the interval $(1, 1.5]$ since the prototypes (medoids) have always a membership of one to their corresponding clusters, thus they do not have sensitivity when raising them up to the m-power. Consequently, when m assumes high values, the flexibility of medoids may become slower from iteration to iteration.

The solution of the minimization problem of the fuzzy K-medoids can be obtained by means of an iterative algorithm where at each iteration r the membership degree matrix $\mathbf{U}^{(r)}$ is updated, keeping fixed $\mathbf{H}^{(r-1)}$, by means of

$$u_{ig} = \frac{1}{\sum_{g'=1}^{K} \left(\frac{d^2(\mathbf{y}_i, \mathbf{h}_g)}{d^2(\mathbf{y}_i, \mathbf{h}'_g)} \right)^{\frac{1}{m-1}}}. \tag{3.15}$$

At the same iteration, the medoids matrix $\mathbf{H}^{(r)}$ is updated, keeping fixed $\mathbf{U}^{(r)}$, by using:

$$q = \text{argmin}_{i'=1}^{N} \sum_{i=1}^{N} u_{ig}^m d^2(\mathbf{y}_i, \mathbf{y}_{i'}) \qquad g = 1, \ldots, K; \mathbf{h}_g = \mathbf{y}_q. \tag{3.16}$$

The algorithm ends when is reached the convergence condition.

The standard fuzzy K-means is usually less robust than the fuzzy K-medoids. Indeed, the fuzzy K-means algorithm can be easily influenced by noisy data and outliers, because these elements have a direct impact on the calculation of the prototypes. On the contrary, by using the medoids instead of prototypes, it could be possible to partially eliminate such drawbacks.

In the following section, an illustrative example on the performance of the fuzzy version of spectral clustering with Spectrum string kernel is presented.

3.3.1 An application of the fuzzy version of spectral clustering algorithm with Spectrum string kernel function

This section introduces some preliminary results on the fuzzy spectral clustering algorithm combined with Spectrum string kernel on Reuters-21578 data set (Lewis, 1997), that will be also used for other analyses that follow.

Reuters-21578 contains stories for the Reuters news agency and it is publicly available; it is currently one of the most extensively used data sets for the classification of text files. It includes 12902 documents for 90 classes.
Given the high number of documents and since this data set is characterized by a very skewed class distribution, we follow the Sebastiani convention (Sebastiani, 2002) considering only a specific subset of documents which is called R8. This particular set considers the first 8 categories in terms of their sizes, i.e. *acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship*, *trade*, for a total of 7674 documents.

To illustrate the advantages of the fuzzy approach, the classes *crude* vs. *money-fx* and *trade* vs. *ship* are taken into consideration. These four classes are characterized by relatively small sample sizes (*crude* and *money-fx* are composed by 374 and 293 documents; *trade* and *ship* have, respectively, 326 and 144 documents) allowing for a clear graphical representation.
In this context, the decision to perform two separated double classifications is exclusively related to a better graphical representation of the clustering results, that would otherwise be poorly distinguishable in a multi classification setting.
To conclude, the composition of the classes in each clustering problem follows the alphabetic order.

As we have illustrated, the very first step in document clustering problems is the pre-processing phase.
In the pre-processing phase, the punctuation signs, the numbers and the words with no semantic meaning (e.g. articles, pronouns, adverbs,...) are removed; the terms have been lower-cased and is carried out the stemming of the remaining words through the Porter's Stemmer algorithm (Porter, 1980).

With reference to the Spectrum string kernel function, and in order to learn more about the influence of the length parameter l on the clustering results, the algorithm is run over a range of values: from $l = 3$ to $l = 8$. For each value of l we let the membership parameter, m, vary from 1.1 to 2. Moreover, multiple random starts are used in order to limit the risk of hitting local optima.

Fuzzy K-medoids algorithm is then used to cluster the normalized eigenvectors of the normalized Laplacian matrix.

The performance of the clustering algorithm is evaluated by means of the average fuzzy Silhouette index (F.SIL) (Campello & Hruschka, 2006) and the fuzzy adjusted Rand index (F.ARI) (Campello, 2007).
The former one is an internal validation measure relying only on information in the data; it evaluates the goodness of the clustering structure without considering external information. If it tends to 1, then the observations are well assigned to the corresponding clusters.
The fuzzy Silhouette index for the partition characterized by K clusters is calculated as follows:

$$\text{F.SIL}(K) = \frac{\sum_{i=1}^{N} (u_{ig} - u_{ig'})^{\alpha} \, \text{SIL}_i(K)}{\sum_{i=1}^{N} (u_{ig} - u_{ig'})^{\alpha}}, \tag{3.17}$$

where u_{ig} and $u_{ig'}$ are, respectively, the first and the second largest membership degrees of the i-th observation of the fuzzy partition matrix; α is the weight that usually assumes value equals to 1. The last element, $\text{SIL}_i(K)$, is the crisp Silhouette index for the i-th observation and it is defined as:

$$\text{SIL}_i(K) = \frac{\beta_i - \tau_i}{max(\beta_i, \tau_i)}, \tag{3.18}$$

where τ_i is the average distance of observation i to all the other observations belonging to the same cluster and β_i is the minimum average distance of observation i to all observations belonging to another cluster. The crisp Silhouette index ranges in the interval $[-1, 1]$.
As it is possible to observe, the fuzzy extension of the Silhouette index integrates the membership degrees with the Silhouette values by calculating a weighted mean such that each individual Silhouette value is associated to a weight corresponding to the difference between the two highest fuzzy membership values of the associated point. The optimal value for K is obtained by maximizing $\text{F.SIL}(K)$.

The other cluster validity measure adopted in this experiment, the fuzzy adjusted Rand index, is an external validation measure which is used when the "true" cluster labels are known in advance. It ranges in the interval $[0, 1]$.
The crisp version of the adjusted Rand index (ARI) is defined as the proportion of the correctly classified observations over the entire sample:

$$ARI(K) = \frac{P + Q}{(P + Q + R + E)}, \tag{3.19}$$

where the terms P and Q are the classification agreements, whereas the terms R and E are the classification disagreements.
The corresponding fuzzy extension is obtained by rewriting the original crisp formulation in a fully equivalent set-theoretic form. In the fuzzy case, the above sets are then converted into fuzzy sets. Since the construction of the fuzzy set-theoretic form does not resolve in a single step, refer to Campello (2007) for the details.

The cluster validity indexes return, for *trade* vs *ship*, $l = 6$ and $m = 1.4$ as the optimal hyper-parameters; while for *crude* vs *money-fx* the optimal corresponding values for the hyper-parameters are, respectively, $l = 5$ and $m = 1.5$. The values for both the indexes are reported in Table 3.1.
Table 3.2 and Table 3.3 are agreement tables between the known partition and the corresponding hard partitions determined by the fuzzy clustering algorithm. In this context, in order to evaluate the clustering results using the external information available, the fuzzy partitions are converted into hard partitions by assigning each object to the cluster characterized by the highest membership degree.

Table 3.1: Cluster validity indexes for both the experiments: fuzzy Silhouette and fuzzy adjusted Rand index.

Categories	F.SIL	F.ARI
trade vs *ship*	0.86	0.93
crude vs *money-fx*	0.89	0.92

Table 3.2: Agreement table of *trade* vs *ship* by fuzzy spectral clustering with Spectrum string kernel.

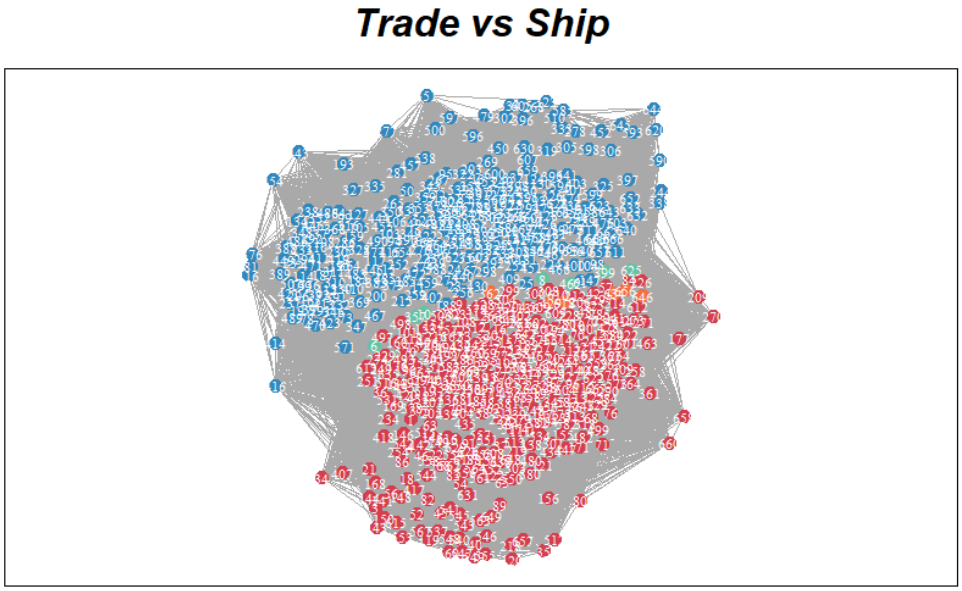
Category	Cluster 1	Cluster 2	Cases
ship	138	6	144
trade	5	321	326
Total	143	327	470

Table 3.3: Agreement table of *crude* vs *money-fx* by fuzzy spectral clustering with Spectrum string kernel.

Category	Cluster 1	Cluster 2	Cases
crude	363	11	374
money-fx	1	292	293
Total	364	303	667

The performance of the fuzzy spectral clustering algorithm is consistent with the corresponding hard version but, whilst in the hard case the obtained membership degrees were either 1 or 0, highlighting a clear assignment of the objects to the clusters, in the fuzzy approach the objects belong to both clusters with different degrees.
Analysing the corresponding hard partitions returned by the fuzzy spectral clustering algorithm, it is possible to notice that the objects that are characterized, in the fuzzy setting, by membership degrees taking values in the interval $[0.5, 0.7]$ are assigned to the other cluster compared to the original partition.
On the other hand, the objects assigned to the same cluster of the original partition are characterized, in the fuzzy setting, by higher membership degrees ranging in the interval $[0.7, 1]$.
This result can be seen, somehow, as an element of coherence between the role of the membership degrees in the fuzzy spectral clustering algorithm and the original crisp classification of the data.

In this context, adjacency matrix and Laplacian matrix are commonly used representations for weighed graph (Janani & Vijayarani, 2019). In Figure 3.1 are reported the Laplacian graphs for both examples, highlighting the assignment of objects to clusters.



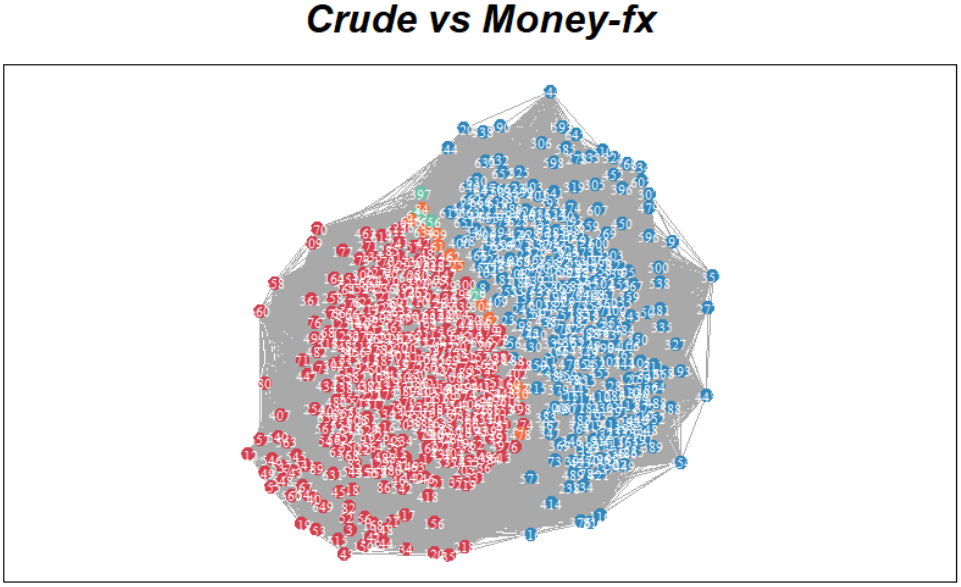


Figure 3.1: The fuzzy spectral clustering representation with Spectrum string kernel for both the experiments; red and blue points denote objects assignments to Cluster 1 and Cluster 2, respectively, with membership degrees higher than 0.70. In particular, for each cluster, light colours (orange and jade green) denote membership degrees in the intervals $[0.50, 0.70)$.

3.4 The novel fuzzy spectral clustering with *Kernel and Set similarity* (KS^2M)

Text data can be understood as sequences of characters appearing in documents. Indeed, the position of a set of characters, as well as their length, can identify a word and in text mining applications these last ones represent the main units on which performing further analyses.
Hence, it is clear that the position of characters within the text is of relevance to understand the inherent sequential structure of the data. Thus, a characteristic of sequential data, such as text data, consists in the disposition of the items within a sequence.
In other words, text data can be recognized as a sequence of elements occurring one after the other one, where the order of the characters matters.

Against this background, Tripathy et al. (2019) introduce a novel similarity measure for sequential data called *Kernel and Set Similarity* (abbreviated as KS^2M), consisting of two different parts: the first part assesses the composition of the set (set similarity) whilst the second part quantifies the sequential aspect (sequence similarity), which corresponds to the amount of similarity considering the order of the items within two different sets.

Given two different sets, a and b, KS^2M has the following form:

$$KS^2M(a,b) = p \cdot J(a,b) + (1-p) \cdot SK(a,b), \tag{3.20}$$

where $p \in [0,1]$ and $1-p$ represent the relative importance attributed to the two components; $J(a,b)$ is the measure of set similarity represented by the Jaccard similarity index (Jaccard, 1901), defined as the ratio between the number of common substrings of characters in set a and set b and the number of unique substrings in the two sets:

$$J(a,b) = \frac{|a \cap b|}{|a \cup b|}. \tag{3.21}$$

$SK(a,b)$ is a sequential similarity measure represented by a generic string kernel function. As proven in Tripathy et al. (2019), KS^2M satisfies the properties of non negativity, symmetry and normalization and hence qualifies as a proper similarity metric.

However, KS^2M has a main disadvantage when comparing two sets containing a different number of items. Indeed, the Jaccard similarity produces poor results as it is illustrated in the following example.

In each line are disposed two sets of words, characterized by different lengths, that resemble two different documents:

1. $\mathcal{A}$: $\{''cat'','' dog'','' bird'','' mouse''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower''\}$

2. $\mathcal{A}$: $\{''cat'','' dog'','' bird''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower'','' person''\}$

3. $\mathcal{A}$: $\{''cat'','' dog''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower'','' person'','' house''\}$

It is worth noticing that in all the above sentences the sets have always two words in common.
In this context, the values of the Jaccard similarity for all the three situations is always equal to 0.33.

The just outlined similarity measure, KS^2M, is used in combination with the previously introduced fuzzy spectral clustering algorithm in combination with fuzzy K-medoids, resulting in a new extension of fuzzy spectral clustering. Examples of applications are provided in Section 3.7.

3.5 The novel fuzzy spectral clustering with a new similarity measure

The following pages introduce an additional version of fuzzy spectral clustering algorithm combined with a new proposed similarity measure for text data.
Indeed, the Jaccard index lacks the ability to capture the different degrees of similarities between documents.

The novel fuzzy spectral clustering method is mainly based on the proper combination of two different elements:

1. the adoption of a new similarity measure, denoted as $\mathbf{S}^*$, which is able to capture both the sequential and non sequential nature of text data;

2. the use of the fuzzy version of spectral clustering algorithm when it comes to identify overlapping groups of observations.

The proposed method returns encouraging results. Moreover, it also proves to increase the accuracy of the clustering results.

3.5.1 A novel similarity measure for sequential data: $\mathbf{S}^*$

It is clear that substrings matching through string kernel functions might be not enough in quantifying the level of similarity between documents, since they do not consider the non-sequential parts that may be similar too.
Hence, it is appropriate to consider also other measures which consider the similarity of the whole sequences.
However, the results highlight that the Jaccard index does not capture accurately the similarity between sets containing a different number of items but having the same intersection size.

This has motivated us to introduce a novel similarity measure, $\mathbf{S}^*$.
The proposed similarity measure employs the use of the overlap coefficient (OC) (Vijaymeena & Kavitha, 2016), also known as the Szymkiewicz–Simpson coefficient, which ranges in the interval $[0, 1]$. Given two sets a and b, it is defined as the ratio between the number of common substrings in a and b over the number of substrings in the smallest set:

$$OC(a, b) = \frac{|a \cap b|}{\min(|a|, |b|)}. \tag{3.22}$$

To better understand the differences between the two measures, consider again the following example:

1. $\mathcal{A}$: $\{''cat'','' dog'','' bird'','' mouse''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower''\}$

2. $\mathcal{A}$: $\{''cat'','' dog'','' bird''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower'','' person''\}$

3. $\mathcal{A}$: $\{''cat'','' dog''\}$ vs $\mathcal{B}$: $\{''cat'','' dog'','' tree'','' flower'','' person'','' house''\}$

In this case, the values of the overlap coefficients are equal, respectively, to 0.5, 0.66 and 1. In the last example, the similarity between set $\mathcal{A}$ and set $\mathcal{B}$ is the highest and a score equals to 1 indicates that the set $\mathcal{A}$ is a complete subset of the set $\mathcal{B}$ (indeed, the "content" of the first document is perfectly included in the second one).
It is evident that different sized sets, with the same number of common members, will result in the same Jaccard index.
Moreover, another disadvantage of the Jaccard index is that it is highly influenced by the size of the data. Indeed, large data significantly increase the union whilst keeping the intersection similar.
Clearly, the same reasoning can be applied when the sets represent two generic

text documents and their items are all the possible substrings of generic length l.

The generic expression of the new similarity measure $\mathbf{S}^*$ is formulated as follows:

$$\mathbf{S}^*(a,b) = p \cdot OC(a,b) + (1-p) \cdot SK(a,b). \tag{3.23}$$

It is worth noting that setting $p = 0$ returns the standard string kernel method, while for $1 - p = 0$ we obtain the overlap coefficient.
The new measure $\mathbf{S}^*$ qualifies as a proper similarity measure since it presents the properties of symmetry:

$$\mathbf{S}^*(a,b) = \mathbf{S}^*(b,a). \tag{3.24}$$

The new measure holds also the non-negativity condition, since by definition both the components can be at worst equal to 0:

$$\mathbf{S}^*(a,b) \geq 0. \tag{3.25}$$

In the end, it is normalized in order to range in the interval $[0,1]$, thus:

$$\mathbf{S}^*(a,b) \in [0,1]. \tag{3.26}$$

The next step consists in incorporating the proposed similarity measure $\mathbf{S}^*$ in the novel spectral clustering algorithm. The novel method, which is discussed in the following section, has the advantage to allow for an overlapping between clusters. In this way it is possible to discover relationships between documents that would otherwise be neglected by hard clustering methods.

3.5.2 Fuzzy spectral clustering algorithm with $\mathbf{S}^*$ similarity

The novel algorithm for document clustering can be summarized as follows.

1. Given as input a corpus of N text documents, $\mathcal{C} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$, perform the pre-processing steps for removing the noise in the data by lower casing the terms, by eliminating the punctuation signs and the stopwords and, if necessary, by eliminating the numbers and other meaningless words. Conclude by performing the stemming or the lemmatization.

2. Apply the Spectrum string kernel to the collection of documents and normalize the string kernel values. It is worth remarking that in this step is necessary to identify the optimal value of the parameter l that controls the length of the matching substrings.

The output is an $N \times N$ matrix containing the normalized values of the Spectrum string kernel (according to the parameter l) for each couple of documents in the corpus.

3. Compute the overlap coefficient as a measure of set similarity for all the documents in the collection.
   Again, the output is another $N \times N$ matrix containing the values of the overlap coefficient for each couple of documents.

4. Calculate the similarity matrix $\mathbf{S}^*$ obtained as a weighted combination between the Spectrum string kernel matrix, calculated at Step 2, and the matrix whose entries are the overlap coefficient values calculated at Step 3. In this phase, it is necessary to select the weight p which controls the relative importance attributed to the sequence and set similarities in the construction of $\mathbf{S}^*$.

5. Compute the Laplacian matrix using $\mathbf{S}^*$ as adjacency matrix instead of $\mathbf{\Omega}$:

$$\mathbf{L} = \mathbf{D} - \mathbf{S}^*. \tag{3.27}$$

   Then, calculate the normalized symmetric version of $\mathbf{L}$, $\mathbf{L}_{norm.sym}$.

6. Given the parameter K that controls the number of clusters in the partition, calculate the first K normalized eigenvectors of $\mathbf{L}_{norm.sym}$.

7. Compute the matrix $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{(N \times K)}$ whose columns correspond to the K eigenvectors calculated at the previous point.

8. As the very last step, cluster the rows of the matrix $\boldsymbol{\mathcal{V}}$ (representing the N documents in the collection) in K groups using the fuzzy K-medoids algorithm and choose an appropriate value for the fuzziness parameter m.

Note that Step 1 and Step 2 are executed also for the fuzzy spectral clustering algorithm in combination with both Spectrum string kernel and KS^2M similarity.

To run the novel aforementioned clustering algorithm is required the selection of an adequate value for the number of clusters, K. However, in most clustering applications the optimum K is not known in advance. In this regard, a frequently

used approach consists in running the algorithm using different values for K at every iteration and then applying specific validity measures, such as the fuzzy Silhouette index (Campello & Hruschka, 2006), to identify which K returns to the optimum partition.

3.6 Latent Dirichlet Allocation (LDA)

Before analysing the experimental results of the fuzzy spectral clustering algorithm with $\mathbf{S}^*$ and KS^2M similarities on benchmark and real data sets, the famous and extensively used topic modelling algorithm Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) is discussed.
Indeed, the real data set used in this work has never been applied in literature. In this context, we tried to apply LDA on the identified partitions returned by the proposed fuzzy spectral clustering method in combination with $\mathbf{S}^*$, since it could provide a more complete overview on their interpretation.

LDA is part of the family of probabilistic topic models, which are a category of algorithms used for the analysis of discrete data (such as the management of document archives).
The intuition behind LDA is that the documents in the collection convey several unrevealed "concepts", also known as topics. In other words, documents are characterized by an uncovered (latent) thematic structure. The aim of topic models, including LDA algorithm, consists in discovering the main "concepts" pervading the collection of documents.

The intuition behind the LDA algorithm can be accurately described by its generative process.
Given a set of documents and having identified the number of topics G to discover, the generative process for each document in the collection is represented as:

1. randomly choose a distribution over topics, known as the *per-document distribution over topics*;
2. for each word in the document:
    - choose, in a random way, one topic from the distribution over topics of Step 1;
    - choose, always randomly, a word from the corresponding distribution over the vocabulary.

The intuition behind the underlying statistical model can be summarized as:

- each topic can be present in each document but in different proportions;
- each word is associated to a specific topic chosen from a probability distribution, known as the *per-document distribution over topics* (*topic assignment*);
- each word is selected from the topic distribution over the vocabulary, according to the topic assignment of the previous point.

As introduced, the main objective of topic modelling consists in automatically identifying the topics inherent in a collection of documents. As it is clear, the only observed variables are the words within the documents while the underlying topic structure is unrevealed. Consequently, the objective of LDA consists in uncovering the hidden topic structure using only the observed variables.

Before proceeding with the description of the model, it is necessary to consider the main probabilistic assumption behind LDA.
The method is based on the BOW assumption. Moreover, it also assumes that the specific order of documents in a corpus is irrelevant.
This corresponds to the assumption of exchangeability for the words in a document and for the documents in a corpus. The definition of exchangeability is reported in Preposition 5.

Preposition 5 *A sequence of random variables is exchangeable if for any r-upla* $(X_{n_1}, \ldots, X_{n_r})$ *and any permutation* $(\rho_1, \ldots, \rho_r)$ *it holds:* $(X_{n_1}, \ldots, X_{n_r}) \stackrel{d}{=} (X_{n_{\rho_1}}, \ldots, X_{n_{\rho_r}})$.

Assuming that a sequence of random variables is exchangeable is equivalent to the assume that the random variables are conditionally independent, where the conditioning is with reference to the unknown parameter of the probability distribution.

De Finetti's notion of exchangeability (de Finetti, 1969; De Finetti, 1972) establishes that any collection of random variables that are exchangeable can be represented in terms of a mixture distribution. Consequently, mixture models are used in LDA since they capture the exchangeability of both words and documents.

The LDA model is represented as a probabilistic graphical model and its generative process can be graphically represented, as reported in Figure 3.2. The structure reported in Figure 3.2 is commonly referred to as a hierarchical model.

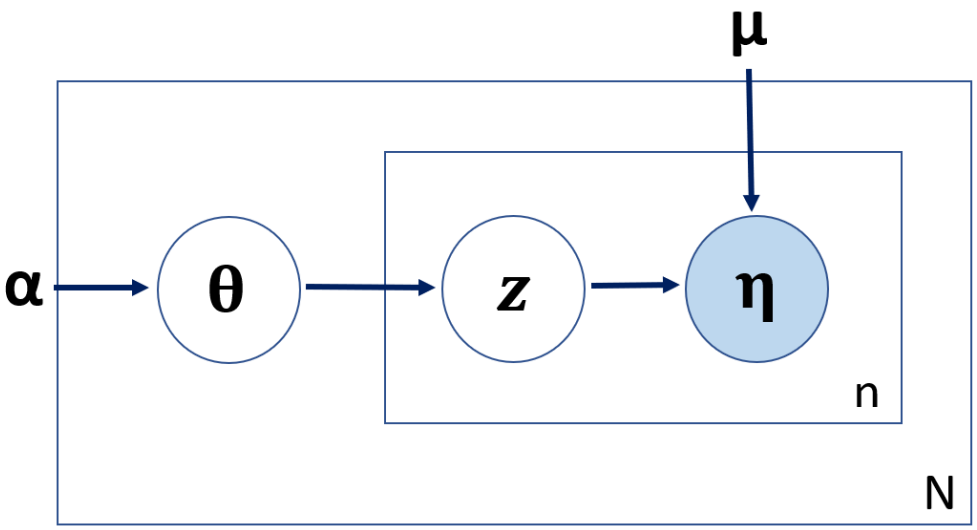


Figure 3.2: Graphical model representation of LDA. The boxes are plates representing replicates. The words, $\boldsymbol{\eta}$, are the only observable variables, while $\boldsymbol{\theta}$ and $\mathbf{z}$ are latent variables. $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are model parameters.

In particular, it is possible to distinguish three different hierarchical levels for the LDA representation.

- Third level: the corpus, where the parameters of relevance are $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$.
- Second level: the document, where the variables $\boldsymbol{\theta}$ are document-level variables and they are sampled once per document.
- First level: the words in the document, where the word-level variables are $\mathbf{z}$ and $\boldsymbol{\eta}$. These last ones are randomly selected once for each word in each document.

The basic idea behind LDA is that each latent topic is identified by a probability distribution over the words characterizing the vocabulary and, at the same time, the documents in the collection are represented in terms of random mixtures over topics.
Given N documents in the corpus $\mathcal{C}$, the complete generative process from which each document arises under the LDA model can be summarized as follows:

1. $n \sim Pois(\epsilon) \longrightarrow n$ is the number of words in each document; it is independent to of all the other variables.

2. $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha}) \longrightarrow \boldsymbol{\theta}$ is a G dimensional vector of probabilities which represents the distribution of topics occurring in each document. It represents the topic proportion and is drawn from a G-dimensional Dirichlet

random variable which can take values in the $(G-1)$ simplex. The vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_G\}$ is the hyper-parameter for the Dirichlet distribution. In the Dirichlet distribution $\boldsymbol{\alpha} \geq 0$ controls the expected value of the distribution which determines the place in the simplex where the distribution is centered. There are some differences in the shape of the distribution if $\boldsymbol{\alpha}$ is bigger or lower than 1.

- $\boldsymbol{\alpha} > 1$ then the Dirichlet is more concentrated in the point corresponding to its expectation and thus all the elements of the vectors have a positive probability. It is characterized by a bump in the middle of the simplex. The peakness of the bump is determined by the specific values assumed by $\boldsymbol{\alpha}$.
- $\boldsymbol{\alpha} < 1$ then the distribution is characterized by sparsity. The Dirichlet distribution will be highly concentrated in a few components and all the rest will have almost no mass.
- $\boldsymbol{\alpha} = 1$ the Dirichlet distribution is equivalent to a uniform distribution over a $G-1$ simplex.

3. $\mathbf{z} \sim Multinomial(\boldsymbol{\theta}) \longrightarrow \mathbf{z}$ is a n dimensional vector of integers between $\{1, \ldots, G\}$ representing the identity of topics for all the words in each document. It is drawn from a multinomial distribution. It is also referred to as the topic assignment.

4. In the end, each word is randomly selected from a the probability distribution of a multinomial, which is conditioned on the topic assignment $\mathbf{z}$. Indeed, $p(\boldsymbol{\eta}|\boldsymbol{z}, \boldsymbol{\mu}) \equiv Multinomial(\boldsymbol{\mu}_{\mathbf{z}})$. In the "classical" approach to LDA, $\boldsymbol{\mu}$ is treated as a fixed quantity that should be estimated and it represents the topics distribution over words.

In this work, we are going to considered the fuller Bayesian approach to LDA, which is characterized by the addition of a Dirichlet prior with parameter $\boldsymbol{\mu}$ on $\boldsymbol{\varphi}$. In the Bayesian approach, the Dirichlet prior represents the per-word topic distribution (Steyvers, Smyth, Rosen-Zvi, & Griffiths, 2004). The main difference is that while in the "classical" model $\boldsymbol{\mu}$ is treated as a fixed quantity that has to be estimated, here both $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are hyper-parameters specifying the nature of the priors on $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. According to Steyvers et al. (2004), the hyper-parameter $\boldsymbol{\mu}$ can be treated as a measure of prior knowledge, before observing the words in the corpus, regarding how many times the terms are sampled from the distributions of the topics. Against this background, the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ should be treated as constant and fixed a priori on values lower than 1 in order to guarantee the concentration of the probability mass on few topics per documents and few

terms per topic (according to the Dirichlet parametrization seen previously).

The new graphical model is represent in Figure 3.3, where $\boldsymbol{\varphi} \sim Dirichlet(\boldsymbol{\mu})$.

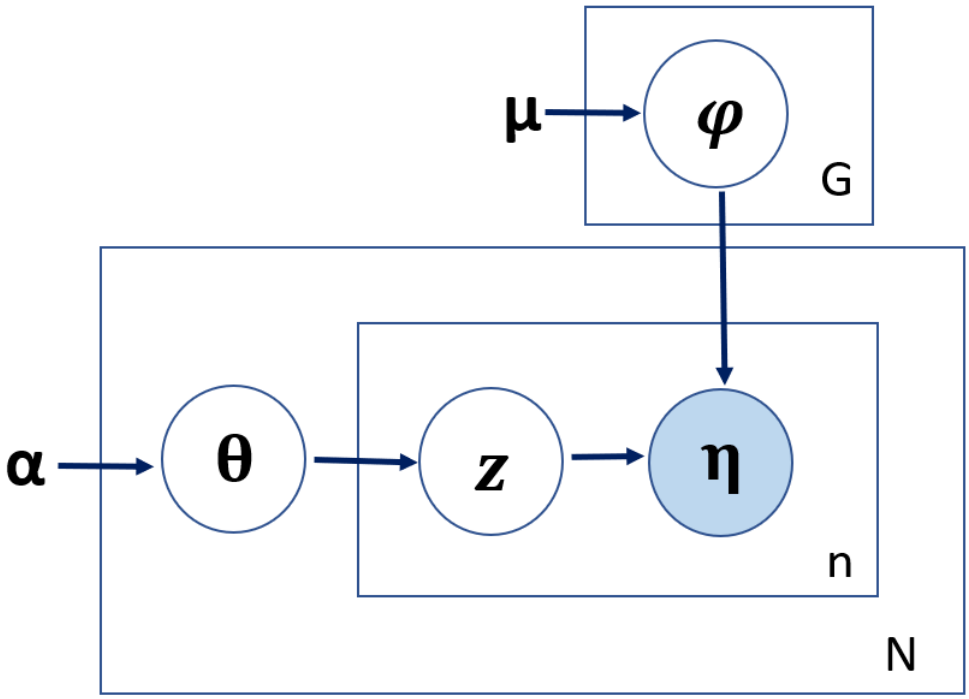


Figure 3.3: Graphical model representation of a fuller Bayesian approach to LDA. The boxes are plates representing replicates.

According to the generative process previously described, the resulting joint distribution of latent ($\boldsymbol{\theta}$, $\mathbf{z}$ and $\boldsymbol{\varphi}$) and observable variables ($\boldsymbol{\eta}$), for the corpus, is given by:

$$p(\boldsymbol{\eta}, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{\alpha}, \boldsymbol{\mu}) = \underbrace{p(\boldsymbol{\varphi}|\boldsymbol{\mu})}_{Dir.} \underbrace{p(\boldsymbol{\theta}|\boldsymbol{\alpha})}_{Dir.} \underbrace{p(\mathbf{z}|\boldsymbol{\theta})}_{Mult.} \underbrace{p(\boldsymbol{\eta}|\boldsymbol{\varphi}, \mathbf{z})}_{Mult.}. \tag{3.28}$$

In particular, we are interested in the topic proportion, $\boldsymbol{\theta}$, and in the topic distribution over words, $\boldsymbol{\varphi}$.

According to the Bayes formula, the conditional distribution of the latent variables, given the observed ones, is proportional to the joint distribution of latent and observed variables divided by the marginal distribution of the documents:

$$p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{z} | \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{z}, \boldsymbol{\eta} | \boldsymbol{\alpha}, \boldsymbol{\mu})}{p(\boldsymbol{\eta} | \boldsymbol{\alpha}, \boldsymbol{\mu})}. \tag{3.29}$$

The drawback is that this quantity can not be computed. The trick used in the fuller Bayesian approach in order to infer the posterior distribution consists

in applying the Gibbs Sampling algorithm. For a more detailed description of Gibbs Sampler refer to Gelfand and Smith (1990); Casella and George (1992). Gibbs Sampling is part of the set of algorithms known as Markov Chain Monte Carlo (MCMC) methods. It does not directly sample from the posterior distribution (which is intractable) but from the full conditional distributions of the variables of the posterior. The MCMC algorithms aim to construct a (ergodic) Markov chain that has the target posterior distribution as its limit distribution. In particular, the limit distribution of the samples generated from the full conditional distributions is the target posterior distribution. For this reason MCMC algorithms require to be run for a big number of iterations, in order to guarantee that the limit distribution of the samples is the target posterior distribution.

Considering LDA, the authors Steyvers et al. (2004) note that instead of directly estimating the posterior distribution (that do not provide a direct estimate of $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$), it is possible to estimate the posterior distribution over $\mathbf{z}$ (the topic assignment), given the words that are observed, $\boldsymbol{\eta}$, and marginalizing with respect to $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$:

$$p(\mathbf{z}|\boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{p(\boldsymbol{\eta}, \mathbf{z})}{\sum_z p(\boldsymbol{\eta}, \mathbf{z})}. \tag{3.30}$$

Then, the conditional distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ can be estimated just using samples from the posterior distribution of the topic assignment. For a complete overview on how this mechanism works, please refer to Blei et al. (2003).

3.7 Empirical analysis

The following section reports the results of the applications of the fuzzy spectral clustering algorithm in combination with $\mathbf{S}^*$ and KS^2M similarities on both benchmark and real data sets.

From a computational perspective and among the different competitors available in literature, the functions of the package `fclust` (Ferraro, Giordani, & Serafini, 2019) are considered when applying the fuzzy spectral clustering algorithm.
Concerning the new proposed similarity measure $\mathbf{S}^*$ and KS^2M, in order to calculate the sequence similarity components, the command `stringdot` of the package `kernlab` (Karatzoglou, Smola, Hornik, & Zeileis, 2004) is used.

In the following experiments the value of the substring length parameter, l, is fixed equal to 5 since, as reported in Lodhi et al. (2002), it is sufficiently high to

represent the average size of a stemmed word in English and at the same time it is short enough to guarantee matches with those terms sharing the same root (stem).

As for the previous introductory experiment, before classifying the data into groups, the documents have been pre-processed.

3.7.1 Benchmark data sets: Reuters-21578 and 20 newsgroups

In the analysis with benchmark data sets, an evaluation of the performance of the proposed methods with $\mathbf{S}^*$ and KS^2M similarities is carried out.
Indeed, the novel fuzzy spectral clustering algorithm is combined with both the similarity measures in order to evaluate their impact on the clustering results.

The experiments are carried out by means of the benchmark data sets Reuters-21578 and 20 newsgroups (Lang, 1995).

For both data sets the true class labels are known a priori, hence the parameter K, representing the number of clusters, assumes a fixed value.
The goodness of the partitions returned by the two methods is evaluated in terms of the fuzzy Silhouette index (Campello & Hruschka, 2006).

As highlighted in the previous sections, the proposed algorithms present a number of free parameters that should be accurately chosen.
In this context, the performance of both methods is assessed for p ranging in $[0, 1]$ by 0.1, in order to evaluate the sensitivity of the results to this parameter. For $p = 1$ the similarity matrix corresponds entirely to the set similarity; on the contrary, for $p = 0$ the string kernel matrix is obtained.

Concerning the fuzzy spectral clustering algorithm, we let the membership degree parameter m vary from 1.1 to 1.5 by 0.1. Moreover, multiple random starts are used in order to avoid finding local optima. In this regard, the inspection of the loss function values shows very similar results, pointing out the robustness of the algorithm.

Reuters-21578 data set

For Reuters-21578, all the 8 categories of R8 data set are used (*acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship*, *trade*), for a total of more than 7000 documents.

By analysing the results of the fuzzy spectral clustering algorithm combined

with $\mathbf{S}^*$ and KS^2M considered separately, what emerges is that the highest value of the fuzzy Silhouette index, 0.76, is returned using $\mathbf{S}^*$ similarity matrix as input of the fuzzy spectral clustering with $m = 1.3$ and $p = 0.4$, corresponding to a weighted combination between set and sequence similarities.
On the other hand, the algorithm with KS^2M similarity returns 0.63 as the highest value of the Silhouette index, obtained for $m = 1.5$ and $p = 0$. In this case KS^2M identifies entirely with the string kernel matrix.
A comparison between the known partitions and the ones returned by the proposed methods with both $\mathbf{S}^*$ and KS^2M is also carried out. In this context the goodness of the clustering algorithms is evaluated in terms of the fuzzy adjusted Rand index.
With $\mathbf{S}^*$ similarity matrix, the comparison between the partition obtained for $m = 1.3$ and $p = 0.4$ and the true-class labels returns a value of the fuzzy adjusted Rand index equals to 0.4. On the other hand, the partition obtained with KS^2M, $m = 1.5$ and $p = 0$ returns a value of the index equals to 0.18.

Some insights into the structure of clusters are provided in Table 3.4 and Table 3.5 which report the main statistics of the membership degrees for both partitions.

In this context, the average values of the membership degrees with KS^2M are lower than the ones obtained with $\mathbf{S}^*$. A similar trend can be identified for the objects with unclear assignment (i.e., those observations whose maximum membership degree is ≤ 0.5): indeed, the percentage of unclear assignments for each cluster is much higher in the partition returned by KS^2M.
Moreover, except for the second cluster, the values of the variation coefficient for $\mathbf{S}^*$ are less scattered than the corresponding ones from KS^2M.

Finally, in order to study the impact of the number of clusters, the parameter K is moved in the interval $[2, 15]$. For both $\mathbf{S}^*$ and KS^2M, the fuzzy Silhouette index returns the highest values for $K = 6$ and $K = 7$, respectively. However, the reduction of the index values when using $K = 8$ are considerably low: -0.058 for $\mathbf{S}^*$ and -0.092 for KS^2M. This last aspect indicates that the value of K chosen based on the Silhouette index reflects well the actual structure of the data.

Table 3.4: Main statistics of the membership degrees for the partition obtained by using $\mathbf{S}^*$ similarity matrix, $m = 1.3$ and $p = 0.4$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	1310	0.19	1	0.92
Cluster 2	1242	0.16	1	0.64
Cluster 3	866	0.37	1	0.97
Cluster 4	1154	0.28	1	0.88
Cluster 5	866	0.16	1	0.66
Cluster 6	556	0.19	1	0.84
Cluster 7	1225	0.16	1	0.86
Cluster 8	455	0.27	1	0.67

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	17.7	4.7%
Cluster 2	52.2	40.4%
Cluster 3	11.3	2.4%
Cluster 4	18.6	4.3%
Cluster 5	35.5	27.8%
Cluster 6	25.6	11.3%
Cluster 7	23.7	2.7%
Cluster 8	27.9	21.1%

Table 3.5: Main statistics of the membership degrees for the partition obtained by using KS^2M similarity matrix, $m = 1.5$ and $p = 0$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	946	0.14	1	0.88
Cluster 2	336	0.17	1	0.35
Cluster 3	644	0.13	1	0.23
Cluster 4	1452	0.14	1	0.60
Cluster 5	1429	0.14	1	0.80
Cluster 6	1309	0.14	1	0.65
Cluster 7	827	0.18	1	0.46
Cluster 8	731	0.17	1	0.46

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	24.2	10.4%
Cluster 2	37.3	87.8%
Cluster 3	43.1	96.9%
Cluster 4	50.7	38.9%
Cluster 5	32.5	17.8%
Cluster 6	41.2	35.1%
Cluster 7	41.3	63.3%
Cluster 8	48.3	62.2%

20 newsgroups data set

The 20 newsgroup data set is a collection, publicly available, containing approximately 20000 newsgroup documents. The documents are partitioned (nearly) evenly across 20 different newsgroups.

For this experiment, following the works of Miao, Duan, Zhang, and Jiao (2009) and L. Shi, Weng, Ma, and Xi (2010), the two categories, *talk* and *science*, characterized by the highest number of news, are used. They have, respectively, 3245 and 3945 documents after pre-processing. Each of the two categories, is characterized by several sub-groups. The sub-groups of *talk* are: *talk.politics.guns*, *talk.politics.mideast*, *talk.politics.misc*, *talk.religion.misc*. The sub-groups of *science* are: *sci.crypt*, *sci.electronics*, *sci.med*, *sci.space*. Clustering is applied separately on the sub-groups of *talk* and *science*.

By investigating the solutions for both the algorithms applied to both categories, we observe better results (with reference to the fuzzy Silhouette index) when using $\mathbf{S}^*$ similarity matrix. In particular, the highest value for *talk*, 0.71, is returned for $p = 0.5$ and $m = 1.5$; while the highest value for *science*, 0.79, is obtained with $p = 0.9$ and $m = 1.1$.
Both categories are characterized by the use of a weighted combination between sequence and set similarities.
Concerning KS^2M, the highest record of the fuzzy Silhouette index for *talk*, 0.64, is returned for $p = 1$ (corresponding entirely to the Jaccard index) and $m = 1.5$; the optimal value for *science*, 0.72, is returned for $p = 0$ (corresponding entirely to the string kernel matrix) and $m = 1.5$.

As before, we inspect the clusters structure of the optimal partitions for both the categories. The main statistics for the category *talk* are reported in Table 3.6 and Table 3.7 considering, respectively, $\mathbf{S}^*$ and KS^2M similarities. From these tables it is possible to notice how $\mathbf{S}^*$ outperforms the competitor: the clusters are characterized by higher average membership degrees, a lower variation coefficient and a lower number of unclear assignments.
In Table 3.8 and Table 3.9 are reported the same statistics for the category *science* where hold similar considerations as before.

In the end, we inspect the solutions for K ranging in the interval $[2, 10]$.
The fuzzy Silhouette index for *talk* returns $K = 3$ with $\mathbf{S}^*$ and $K = 6$ with KS^2M. Also in this experiment, the reduction in the values of the index compared to the partitions with $K = 4$ are not substantial: -0.041 for $\mathbf{S}^*$ and -0.018 for KS^2M.
A different result is found for *science*: with $\mathbf{S}^*$ the optimal value for K coincides

with the true number of classes. On the contrary, KS^2M returns $K = 6$ with a non-substantial increase of 0.028 compared to the value returned by $K = 4$.

Table 3.6: Main statistics of the membership degrees for the partition obtained by using $\mathbf{S}^*$ similarity matrix on *talk* category, $m = 1.5$ and $p = 0.5$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	862	0.50	1	0.95
Cluster 2	863	0.25	1	0.90
Cluster 3	724	0.50	1	0.95
Cluster 4	796	0.36	1	0.90

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	10.28	0%
Cluster 2	16.22	1.04%
Cluster 3	10.99	0.27%
Cluster 4	15.15	0.63%

Table 3.7: Main statistics of the membership degrees for the partition obtained by using KS^2M similarity matrix on *talk* category, $m = 1.5$ and $p = 1$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	730	0.26	1	0.79
Cluster 2	853	0.27	1	0.85
Cluster 3	696	0.27	1	0.76
Cluster 4	966	0.28	1	0.85

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	24.12	9.7%
Cluster 2	22.98	9.4%
Cluster 3	33.95	22.4%
Cluster 4	28.39	6%

Table 3.8: Main statistics of the membership degrees for the partition obtained by using $\mathbf{S}^*$ similarity matrix on *science* category, $m = 1.1$ and $p = 0.9$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	1137	0.38	1	0.98
Cluster 2	861	0.51	1	0.98
Cluster 3	809	0.51	1	0.97
Cluster 4	1138	0.47	1	0.98

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	7.61	0.09%
Cluster 2	6.42	0%
Cluster 3	8.26	0%
Cluster 4	8.26	0.2%

Table 3.9: Main statistics of the membership degrees for the partition obtained by using KS^2M similarity matrix on *science* category, $m = 1.5$ and $p = 0$.

Cluster	N.observations	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	681	0.34	1	0.89
Cluster 2	776	0.35	1	0.91
Cluster 3	977	0.36	1	0.87
Cluster 4	1511	0.34	1	0.90

Cluster	Var.coeff. $\times$ 100	%Unclear.assign.
Cluster 1	17.87	4.4%
Cluster 2	16.83	3.4%
Cluster 3	18.91	4.7%
Cluster 4	15.75	2.5%

3.7.2 The novel fuzzy spectral clustering algorithm in combination with $\mathbf{S}^*$ on real data: a corpus of abstracts from statistical articles collected from ArXiv database

In this section a new data set composed by abstracts of articles collected from ArXiv database is presented. ArXiv is an open-access archive containing more than 1868555 articles distributed in eight different macro-areas: statistics, economics, quantitative finance, quantitative biology, electrical engineering and systems science, mathematics, physics and computer science.

We restrict our interest to the category of statistics considering those articles published from January 1991 (year of foundation of ArXiv) to January 2021, for a total of more than 23000 documents. The data are extracted by using the technique of Python web scraping. Our goal is to evaluate the goodness of the

partition, returned by the fuzzy spectral clustering algorithm in combination with $\mathbf{S}^*$, applied to real data.

Given the initial extracted documents, the experiments are conducted on different years evenly spaced in time: 2010, 2015 and 2020. For each year, two subsets of, respectively, 500 and 1000 randomly selected abstracts are considered. It worth noticing that before 2007 the number of published articles in the category of statistics available in ArXiv database is lower than 1000, leading us to consider only the most recent years.
In this case, nor the class labels nor the number of clusters are known in advance.

In this experimental setup the performance of the algorithm for each different sample size in each year is analysed. In particular, we let p and m vary, respectively, in the intervals $[0, 1]$ and $[1.1, 1.5]$ with step 0.1.
Moreover, since the optimal value of K is not known, for each combination of the two parameters we let K assume values in the set $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20\}$. For a better selection of the optimal combination of parameters (K, p and m) is used the average fuzzy Silhouette width, as a cluster validation factor, together with the Xie and Beni index (Xie & Beni, 1991). This last one is a popular fuzzy cluster validity measure defined as the ratio between the total within-cluster distance, that is the compactness of the fuzzy partition, and a measure of separation between one cluster and another cluster:

$$XB(K) = \frac{\sum_{i=1}^{N} \sum_{g=1}^{K} u_{ig}^2 d^2(\mathbf{x}_i, \mathbf{h}_g)}{N \cdot \min_{g,g'} d^2(\mathbf{h}_g, \mathbf{h}_{g'})}. \tag{3.31}$$

The parameters which return the maximum average fuzzy Silhouette index and the lowest value for the Xie and Beni index are selected. In case there is not a perfect correspondence between the value of K returning the highest average fuzzy Silhouette index and the value of K returning the lowest Xie and Beni index, the candidate partitions are manually inspected.

In Table 3.10 and Table 3.11 are reported the combinations of parameters which return the optimal values of the average fuzzy Silhouette index and the Xie and Beni index for each sample size of each year.

Table 3.10: Optimal combination of parameters for each year (2010, 2015 and 2020) considering the sample size of 1000 randomly selected documents.

Parameter	2010	2015	2020
m	1.5	1.5	1.5
K	2	2	2
p	0.6	0.8	0.9
Average Fuzzy Silhouette	0.95	0.84	0.87
Xie & Beni	0.02	0.07	0.05

Table 3.11: Optimal combination of parameters for each year (2010, 2015 and 2020) considering the sample size of 500 randomly selected documents.

Parameter	2010	2015	2020
m	1.4	1.5	1.4
K	2	2	2
p	0.4	0.7	0.8
Average Fuzzy Silhouette	0.93	0.84	0.86
Xie & Beni	0.03	0.07	0.06

As clearly visible, the number of clusters remain unchanged for both the sample sizes. Also the value of the fuzzifier parameter remains approximately constant around 1.4 and 1.5 for all the years and for each sample size. The similar values taken by p highlight the use of a weighted combination of both the string kernel and the overlap coefficient.
The corresponding values of the average fuzzy Silhouette index and the Xie & Beni index establish that the articles are well associated to the corresponding clusters.

By inspecting the underlying clusters structures it appears clear that the averages membership degrees for all the partitions (for each year and for each sample size) are higher than 0.9.
Nonetheless, as we have introduced a fuzzy clustering algorithm, the clusters are characterized by an overlapping structure: indeed each single abstract can be associated to both clusters simultaneously with membership values in $[0, 1]$.

In Table 3.12, Table 3.13 and Table 3.14 are reported the main statistics on the distribution of the membership degrees for the considered years.

Table 3.12: Membership degrees statistics for the year 2010 on both sample sizes.

Sample size: 1000			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.55	1	0.99
Cluster 2	0.63	1	0.94
Sample size: 500			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.54	1	0.92
Cluster 2	0.62	1	0.99

Table 3.13: Membership degrees statistics for the year 2015 on both sample sizes.

Sample size: 1000			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.55	1	0.97
Cluster 2	0.51	1	0.96
Sample size: 500			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.5	1	0.95
Cluster 2	0.5	1	0.96

Table 3.14: Membership degrees statistics for the year 2020 on both sample sizes.

Sample size: 1000			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.52	1	0.97
Cluster 2	0.5	1	0.96
Sample size: 500			
Cluster	Min.memb.deg.	Max.memb.deg	Av.memb.deg
Cluster 1	0.51	1	0.97
Cluster 2	0.52	1	0.98

Moreover, for each sample size and for each year, it is possible to identify some documents which are fuzzier than the others, i.e. they are assigned to their corresponding clusters with low membership degrees.
Table 3.15 reports, for each sample size, the number of abstracts characterized by maximum membership degrees lower than 0.6.

Table 3.15: Number of abstracts for each sample size whose membership degrees are lower than 0.6.

2010		2015		2020	
500	1000	500	1000	500	1000
7	14	9	16	5	17

It is also provided an overview on the interpretation of the partitions characterized by 1000 documents, by inspecting the topics in each cluster and for each year. In this regard, the LDA algorithm, which has been carefully explained in Section 3.6, is applied in an attempt to provide a more accurate analysis on the "concepts" conveyed by the identified partitions.

The fuller Bayesian approach to LDA is applied to the results of the clustering process with the aim to identify the topics in each group.
However, there is an important parameter that has to be specified before everything else: G, the number of topics for the documents of each cluster. Additionally, to perform the estimation through Gibbs Sampling, is necessary to specify the values of the parameters for the prior distributions, that is $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$. Following the work of Blei et al. (2003), both parameters are set equal to 0.1, in order to select few topics per document and few terms per topic.
Since often the information about the "true" number of topics is not available, the model is run considering different values of the parameter G in order to determine which is the best one. As reported in Blei et al. (2003), a possible way to evaluate the models is to compute the perplexity index.

Perplexity index is a measurement of how well a probability model (in this case LDA) predicts a new set of data. A good text model is the one which assigns a higher probability to the word that actually occurs. Minimizing perplexity is the same as maximizing this probability: a low perplexity indicates that the new set of data is accurately predicted by the probability distribution (the lower, the better).

For each year, the documents assigned to each one of the two clusters are divided, in turn, into training and testing and is applied a 5-fold cross validation to identify the most suitable choice for G. In other words, for each possible candidate of the number of topics, G, the following steps are executed.

1. 4 subsets out of 5 get four turns as part of the training set, which is the sample of data used to fit the LDA model.

2. The remaining subset gets one turn as part of the test set, which is the

sample of data (totally unused for training set) used to provide an evaluation of the model fitted on the training set; the test set is used for the selection of the model by calculating the perplexity index on it.

From an implementation perspective, we have let vary the parameter G in the interval $[2, 10]$ with step equal to 1. Moreover, since LDA is quite computing-intensive, the code is parallelized.

Table 3.16, Table 3.17 and Table 3.18 report the topic distribution over words for the topics characterizing the documents in each cluster. For each topic, the most ten important words according to their probabilities, are listed.
In particular, for 2010 and 2015, the optimal number of topics, G, according to perplexity index, is equal to 3 for both the clusters. For the abstracts of 2020, the optimal value of G is 2.

Table 3.16: Topic distribution over words for the 2010-clusters. The optimal number of topics, G, is equal to 3 for both the clusters.

Cluster 1			Cluster 2		
Topic 1	Topic 2	Topic 3	Topic 1	Topic 2	Topic 3
Estim	Measure	Model	Covari	Discuss	Multivari
Distribut	Data	Problem	Distance	Regress	Model
Function	Statist	Propos	Estim	Quantile	Statist
Result	Study	Algorithm	Cluster	Brownian	Correl
Process	Use	Variabl	Random	Motion	Process
General	Test	Base	Sampl	Distribut	Analysi
Paramet	Time	Number	Infer	Normal	Consider
Random	Analysi	Comput	Label	Limit	Likelihood
Paper	Differ	Approxim	Distribut	Multipl	Matrix
Condit	Network	Demonstr	Independent	Basi	Approxim

Table 3.17: Topic distribution over words for the 2015-clusters. The optimal number of topics, G, is equal to 3 for both the clusters.

Cluster 1			Cluster 2		
Topic 1	Topic 2	Topic 3	Topic 1	Topic 2	Topic 3
Use	Algorithm	Model	Sample	Problem	Function
Analysi	Learn	Data	Test	Algorithm	Paramet
Studi	Problem	Estim	Data	Optim	Mean
Data	Method	Posterior	Process	Base	Simul
Statist	Function	Approach	Statist	Rate	Probabl
Time	Show	Base	Method	Consid	Comput
Predict	Network	Propos	Point	Matrix	Conside
Effect	Result	Paramet	Asymptot	Perform	Set
Base	Perform	Comput	Size	Deriv	Design
Differ	Structur	Bayesian	Base	Show	Random

Table 3.18: Topic distribution over words for the 2020-clusters. The optimal number of topics, G, is equal to 2 for both the clusters.

Cluster 1		Cluster 2	
Topic 1	Topic 2	Topic 1	Topic 2
Learn	Predict	Model	Propos
Network	Graph	Data	Studi
Train	Machin	Distribut	Result
Neural	Featur	Sample	Problem
Dataset	System	Bound	Paper
Architecture	Infer	Algorithm	Challeng
Classif	Accuraci	Test	Show
Imag	Recent	Statist	Task
Deep	Stochast	Variabl	Effect
Gradient	Converg	Rate	Analysi

From the assessment of the contents conveyed by each topic of the identified partitions, we can deduce as follows:

- 2010-partition: neither of the 3 topics characterizing Cluster 1 convey a clearly identifiable concept. Indeed, the 10 most important words of each topic refer mainly to generic terminology used in the statistical domain. On the other hand, based on the observation of the most important words, the topics of Cluster 2 can be associated, respectively, to: cluster analysis, (stochastic) regression techniques and inference & multivariate analysis.

- 2015-partition: the first cluster of 2015-partition is characterized by 3 topics that can be associated, somehow, to econometric problems (treatment and controlling groups), data analysis and Bayesian modelling. The second cluster is characterized by the presence of more inference-related terms such as "sample", "test", "size", "optim", "base", "matrix", "mean", "simul",

"design", "random". Consequently, as seen for the first cluster of 2010-partition, it can be associated to more generic and standard statistical concepts.

- 2020-partition: the two topics characterizing Cluster 1 can be both associated to machine learning techniques. In particular, the first one refers specifically to neural networks. It worth noticing that this specific topic appears only in 2020-partitions, denoting an increasing interest in this thematic in the last years. To conclude, also for the abstracts of 2020, the second cluster is characterized by two topics conveying more general statistical concepts.

What emerges from the content assessment of the analysed partitions, is that there is always one cluster conveying the main research theme of the reference year. On the other hand, the other cluster of the partition can be considered as a sort of "repository" of standard statistical terms/concepts.

3.8 Concluding remarks

To conclude, in this chapter a novel version of fuzzy spectral clustering algorithm to use in combination with string kernel functions is introduced. Moreover, KS^2M similarity is used together with the proposed fuzzy algorithm, identifying a new fuzzy clustering method for text data. However, given the drawbacks of the aforementioned method, a novel similarity measure, $\mathbf{S}^*$, is presented. The new metric is applied together with the novel fuzzy spectral clustering algorithm, producing better results in terms of clustering accuracy.

The empirical results show that the proposed fuzzy spectral clustering algorithm combined with $\mathbf{S}^*$ similarity matrix is able to achieve good performance in both benchmark and real data sets.
In this regard, it is worth noticing that the behavior of graph-based algorithms strongly depend on the structure of the data passed as input; in the case of spectral clustering the performance can be improved by developing more appropriate similarity metrics.
Against this background, the results coming from the introduced experiments on benchmark data sets show that $\mathbf{S}^*$ is able to identify better clusters than KS^2M as evaluated by external and internal validity measures.

Chapter 4

Fuzzy spectral bi-clustering

In line with the spectral approach to clustering, the bi-clustering problem can be seen as a multi-partitioning problem of the original graph that can be solved through the application of spectral methods.
The work illustrated in this chapter presents two novel fuzzy spectral bi-clustering algorithms, allowing both documents and words to belong to all clusters with different degrees of membership. Both algorithms use a graph representation of the data.
The first algorithm operates following a simultaneous approach for the identification of the document and the word partitions. On the other hand, the second method follows a sequential approach.
Applications on both benchmark and real data sets demonstrate that both algorithms are able to identify overlapping clusters and therefore they can be of support for different tasks with text data. Moreover, the novel fuzzy spectral bi-clustering algorithms prove to improve the accuracy of the clustering results.

4.1 Introduction

Bi-clustering, also known as co-clustering or two mode clustering, is a data mining technique used to simultaneously partition the rows and the columns of a data matrix. Bi-clustering methods have been extensively used in different domains: for instance, in the field of genetic and biology, bi-clustering algorithms are employed to simultaneously analyse the different patterns of genes and conditions in microarray data sets (Kluger, Basri, Chang, & Gerstein, 2003). Text mining represents another well known field of application for bi-clustering techniques given the (almost) necessary construction of the VSM in the pre-processing phase. Indeed, in text mining tasks, bi-clustering algorithms allow for a more comprehensive analysis of the corpus of documents, exploiting also the information coming from the feature vectors.
In this context, differently from one-mode clustering, the objective of bi-clustering

consists in identifying clusters of data points that are homogeneous on subsets of features and, on the other hand, identifying clusters of features which are homogeneous on subsets of data points.

Before analysing the two most commonly used spectral bi-clustering algorithms, it worth focusing the attention on the bi-clustering method known as double K-means, introduced by Vichi (2001). The method is an extension to the bi-clustering setting of the K-means algorithm (MacQueen, 1967).

Given a data matrix $\mathbf{X}$ containing N observations and T features, the double K-means problem consists in solving the following constrained minimization problem:

$$\begin{aligned}
\min_{\mathbf{U},\,\mathbf{O},\,\mathbf{H}} \quad & J = \sum_{i=1}^{N}\sum_{j=1}^{T}\sum_{g=1}^{K}\sum_{c=1}^{C}(x_{ij} - h_{gc})^2 u_{ig} o_{jc}, \\
\text{s.t.} \quad & u_{ig} \in \{0,1\} \quad \forall i = 1,\ldots,N, \quad \forall g = 1,\ldots,K, \\
& o_{jc} \in \{0,1\} \quad \forall j = 1,\ldots,T, \quad \forall c = 1,\ldots,C, \\
& \sum_{g=1}^{K} u_{ig} = 1 \quad i = 1,\ldots,N, \\
& \sum_{c=1}^{C} o_{jc} = 1 \quad j = 1,\ldots,T,
\end{aligned} \tag{4.1}$$

where $\mathbf{U}$ is the membership degree matrix for the data points of dimension $N \times T$; $\mathbf{O}$ is the $T \times C$ membership degree matrix for the feature vectors and $\mathbf{H}$ is the $K \times C$ matrix containing the prototypes.

Ferraro, Giordani, and Vichi (2021) have also proposed a fuzzy extension of the previously introduced method in which both the membership degree matrices, $\mathbf{U}$ and $\mathbf{O}$, assume values in the interval $[0, 1]$. The constrained minimization problem becomes:

$$
\begin{aligned}
\min_{\mathbf{U},\,\mathbf{O},\,\mathbf{H}} \quad & J_{fuzzy} = \sum_{i=1}^{N}\sum_{j=1}^{T}\sum_{g=1}^{K}\sum_{c=1}^{C}(x_{ij} - h_{gc})^2 u_{ig}^{m} o_{jc}^{m'}, \\
\text{s.t.} \quad & u_{ig} \in [0,1] \quad \forall i = 1,\dots,N, \quad \forall g = 1,\dots,K, \\
& o_{jc} \in [0,1] \quad \forall j = 1,\dots,T, \quad \forall c = 1,\dots,C, \\
& \sum_{g=1}^{K} u_{ig} = 1 \quad i = 1,\dots,N, \\
& \sum_{c=1}^{C} o_{jc} = 1 \quad j = 1,\dots,T,
\end{aligned}
\tag{4.2}
$$

where m and m' are the fuzziness parameters and their objective consists in controlling the fuzziness of the two partitions. They should be larger than 1. A common choice is to set $m = m' = 2$.

4.2 Spectral bi-clustering

The majority of the applications on spectral bi-clustering methods available in literature rely on partitioning a bipartite graph.
For instance, in Wieling and Nerbonne (2009) a bipartite spectral graph partitioning method is employed in the field of language detection. Indeed, the authors' objective consists in simultaneously cluster the citizens' geographical area with reference to their sound pronunciation of Dutch dialect.
In Guan, Qiu, and Xue (2005) a novel bi-clustering algorithm, employing the spectral decomposition of a bipartite graph, is presented. It is used to simultaneously cluster images and feature vectors.
Xu, Zong, Dolog, and Zhang (2010) propose a bipartite spectral clustering method to bi-cluster web users and web pages in order to identify eventual relationship between the users navigation pages and their preferences.
Finally, in Green, Rege, Liu, and Bailey (2011), a novel spectral bi-clustering algorithm applied to data evolving over time by including historical clustering results, is proposed.

The following section analyses the construction of bipartite graphs, often used as input of spectral bi-clustering algorithms.

4.2.1 Bipartite graph

As the name suggests, the bipartite graph model is based on a double partition of the nodes in the graph. This approach differs from the former one-mode clus-

tering set-up, where the graphs are characterized by a single partition.

The graph bipartitioning model was initially introduced by Zha, He, Ding, Simon, and Gu (2001).
With specific reference to text mining applications, the goal consists in building a bipartite graph in order to model the mutual relationship between words and documents.

The graph $G = (V, E)$, with vertex set $V = \{v_1, ..., v_N\}$, is said to be bipartite if there exist two subsets of V, denoted as V_1 and V_2, with $V_1 \cap V_2 = \emptyset$, such that each edge in E has an extremity in V_1 and the other one in V_2. In document clustering applications, V_1 and V_2 represent, respectively, the set of documents and the set of words. If V_1 and V_2 have equal sizes, G is called *balanced* bipartite graph.

Against this background, it is possible to define a weighted bipartite graph as $G = (V_1, V_2, \mathbf{\Omega})$, where each entry of $\mathbf{\Omega}$ quantifies the (non-negative) weight of the edge connecting the two corresponding endpoints.
If the weight is equal to zero, then there is no edge connecting the two vertices.

In text mining application, the construction of the weighted adjacency matrix $\mathbf{\Omega}$ for bipartite graphs is based on the **DTM**:

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{0} & \mathbf{DTM} \\ \mathbf{DTM}' & \mathbf{0} \end{bmatrix}. \tag{4.3}$$

The two non-diagonal blocks have all the entries equal to zero because there are no edges between documents and words.

4.2.2 Dhillon's spectral bi-clustering algorithm

Concerning text mining applications, one of the most famous bi-clustering algorithm is the one proposed by Dhillon (2001).

The author's solution to the graph clustering problem consists in employing an heuristic spectral graph partitioning method, initially introduced by Pothen, Simon, and Liou (1990), in order to find an approximate solution to minimizing the objective function of the normalized-cut problem. For more details, refer to Dhillon (2001).

As demonstrated in Dhillon's paper, the eigenvalues and the eigenvectors of the Laplacian matrix, $\mathbf{L}$, contain relevant information on how partitioning the graph.

However, Dhillon's method works directly with the original **DTM**, characterized by lower dimensions compared to the Laplacian matrix of the bipartite graph. For instance, if the **DTM** has dimensions $N \times T$, the shape of the Laplacian matrix for the corresponding bipartite graph is $(N + T) \times (N + T)$.

In Dhillon's paper is proved that as a way to find a K-partition of the bipartite graph G, is sufficient to calculate the first $r = \lceil log_2 K \rceil$ left and right singular vectors of the normalized version of the **DTM**, since they contain K-modal information about the data set.
The first $r = \lceil log_2 K \rceil$ left and right singular vectors correspond to the first r largest singular values of the normalized **DTM**.

Moreover, it is also proved that computing the first $r = \lceil log_2 K \rceil$ left and right singular vectors of the normalized **DTM**, is equivalent to calculate the eigenvectors associated to the r (smallest) eigenvalues of the following generalized eigenvalue problem involving the Laplacian matrix:

$$\mathbf{L}\mathbf{\Psi}_{double} = \mathbf{D}\mathbf{\Psi}_{double}\mathbf{\Sigma}, \tag{4.4}$$

where $\mathbf{\Psi}_{double}$ and $\mathbf{\Sigma}$ are, respectively, the matrices of eigenvectors and eigenvalues.

The first $r = \lceil log_2 K \rceil$ eigenvectors of the above generalized eigenvalue problem provide an approximate solution to the issue of finding the minimum normalized cut.
However, it is possible to obtain the same results using the normalized version of the original **DTM**.

More specifically, Dhillon's spectral bi-clustering algorithm is summarized below.

1. Given the $\mathbf{DTM} = [w_{i,j}]_{i=1,...N;,j=1,...,T}$, its normalized version is calculated as:
$$\mathbf{DTM}_{norm} = \mathbf{R}^{-\frac{1}{2}} \cdot \mathbf{DTM} \cdot \mathbf{C}^{-\frac{1}{2}}, \tag{4.5}$$
where $\mathbf{R}$ is the diagonal matrix with entry i equal to $\sum_{j=1}^{T} w_{ij}$ and $\mathbf{C}$ is the diagonal matrix with entry j equal to $\sum_{i=1}^{N} w_{ij}$.

2. Apply the Singular Value Decomposition to $\mathbf{DTM}_{norm}$:
$$\mathbf{DTM}_{norm} = \mathbf{\Gamma}_1 \mathbf{\Sigma} \mathbf{\Gamma}_2^{'}. \tag{4.6}$$

The row partition is obtained from a subset of the left singular vectors while, on the other hand, the column partition is obtained from a subset of the right singular vectors.

3. Compute the $r = \lceil log_2 K \rceil$ singular vectors of $\mathbf{DTM}_{norm}$, which provide the most relevant information for both the documents and the words; then build the $(N + T) \times r$ dimensional matrix $\mathbf{Z}$:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{R}^{-\frac{1}{2}}\mathbf{\Gamma}_1 \\ \mathbf{C}^{-\frac{1}{2}}\mathbf{\Gamma}_2 \end{bmatrix}. \tag{4.7}$$

4. Apply K-means clustering algorithm on the $(N+T)\times r$ dimensional matrix $\mathbf{Z}$. The first N cluster-assignments represent the document partition while the remaining T cluster-assignments provide the term partition.

4.2.3 A novel fuzzy version of spectral bi-clustering based on a simultaneous approach

The papers available in literature on fuzzy spectral bi-clustering algorithms are very few, highlighting how these methods represent a novel and promising field of research for text mining applications.
The most relevant works include the one from N. Liu, Chen, and Lu (2013), employing the use of fuzzy K-harmonic means, and Cano, Adarve, López, and Blanco (2007) that use a modified Improved Possibilistic Clustering algorithm (IPC), by mixing possibilistic and probabilistic approaches (Zhang & Leung, 2004) in order to find bi-clusters under a spectral setting.
Against this background, a further step in the direction of developing a fuzzy spectral bi-clustering method consists in extending Dhillon's spectral bi-clustering algorithm under a fuzzy set-up in order to allow for an overlapping of both document and word clusters.
Indeed, the standard K-means on which relies Dhillon's method, belongs to the category of hard clustering methods. Consequently, it does not allow for an overlapping between clusters.

Against this background, different fuzzy clustering algorithms can be applied to cluster the $(N + T) \times r$ dimensional observations. Among the competitors, fuzzy K-means, fuzzy spherical K-means and fuzzy K-medoids are considered in our proposal.
The proposed fuzzy extension of Dhillon's spectral bi-clustering algorithm can be briefly formulated as follows.

Algorithm 1 Fuzzy extension of Dhillon's spectral bi-clustering algorithm

Input DTM, number of clusters $K_{docs} = K_{words} = K$, fuzziness parameter m.

Output Simultaneous K-partitions for the documents and the words.

1: **procedure** :

2: Given the **DTM**, calculate its normalized version as

$$\mathbf{DTM}_{norm} = \mathbf{R}^{-\frac{1}{2}} \cdot \mathbf{DTM} \cdot \mathbf{C}^{-\frac{1}{2}}.$$

3: Apply the SVD to the normalized version of the **DTM**:

$$\mathbf{DTM}_{norm} = \mathbf{\Gamma}_1 \mathbf{\Sigma} \mathbf{\Gamma}_2$$

and compute the first $r = \lceil log_2 K \rceil$ singular vectors. They are associated to the first r singular values providing the most relevant information for both the documents and the words.

4: Build the matrix $\mathbf{Z}$ as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{R}^{-\frac{1}{2}} \mathbf{\Gamma}_1 \\ \mathbf{C}^{-\frac{1}{2}} \mathbf{\Gamma}_2 \end{bmatrix}.$$

5: Cluster the $N + T$ points, corresponding to the rows of the $(N + T) \times r$ dimensional matrix $\mathbf{Z}$, in K groups with one of the following fuzzy clustering algorithms: fuzzy K-means, fuzzy spherical K-means and fuzzy K-medoids. In this step, it is necessary to choose an appropriate value for the fuzziness parameter m.

6: **end procedure**

Following the procedure described in the above algorithm, the first N rows of the matrix $\mathbf{Z}$ provide the fuzzy partition of the documents (the rows of the **DTM**), while the remaining T data points provide the partition of the words (corresponding to the columns of the **DTM**).

One of the main advantages of this approach, apart from allowing for an overlapping between clusters, is the possibility to obtain simultaneously both the document and the word partitions. Even though it may seem obvious following the standard definition of bi-clustering, it is not when dealing with spectral bi-clustering algorithms. Indeed, spectral bi-clustering methods rely on the spectral decomposition of a transformation of the **DTM** or the Laplacian matrix. In this setting, some methods identify some latent lower dimensional spaces where projecting the rows and columns of the input matrix. Then, the clustering algorithm is applied separately on the two spaces in order to find the desired partitions.

In Section 4.3 experiments are conducted to show the accuracy of the proposed method.

4.2.4 A novel fuzzy version of spectral bi-clustering based on a sequential approach

As analysed above, the method introduced in Section 4.2.3 is characterized by the following advantages:

1. the word partition and the document partition are identified simultaneously;
2. the overlapping between document clusters and word clusters is allowed.

However, this approach suffers from a major drawback that consists in selecting the same number of clusters for the partition of the rows and the partition of the columns (in the algorithm: $K_{docs} = K_{words} = K$).
Indeed, the clustering algorithms are directly applied to the $N + T$ rows of the matrix $\mathbf{Z}$, with the consequence to avoid selecting two partitions characterized by a distinct number of groups.
To overcome the above (restrictive) problem, we focus our attention to the work of Kluger et al. (2003) who introduce a spectral-based approach found on the identification of two separate spaces where projecting the rows and the columns of the data matrix. The authors have applied their algorithm to cluster genes and experimental conditions in order to automatically classify cancer data sets.

However, in the method described in Kluger et al. (2003), the data points and the features are uniquely assigned to the corresponding clusters, determining a classification that can lead to unrealistic results. Our proposed approach, which is carefully explained in the following pages, has the advantage to extend Kluger's method under a fuzzy perspective.

In particular, Kluger's method is based on the assumption that the input data matrix is characterized by an uncovered checkerboard structure which can be reformulated in terms of an eigenvalue problem.
To understand the connection between the identification of the checkerboard structure in the data matrix and the eigenvalue problem, consider the following example.

Suppose that the matrix $\mathbf{B}$ has a perfect checkerboard structure and $\mathbf{x}$ and $\mathbf{y}$ represent two piecewise constant vectors identifying, respectively, the row and the column partitions.

The application of the two classification vectors to the input matrix **B** returns, as output, two new classification vectors: one for the rows, **x'**, and the other one for the columns, **y'**:

$$\mathbf{By} = \mathbf{x'}, \tag{4.8}$$

$$\mathbf{B}^T\mathbf{x} = \mathbf{y'}. \tag{4.9}$$

It worth noticing that **x'** and **y'** maintain the same partitioning structure of, respectively, **x** and **y**.
A graphical representation of this scheme is reported in Figure 4.1.

From the previous equations is obtained:

$$\mathbf{B}^T\mathbf{Bx} = \mathbf{x'}, \tag{4.10}$$

$$\mathbf{BB}^T\mathbf{y} = \mathbf{y}. \tag{4.11}$$

This result evidences that if **B** is characterized by the presence of a checkerboard structure, it can be revealed through solving an eigenvalue problem:

$$\mathbf{B}^T\mathbf{Bu*} = \sigma^2\mathbf{u*}, \tag{4.12}$$

$$\mathbf{BB}^T\mathbf{v*} = \sigma^2\mathbf{v*}, \tag{4.13}$$

where **u*** and **v*** are the eigenvectors and σ^2 is the common eigenvalue.

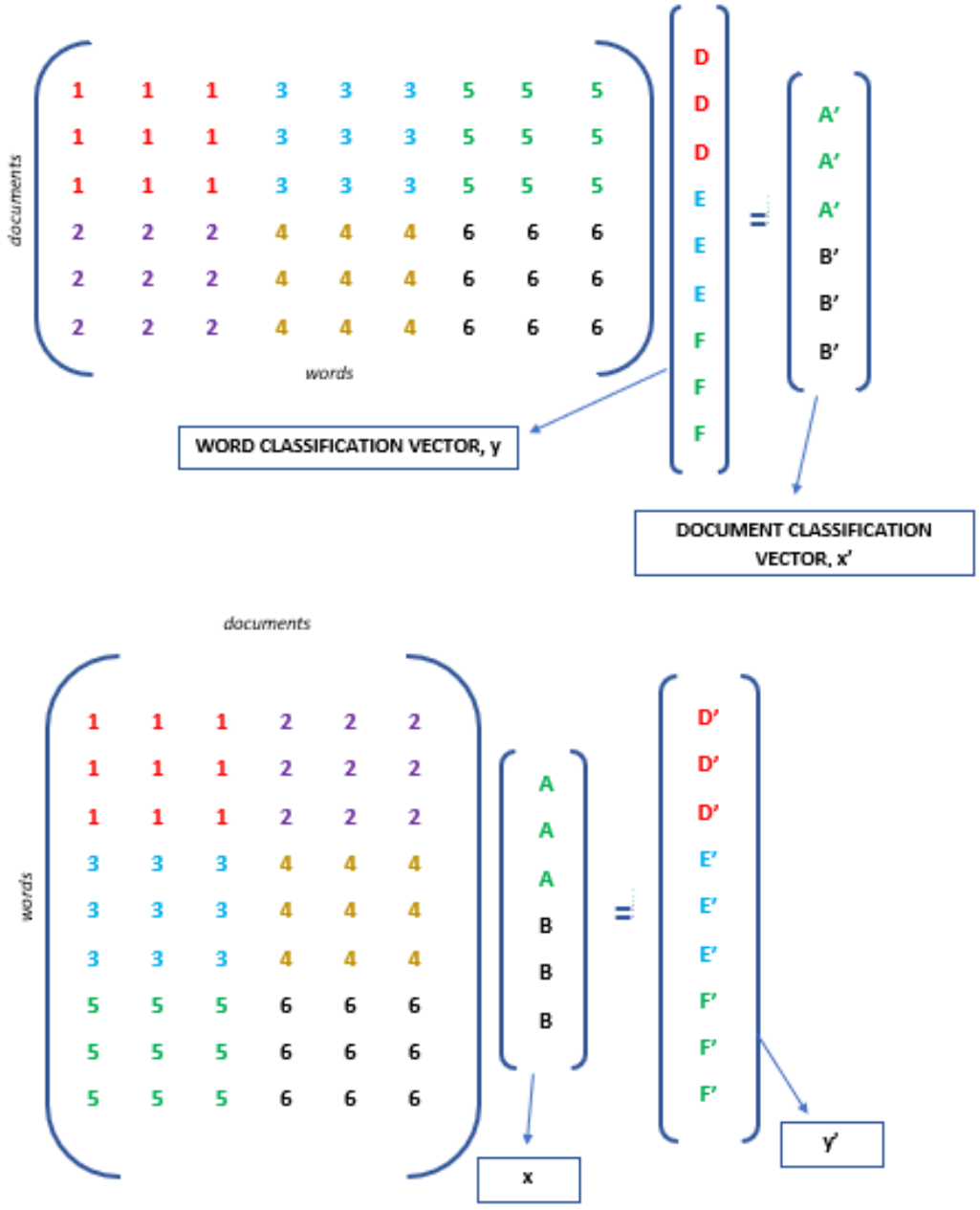


Figure 4.1: Graphical representation of Kluger bi-clustering method.

Against this background, to identify weather the input data matrix has a (hidden) checkerboard arrangement, the documents and words eigenvectors (in the example, **u*** and **v***) are investigated in order to verify if a pair of them has an approximate piecewise constant structure.
In other words, to check if the data matrix $\mathbf{B}$ has a checkerboard arrangement, is required to solve an eigenvalue problem involving $\mathbf{B}^T\mathbf{B}$ and $\mathbf{B}\mathbf{B}^T$, which is equivalent to find the SVD of $\mathbf{B}$.
Consequently, identify the existence of a pair of piecewise constant eigenvectors is of relevance to establish the presence of a checkerboard pattern in the data matrix.

Against this background, our second proposed fuzzy spectral bi-clustering algorithm is described below.

1. Given a corpus of N documents with T unique terms, the first step consists in the normalization of the input **DTM**, through the log-normalization technique (Kluger et al., 2003). This step has the effect to eliminate eventual noise in the data.
   It is initially computed the logarithmic version of the **DTM**, $\mathbf{DTM}_{log}$:

$$\mathbf{DTM}_{log} = Log(\mathbf{DTM}). \tag{4.14}$$

   Then, a rescaling and a normalization of both the dimensions (rows and columns) is carried out. In particular, each entry of the final normalized matrix, $\mathbf{A}$, is computed according to the following formula:

$$\mathbf{A}_{ij} = \mathbf{DTM}_{log;ij} - \mathbf{DTM}_{log;i.} - \mathbf{DTM}_{log;.j} + \mathbf{DTM}_{log;..}, \tag{4.15}$$

   where $\mathbf{DTM}_{log;i.}$ is the column mean for the i-th row, $\mathbf{DTM}_{log;.j}$ is the row mean for the j-th column and $\mathbf{DTM}_{log;..}$ is the overall mean.
   This transformation removes the systematic variability among rows and columns. The remaining values A_{ij} capture the interaction between the i-th row and the j-th column that cannot be explained by systematic variability among rows, among columns, or within the entire matrix.

2. Similarly to Dhillon's method, the SVD is applied to the transformed matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{\Gamma}_{1,seq}\mathbf{\Sigma}_{seq}\mathbf{\Gamma}_{2,seq}. \tag{4.16}$$

   Then, the three "best" eigenvectors are selected. The decision to select three eigenvectors derives directly from the work of Kluger et al. (2003)

where the authors, for their experiments, apply the clustering step on the data projected to the "best" three eigenvectors.
It worth noticing that the term "best" refers to the eigenvectors containing the optimal partitioning information. In this regard, the best eigenvectors are usually considered as the ones associated to the largest eigenvalues. However, Kluger et al. (2003) show that in some rare cases an eigenvector corresponding to a small eigenvalue can also contain relevant partitioning information.
In order to identify the three "best" eigenvectors, all the candidate vectors of the eigensystem are examined by fitting them to piecewise constant vectors. To carry out this step, the entries of each examined eigenvector are initially monotonically sorted. Then, all the possible thresholds are analysed in order to find a possible piecewise constant vector that can approximate the eigenvectors. Finally, the "best" eigenvectors are selected. From a practical perspective, this procedure is equivalent to apply the K-means clustering algorithm to each one-dimensional eigenvector. The best eigenvectors are identified by means of the minimum Euclidean distance between each vector and its piecewise constant approximation.

3. Let $\mathbf{\Gamma}^{best}_{1,seq}$ be the matrix whose columns correspond to the three best left singular vectors and $\mathbf{\Gamma}^{best}_{2,seq}$ be the matrix whose columns are the three best right singular vectors. In order to find a partition of the documents, the rows of $\mathbf{A}$ are projected to the three-dimensional space $\mathbf{A\Gamma}^{best}_{2,seq}$. On the other hand, to partition the words, the columns of $\mathbf{A}$ are projected to $\mathbf{A}^T\mathbf{\Gamma}^{best}_{1,seq}$.

4. The conclusive step is the application of a fuzzy clustering algorithm (fuzzy K-means, fuzzy spherical K-means and fuzzy K-medoids) to $\mathbf{A\Gamma}^{best}_{2,seq}$ and $\mathbf{A}^T\mathbf{\Gamma}^{best}_{1,seq}$.
   In this way, the partition of the documents and the partition of the words are obtained.
   Notice that, differently from Dhillon's method, in this setting is possible to specify two distinct numbers of clusters for both the documents and the words.

Contrary to what is claimed in the fuzzy extension of Dhillon's bi-clustering algorithm, the method just described follows a sequential approach. Indeed, the two new projecting spaces $\mathbf{A\Gamma}^{best}_{2,seq}$ and $\mathbf{A}^T\mathbf{\Gamma}^{best}_{1,seq}$, for the documents and the words, are clustered independently from each other.

4.3 Applications

In this section, a comparison of the two proposed fuzzy spectral bi-clustering algorithms on both benchmark and real data sets is carried out.
We will refer to the first algorithm as *Joint Fuzzy Spectral Bi-clustering* algorithm (JFSB) and to the second one as *Sequential Fuzzy Spectral Bi-clustering* algorithm (SFSB).

4.3.1 Benchmark data set: WebKB

The publicy available benchmark data set used for this experiment is WebKB. Each item of the data set is a document representing a web page collected by the World Wide Knowledge Base project of the CMU text learning group. The web pages are collected from the departments of the computer science faculty of various universities in 1997 (Cornell University, The University of Texas, The University of Washington and The University of Wisconsin), manually classified into seven different classes: *student*, *faculty*, *staff*, *department*, *course*, *project*, and *other*. In particular, the classes *staff* and *department* are discarded since there are only a few documents for both the categories. The class *other* has not been considered since it is a miscellaneous of the other classes.
From the remaining 4199 documents and following the work of Joachims, Cristianini, and Shawe-Taylor (2001), the analysis is carried out on a random sample of 800 web pages.
Before applying the two fuzzy spectral bi-clustering algorithms, the documents have been pre-processed by removing the punctuation signs and the stopwords, by lower casing the remaining text and by applying the Porter Stemmer algorithm. Moreover, four empty documents have been removed from the collection, for a total of 796 documents.

In this experiment, the true-class labels for the document partitions are known a priori. Consequently, the parameter K_{docs} is fixed equal to 4 for both the algorithms.
The *Joint Fuzzy Spectral Bi-clustering* algorithm requires that the column partition and the row partition are characterized by the same number of clusters. Hence, $K_{docs} = K_{words} = 4$.
On the contrary, the *Sequential Fuzzy Spectral Bi-clustering* has the advantage to choose a distinct number of groups for the two partitions. In this setting, the optimal result for K_{words} has been selected within a certain range of values.

To inspect the goodness of the document partitions the Purity index is used (Schütze, Manning, & Raghavan, 2008).

In the context of cluster analysis, Purity is an external validation index and is defined as the proportion of the correctly classified data points over the total number of observations. It ranges in the interval $[0, 1]$.
With reference to the partition of the documents, it is mathematically expressed by the following equation:

$$Purity(K) = \frac{1}{N} \sum_{g=1}^{K_{docs}} max_g |C_g \cap TC_{g*}|, \tag{4.17}$$

Where C_g is the g-th cluster of the partition and TC_{g*} is the cluster $g*$ of the true classification having the maximum count for cluster C_g. A value of the Purity index equals to 1 indicates a perfect clustering, with a perfect matching between the identified clusters and the true document classification.
It worth emphasizing that the Purity index is used only to evaluate the document partitions. In this context, the fuzzy document partitions returned by the two methods are converted into crisp partitions by assigning each object to the cluster characterized by the highest membership degree. Indeed, this external validity measure is used to compare the corresponding crisp document partitions returned by the two fuzzy bi-clustering algorithms with the known partition of the documents (which is the only external information available), that acts like a sort of benchmark.
Since we do not have any information about the original classification of the words, the possible comparison between the true word classification and the word partitions could not have been taken into consideration for this purpose (indeed, the word partitions are evaluated only in terms of internal validity measure which considers also the membership degrees of the words).
However, since the Purity index does not take into account any information about the membership degrees characterizing the fuzzy partitions, the fuzzy Silhouette index has also been considered. In particular, this last one is used to evaluate both the document and the word partitions.
With reference to the *Sequential Fuzzy Spectral Bi-clustering* algorithm, the fuzzy Silhouette index is used to identify the optimal number of clusters for the word partition.
All the cluster validity indexes used (both internal and external) are calculated for the parameter m varying in the interval $[1.1, 2]$ with step 0.1.
The criteria for evaluating the two fuzzy spectral clustering algorithms are summarized as follows.

1. *Joint Fuzzy Spectral Bi-clustering*: given $K_{docs} = K_{words} = 4$ for each value of the parameter m in $[1.1, 2]$, the Purity index for the document partitions is calculated; furthermore, the fuzzy Silhouette index is calculated for both the document and the word partitions.

When applying the K-means algorithm the Silhouette index is used.

2. *Sequential Fuzzy Spectral Bi-clustering*: for each value of m in the interval $[1.1, 2]$, K_{docs} is fixed equal to 4 while the parameter K_{words} assumes values from 2 to 10. As before, the document partitions are evaluated by means of the Purity index and the fuzzy Silhouette index, which is also used in the evaluation of the word partitions.
When applying the K-means algorithm the Silhouette index is used.

For both algorithms, the optimal combination of parameters, (K_{docs}, K_{words}, m), is selected as the one maximizing the following cluster validity measures: Purity index for the document partitions, fuzzy Silhouette index for the document partitions and fuzzy Silhouette index for the word partitions. However, in case the cluster validity indexes assume not clearly distinguishable values, all the partitions candidate to be the optimal ones are manually inspected.

An analysis of the structure of the membership degrees is also carried out. Moreover, in order to investigate the nature of the word partitions, the most frequent terms, for both spectral bi-clustering algorithms, are examined.

A comparison between the results of the standard Dhillon's and Kluger's spectral bi-clustering algorithms and the results of our proposed methods in combination with fuzzy K-means, fuzzy K-medoids and fuzzy spherical K-means follow. Before analysing in details the clustering results, Table 4.1 reports the parameters identifying the optimal partitions for each version of the fuzzy spectral bi-clustering methods.

Table 4.1: Parameters returning the optimal partitions for all the versions of the *Joint Fuzzy Spectral Bi-clustering* algorithm and the *Sequential Fuzzy Spectral Bi-clustering* algorithm. It worth emphasizing that the parameters $K_{docs} = K_{words} = 4$ for the *Joint Fuzzy Spectral Bi-clustering*, as well as the parameter $K_{docs} = 4$ for the *Sequential Fuzzy Spectral Bi-clustering*, are chosen to be fixed for this specific experiment.

Parameters	Fuzzy K-means		Fuzzy Spherical K-means		Fuzzy K-medoids	
	Joint	*Sequential*	*Joint*	*Sequential*	*Joint*	*Sequential*
K_{docs}	4	4	4	4	4	4
K_{words}	4	3	4	2	4	4
m	1.7	1.2	1.9	1.6	2	1.2

As it is possible to notice, the number of clusters for the partition of the words assumes values from 2 to 4. On the other hand, the values assumed by the fuzziness parameter m are characterized by more heterogeneity, ranging from a minimum of 1.2 to a maximum of 2.

For the two standard methods, Dhillon's and Kluger's spectral bi-clustering algorithms, the optimal numbers of clusters for the word partitions are, respectively, 4 and 3.

What emerges from the results of the comparative analyses, reported in Table 4.2, Table 4.3, Table 4.4 and Table 4.5, is that the proposed versions of fuzzy spectral bi-clustering algorithms, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering*, outperform the corresponding crisp competitors (respectively, Dhillon's and Kluger's methods) in terms of Purity index calculated on the document partitions. Indeed, Dhillon's method returns a value of the Purity equals to 0.51, against 0.6, 0.56 and 0.6 of the *Joint Fuzzy Spectral Bi-clustering* combined, respectively, with fuzzy K-means, fuzzy spherical K-means and fuzzy K-medoids. With respect to Kluger's method, the difference is even more pronounced: the crisp approach returns a Purity value equals to 0.49, while the *Sequential Fuzzy Spectral Bi-clustering* algorithm, in combination with fuzzy K-means, fuzzy spherical K-means and fuzzy K-medoids, returns values of the index equal to, respectively, 0.55, 0.63 and 0.54.

Fuzzy K-means

Table 4.2: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy K-means.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.59	0.55
fuzzy Silhouette (docs)	0.64	0.73
fuzzy Silhouette (word)	0.60	0.92

Fuzzy K-medoids

Table 4.3: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy K-medoids.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.59	0.54
fuzzy Silhouette (docs)	0.59	0.54
fuzzy Silhouette (word)	0.58	0.84

Fuzzy spherical K-means

Table 4.4: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy spherical K-means.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.56	0.63
fuzzy Silhouette (docs)	0.19	0.43
fuzzy Silhouette (word)	0.21	0.72

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.5: Comparison, in terms of validity indexes, of Dhillon's and Kluger's spectral bi-clustering algorithms.

indexes	*Dhillon*	*Kluger*
Purity (docs)	0.51	0.49
Silhouette (docs)	0.56	0.39
Silhouette (word)	0.57	0.70

The comparison of the different versions of both the *Joint Fuzzy Spectral Bi-clustering* and the *Sequential Fuzzy Spectral Bi-clustering* algorithms evidences that the latter returns more accurate clustering results for both the document and the word partitions. Indeed, the values of the fuzzy Silhouette indexes for the two partitions are almost always higher than the corresponding ones obtained from the joint approach[1].

[1]Only the fuzzy Silhouette index of the document partition returned by the employment of the fuzzy K-medoids clustering algorithm is higher in correspondence of the *Joint Fuzzy Spectral Bi-clustering* (0.59 vs. 0.54).

Both versions with fuzzy spherical K-means return lower results compared to the competitors: indeed, the *Joint Fuzzy Spectral Bi-clustering* algorithm returns 0.19 and 0.21 as values of the fuzzy Silhouette index for the document and the word partitions, respectively. On the other hand, the values of the fuzzy Silhouette index for the *Sequential Fuzzy Spectral Bi-clustering* are slightly higher and are equal to 0.43 for the document partition and 0.72 for the word partition.
Analysing the word partitions, the *Sequential Fuzzy Spectral Bi-clustering* algorithm assumes 0.92 as the highest value (in combination with fuzzy K-means) and 0.72 as the minimum value (in combination with fuzzy spherical K-means). On the contrary, when employing the *Joint Fuzzy Spectral Bi-clustering* algorithm, the highest value of the fuzzy Silhouette index calculated on the word partition is equal to 0.6 (in combination with fuzzy K-means), while the minimum value is equal to 0.21 (in combination with fuzzy spherical K-means).

Differently from Dhillon's and Kluger's methods, the *Joint Fuzzy Spectral Bi-clustering* and the *Sequential Fuzzy Spectral Bi-clustering* algorithms have the advantage to identify overlapping structures in the document and word partitions. Analysing the membership degrees structures of both the document and the word partitions, for all the different combinations of fuzzy clustering algorithms, what emerges from Table 4.6, Table 4.7, Table 4.8, Table 4.9, Table 4.10 and Table 4.11, is that the two methods produce word and document partitions characterized by average values of the membership degrees almost always higher than 0.8. With specific reference to the *Sequential Fuzzy Spectral Bi-clustering* algorithm the average membership degrees very often far exceed the value of 0.9.

Fuzzy K-means

Table 4.6: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	0.99	0.87
Cluster 2	0.45	0.99	0.88
Cluster 3	0.25	0.99	0.81
Cluster 4	0.48	0.99	0.86
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.53	1	0.92
Cluster 2	0.44	1	0.94
Cluster 3	0.57	1	0.94
Cluster 4	0.51	1	0.97

Table 4.7: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.26	1	0.91
Cluster 2	0.47	0.99	0.87
Cluster 3	0.25	0.99	0.88
Cluster 4	0.47	0.99	0.87
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.47	1	0.90
Cluster 2	0.45	1	0.93
Cluster 3	0.44	1	0.91

Fuzzy K-medoids

Table 4.8: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.41	0.99	0.83
Cluster 2	0.46	0.99	0.81
Cluster 3	0.49	0.99	0.82
Cluster 4	0.25	0.99	0.78
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.40	1	0.92
Cluster 2	0.51	1	0.95
Cluster 3	0.49	1	0.95
Cluster 4	0.45	1	0.93

Table 4.9: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.40	1	0.82
Cluster 2	0.44	1	0.79
Cluster 3	0.25	1	0.85
Cluster 4	0.25	1	0.82
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.36	1	0.99
Cluster 2	0.39	1	0.92
Cluster 3	0.39	1	0.90
Cluster 4	0.39	1	0.89

Fuzzy spherical K-means

Table 4.10: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.35	0.99	0.81
Cluster 2	0.50	1	0.95
Cluster 3	0.49	0.99	0.98
Cluster 4	0.43	0.99	0.84
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.58	0.99	0.93
Cluster 2	0.60	0.99	0.90
Cluster 3	0.71	0.99	0.96
Cluster 4	0.43	0.99	0.81

Table 4.11: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.33	1	0.98
Cluster 2	0.31	1	0.97
Cluster 3	0.36	1	0.81
Cluster 4	0.38	0.99	0.81
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.83
Cluster 2	0.51	1	0.92

In the end, a comparison of the content conveyed by the word partitions is also carried out. In this context, despite the two algorithms are characterized by a different number of clusters for the partition of the words, no significant differences emerge from the two fuzzy spectral bi-clustering methods. The majority of the clusters can be associated to one of the following topics: "courses and classes"; "student commitments" (characterized by words such as: homework, assignments, exams, ...); "research issues" (characterized by words such as: visiting, PhD, research, libraries, ...); "graduation" and"academic/scholastic furniture".

Table 4.12 summarizes the content of each cluster for all the versions of the proposed fuzzy spectral bi-clustering algorithms. Note that not all the clusters are characterized by an easily identifiable content. As it is possible to notice from Table 4.12, the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-medoids is the method returning clusters which are more easily identifiable, according to the first ten most frequent words. The detailed list of the ten most frequent words for each cluster are reported in Table 4.13, Table 4.14, Table 4.15 and Table 4.16.

Table 4.12: Main concepts conveyed by the most ten frequent terms for each cluster in the identified partitions of the fuzzy spectral bi-clustering algorithms.

Fuzzy K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	"research issues"	Not clearly distinguishable
Cluster 2	Not clearly distinguishable	"academic/scholastic furniture"
Cluster 3	"student commitments"	"student commitments"
Cluster 4	"courses and classes"	-
Fuzzy K-medoids		
Cluster	*Joint*	*Sequential*
Cluster 1	"courses and classes"	"research issues"
Cluster 2	Not clearly distinguishable	"student commitments"
Cluster 3	"research issues"	"graduation"
Cluster 4	"student commitments"	"courses and classes"
Fuzzy spherical K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	"research issues"	"courses and classes"
Cluster 2	"courses and classes"	Not clearly distinguishable
Cluster 3	"student commitments"	-
Cluster 4	Not clearly distinguishable	-

Fuzzy K-means

Table 4.13: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with fuzzy K-means algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*		
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3
system	comput	assign	program	comput	phone	page
research	science	homework	class	science	graduat	home
depart	page	lectur	problem	system	fax	student
interest	project	note	lab	research	mathemat	assign
professor	home	solut	read	univers	graphic	class
architectur	work	exam	file	program	school	mail
graduat	student	final	code	work	multimedia	homework
technolog	softwar	instructor	schedul	engin	internt	lectur
confer	parallel	chapter	topic	softwar	visual	grade

Fuzzy K-medoids

Table 4.14: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with fuzzy K-medoids algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*			
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3	Cluster 4
comput	program	research	assign	site	page	graduat	comput
scienc	office	engin	class	homepage	program	year	science
system	languag	perform	homework	info	home	artificial	system
page	problem	technolog	lectur	columbia	offic	implement	research
inform	mail	proceed	note	dept	student	colleg	parallel
project	lab	confer	grade	visit	assign	faculti	algorithm
depart	read	member	final	physic	time	librari	network
home	file	intern	solut	archiv	class	knowledg	databas
interest	code	laboratori	due	phd	problem	submit	machin
work	schedul	colleg	exam	tech	homework	interact	intellig

Fuzzy spherical K-means

Table 4.15: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with the fuzzy spherical K-means algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
research	comput	page	method	comput	resum
professor	science	program	tool	science	visitor
associ	system	project	techniqu	system	phd
bill	univers	office	numer	page	pittsburgh
photo	inform	assign	background	research	physic
uncertainti	depart	algorithm	job	program	nyu
bibliographi	home	class	yahoo	project	biologi
linguist	interest	link	challeng	depart	browser
bookmark	work	homework	formula	offic	geometri

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.16: Most frequent terms of each word cluster returned by Dhillon's spectral bi-clustering algorithm and Kluger's spectral bi-clustering algorithm.

Dhillon				*Kluger*		
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3
abstract	assign	comput	page	page	comput	graduat
seminar	class	science	program	home	scienc	laboratori
understand	homework	system	inform	offic	system	mathemat
midterm	lectur	univers	project	student	research	cours
semest	read	research	home	email	program	colleg
prerequisit	hour	depart	offic	assign	inform	educ
textbook	exam	interest	student	time	project	protocol
scheme	note	work	languag	class	depart	faculti
matlab	solut	engin	time	link	parallel	submit
quiz	due	softwar	algorithm	homework	distribut	berkelei

4.3.2 Benchmark data set: the category *science* of 20 newsgroup data set

This experiment is carried out on 20 newsgroup data set, which has already used in Section 3.7.1.
The analysis is carried out using the category *science* (corresponding to the first one following the alphabetic order). As previously stated, the category *science* is characterized by 4 distinct sub-groups: *sci.crypt*, *sci.electronics*, *sci.med* and *sci.space*.
Likewise to the previous application on WebKB data set, the documents have been pre-processed.
Since the true class labels for the document partitions are known in advance, the parameter K_{docs} is fixed equal to 4 for both *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms. However, for the latter one, the parameter K_{words} varies in the interval $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
The two algorithms are evaluated for m assuming values in $[1.1, 2]$, with step 0.1.
In particular, the Purity index and the fuzzy Silhouette index are used to evaluate the goodness of the partitions. As in the previous experiment, the former one is used to evaluate only the document partitions since the true class labels are known a priori. The latter one is used for both document and word partitions .

For both algorithms, the optimal combination of parameters (K_{docs}, K_{words}, m) is selected following the same criteria of the previous experiment on WebKB data set.

The results of the comparison between the standard Dhillon's and Kluger's spectral bi-clustering and our proposed algorithms in combination with fuzzy K-means, fuzzy K-medoids and fuzzy spherical K-means are discussed below.
Before inspecting the details, the parameters identifying the optimal partitions for each version of the fuzzy spectral bi-clustering algorithms are reported in Table 4.17.

Table 4.17: Parameters returning the optimal partitions for all the versions of the *Joint Fuzzy Spectral Bi-clustering* algorithm and the *Sequential Fuzzy Spectral Bi-clustering* algorithm for the category *science*.

Parameters	Fuzzy K-means		Fuzzy Spherical K-means		Fuzzy K-medoids	
	Joint	*Sequential*	*Joint*	*Sequential*	*Joint*	*Sequential*
K_{docs}	4	4	4	4	4	4
K_{words}	4	2	4	3	4	2
m	2	1.4	1.9	2	1.7	1.5

Concerning the two standard methods, Dhillon's and Kluger's spectral bi-clustering algorithms, the optimal number of clusters for the word partition identified by the *Sequential Fuzzy Spectral Bi-Clustering* is always equal to 4.

What emerges from Table 4.17 is that both the versions of the *Sequential Fuzzy Spectral Bi-Clustering* with fuzzy K-means and fuzzy K-medoids return the value 2 as the optimal one for the parameter K_{words}. Moreover, the majority of the versions of the *Joint Fuzzy Spectral Bi-Clustering* are characterized by higher values for m then the *Sequential Fuzzy Spectral Bi-Clustering*.

An overview of the values of the external and internal validity measures for all the identified partitions is reported in Table 4.18, Table 4.19, Table 4.20 and Table 4.21.

Fuzzy K-means

Table 4.18: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy K-means for the category of *science*.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.75	0.56
fuzzy Silhouette (docs)	0.73	0.77
fuzzy Silhouette (word)	0.66	0.93

Fuzzy K-medoids

Table 4.19: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy K-medoids for the category of *science*.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.74	0.64
fuzzy Silhouette (docs)	0.71	0.65
fuzzy Silhouette (word)	0.65	0.92

Fuzzy spherical K-means

Table 4.20: Comparison, in terms of validity indexes, of the fuzzy spectral bi-clustering algorithms using fuzzy spherical K-means for the category of *science*.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
Purity (docs)	0.82	0.46
fuzzy Silhouette (docs)	0.68	0.53
fuzzy Silhouette (word)	0.63	0.70

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.21: Comparison, in terms of validity indexes, of Dhillon's and Kluger's spectral bi-clustering algorithms for the category of *science*.

Indexes	*Dhillon*	*Kluger*
Purity (docs)	0.62	0.31
Silhouette (docs)	0.45	0.42
Silhouette (word)	0.42	0.65

What emerges from the aforementioned tables is that the Purity index assumes always higher values for the *Joint Fuzzy Spectral Clustering* compared to the *Sequential Fuzzy Spectral Clustering*: 0.75 vs. 0.56 with fuzzy K-means, 0.74 vs. 0.64 with fuzzy K-medoids and 0.82 vs. 0.46 with fuzzy spherical K-means. Dhillon and Kluger versions are also characterized by a similar situation: 0.62 vs. 0.31. The main difference compared with the fuzzy counterparts is that the values of the Purity index are sensibly lower.
Moreover, the *Joint Fuzzy Spectral Clustering* returns document partitions characterized by slightly higher values of the fuzzy Silhouette index compared to the ones obtained from the application of the *Sequential Fuzzy Spectral Clustering* algorithm (apart from the version with fuzzy K-means).
On the contrary, all the versions of the *Sequential Fuzzy Spectral Clustering* algorithm return noticeably better word partitions, in terms of the fuzzy Silhouette index, compared to the *Joint Fuzzy Spectral Clustering*. Indeed, the values of the index for the *Sequential Fuzzy Spectral Clustering* algorithm combined with fuzzy K-means and fuzzy K-medoids are higher than 0.9.
The same considerations hold also for Dhillon and Kluger versions: the former one returns better document partitions in terms of the Silhouette index (0.45 vs. 0.42); the latter one returns better column partitions in terms of the Silhouette index (0.65 vs. 0.42).

The analysis of the structures of the membership degrees for each identified partition is also carried out. The results are visible in Table 4.22 and in Table 4.23 for fuzzy K-means; in Table 4.24 and in Table 4.25 for fuzzy K-medoids; in Table 4.26 and in Table 4.27 for fuzzy spherical K-means.
With reference to the document partitions, the results for fuzzy K-medoids and fuzzy spherical K-means show that the *Joint Fuzzy Spectral Clustering* algorithm return average values of the membership degrees slightly higher compared to the corresponding ones of the *Sequential Fuzzy Spectral Clustering*. In general, the membership degrees structures for all the versions of the two proposed methods present an overall good behaviour.

Fuzzy K-means

Table 4.22: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.42	0.99	0.81
Cluster 2	0.36	0.99	0.74
Cluster 3	0.33	0.99	0.77
Cluster 4	0.45	0.99	0.92
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.46	1	0.83
Cluster 2	0.49	1	0.97
Cluster 3	0.40	1	0.86
Cluster 4	0.40	1	0.81

Table 4.23: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.33	0.99	0.77
Cluster 2	0.32	0.99	0.70
Cluster 3	0.31	0.99	0.78
Cluster 4	0.34	0.99	0.79
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.99
Cluster 2	0.50	1	0.88

Fuzzy K-medoids

Table 4.24: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.38	0.99	0.83
Cluster 2	0.49	0.99	0.92
Cluster 3	0.42	0.99	0.82
Cluster 4	0.48	0.99	0.87
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.47	1	0.94
Cluster 2	0.32	1	0.78
Cluster 3	0.33	1	0.72
Cluster 4	0.33	1	0.70

Table 4.25: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.35	1	0.86
Cluster 2	0.39	0.99	0.87
Cluster 3	0.35	1	0.80
Cluster 4	0.35	1	0.85
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.83
Cluster 2	0.50	1	0.81

Fuzzy spherical K-means

Table 4.26: Main statistics of the membership degrees for the **document** partitions obtained by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.41	1	0.90
Cluster 2	0.41	1	0.94
Cluster 3	0.45	0.99	0.80
Cluster 4	0.44	1	0.93
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.31	0.99	0.80
Cluster 2	0.31	0.99	0.70
Cluster 3	0.30	0.99	0.71
Cluster 4	0.30	0.99	0.70

Table 4.27: Main statistics of the membership degrees for the **word** partitions obtained by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms for the category *science*.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.41	1	0.88
Cluster 2	0.41	1	0.89
Cluster 3	0.44	0.99	0.80
Cluster 4	0.41	1	0.90
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.34	0.99	0.76
Cluster 2	0.35	0.99	0.92
Cluster 3	0.34	0.99	0.71

To conclude, an inspection of the most frequent terms in each word cluster identified by the proposed fuzzy spectral bi-clustering algorithms is carried out. Considering the 4 sub-categories for the articles falling under the category of *science* (*sci.crypt*, *sci.electronics*, *sci.med* and *sci.space*), the concepts that we have identified, according to the first ten most frequent words of each word cluster, can be easily assimilated to the original classification. Indeed, the main concepts are related to: "computer & technology", "health", "medicine", "space", "research" and "airline/aircraft management".
In particular, the *Joint Fuzzy Spectral Bi-Clustering* in combination with fuzzy K-medoids is the method returning more understandable word clusters. On the contrary, among the proposed fuzzy spectral bi-clustering methods, *Joint Fuzzy Spectral Bi-Clustering* in combination with fuzzy K-means is the version whose word clusters are characterized by a lower level of clearness.
Concerning the *Sequential Fuzzy Spectral Bi-Clustering*, maybe due to the smaller number of word clusters identified by the parameter K_{words}, no particular difficulties have occurred in the identification of the main "concepts".
Following the same format of the previous sections, an overview of the main "concepts" emerging in each word cluster is presented in Table 4.28.

Table 4.28: Main concepts conveyed by the most ten frequent terms for each cluster in the identified partitions of the fuzzy spectral bi-clustering algorithms.

Fuzzy K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	Not clearly distinguishable	"computer & technology"
Cluster 2	"computer & technology"	"health"
Cluster 3	Not clearly distinguishable	-
Cluster 4	"airline/aircraft management"	-
Fuzzy K-medoids		
Cluster	*Joint*	*Sequential*
Cluster 1	"medicine"	"computer & technology"
Cluster 2	"computer & technology"	"research"
Cluster 3	"research"	-
Cluster 4	"space"	-
Fuzzy spherical K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	"computer & technology"	"computer & technology"
Cluster 2	"health"	"medicine"
Cluster 3	"research"	"space"
Cluster 4	Not clearly distinguishable	-

Fuzzy K-means

Table 4.29: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with fuzzy K-means algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
agenc	encrypt	article	data	encrypt	drug
american	byte	algorithm	design	effect	cholesterol
center	comput	devic	control	comput	alcohol
chang	develop	experi	connect	develop	disclaim
box	chip	correct	batteri	algorithm	danger
cancer	diseas	ask	california	file	difficult
flight	electron	galileo	fuel	anonym	die
disk	code	delta	air	circuit	egg
cryptographi	file	determin	earth	attack	action
energi	crypto	estim	amateur	disk	advis

Fuzzy K-medoids

Table 4.30: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with fuzzy K-medoids algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
effect	encrypt	write	space	comput	edu
medic	chip	article	nasa	compani	email
studi	number	system	orbit	devic	articl
patient	secur	peopl	launch	engin	book
diseas	phne	work	earth	digit	document
drug	mail	year	mission	attack	analysi
food	comput	inform	satellit	disk	class
doctor	technolog	univers	shuttle	byte	access
health	internet	research	spacecraft	decrypt	discov
cancer	algorithm	question	air	experi	answer

Fuzzy spherical K-means

Table 4.31: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with the fuzzy spherical K-means algorithm.

Joint Fuzzy Spectral				*Sequential Fuzzy Spectral*		
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3
encrypt	diet	book	control	code	diseas	control
base	check	address	engin	build	drug	compani
electron	diseas	answer	earth	email	diet	earth
file	form	author	batteri	circuit	aid	devic
email	attempt	chemestri	california	digit	fever	engin
decrypt	chronic	columbia	air	cryptographi	error	area
futur	fit	develop	complet	cryptosystem	fraction	flight
export	brain	audit	express	band	deal	board
agenc	finger	depart	danger	american	challeng	explor
commerci	cell	colleg	action	avoid	dose	astronomi

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.32: Most frequent terms of each word cluster returned by Dhillon's spectral bi-clustering algorithm and Kluger's spectral bi-clustering algorithm.

Dhillon				*Kluger*			
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3	Cluster 4
find	article	engin	design	encrypt	difficult	familiar	edu
compani	algorithm	electron	control	chip	action	aggreg	articl
commun	devic	direct	access	cryptographi	danger	fusion	find
exampl	experi	charg	earth	digit	acetaminophen	convei	develop
center	enforc	establish	activ	degree	cholesterol	explanatori	center
chang	correct	energi	cnnect	archiv	elbow	debugg	answer
acid	determin	cost	batteri	error	biostatist	disregard	board
cancer	databas	bank	air	backup	fingernail	employ	correct
area	cellular	billion	damag	challeng	deficit	economi	applic
distribut	excel	cut	expect	ascii	frequent	consortium	author

4.3.3 Real data set: Trump and Clinton speeches

An evaluation of the *Joint Fuzzy Spectral Bi-clustering* and the *Sequential Fuzzy Spectral Bi-clustering* algorithms on real data is also carried out. For this purpose, a publicly available data set containing 118 individual speeches by Donald Trump and Hillary Clinton, during their electoral campaign of 2016, is analysed. In this setting, neither the number of clusters for the partition of the documents, nor the number of clusters for the partition of the words are known in advance. Consequently, for all the clustering algorithms analysed, both the parameters K_{docs} and K_{words} are let vary in the set $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$, for a total of 81 different combinations of parameters. For each one of these combinations, the parameter m assumes values in the interval $[1.1, 2]$ with step 0.1. The optimal combination of values (K_{docs}, K_{words}, m) is the one maximizing the fuzzy Silhouette index.

As in the previous section, a comparison between the standard methods from Dhillon and Kluger and the *Joint* and the *Sequential Fuzzy Spectral Bi-clustering* algorithms, combined with fuzzy K-means, fuzzy K-medoids and fuzzy spherical K-means, is carried out.
In this analysis, the structure of the partitions in terms of the membership degrees and the main "concepts" conveyed by the most frequent terms in each word cluster are also evaluated.
Before inspecting the clustering results, an overview of the optimal parameters selected for each method is presented in Table 4.33.

Table 4.33: Parameters returning the optimal partitions for all the versions of the *Joint Fuzzy Spectral Bi-clustering* algorithm and the *Sequential Fuzzy Spectral Bi-clustering* algorithm on the corpus of Trump and Clinton speeches.

Parameters	Fuzzy K-means		Spherical Fuzzy K-means		Fuzzy K-medoids	
	Joint	*Sequential*	*Joint*	*Sequential*	*Joint*	*Sequential*
K_{docs}	2	2	2	3	2	2
K_{words}	2	2	2	2	2	2
m	1.7	1.8	1.8	1.9	2	1.5

Using Dhillon's method, the optimal number of clusters for the partitions of the documents and the words is equal to 4. On the other hand, using Kluger's method, the optimal number of clusters for the document partition is 2 while the optimal one for the word partition is 5.

What emerges from Table 4.33 is that all the versions of two fuzzy spectral bi-clustering algorithms are pretty robust in the identification of the optimal partitions. Indeed, except from the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with the fuzzy spherical K-means, the optimal number of clusters

for both the document and the word partitions is always equal to 2. On the contrary, the two standard methods return very different results. A certain homogeneity is identifiable also in the choice of the fuzziness parameter m. Indeed, apart from the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with the fuzzy K-medoids, all the other methods return a value of m ranging between 1.7 and 2.

The first results of the comparative analysis are reported in Table 4.34, Table 4.35, Table 4.36 and Table 4.37.

Fuzzy K-means

Table 4.34: Comparison of the fuzzy spectral bi-clustering algorithms on real data using fuzzy K-means.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
fuzzy Silhouette (docs)	0.94	0.73
fuzzy Silhouette (word)	0.89	0.94

Fuzzy K-medoids

Table 4.35: Comparison of the fuzzy spectral bi-clustering algorithms on real data using fuzzy K-medoids.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
fuzzy Silhouette (docs)	0.94	0.72
fuzzy Silhouette (word)	0.89	0.93

Fuzzy spherical K-means

Table 4.36: Comparison of the fuzzy spectral bi-clustering algorithms on real data using fuzzy spherical K-means.

Indexes	*Joint Fuzzy Spectral*	*Sequential Fuzzy Spectral*
fuzzy Silhouette (docs)	0.94	0.72
fuzzy Silhouette (word)	0.75	0.69

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.37: Comparison Dhillon's and Kluger's spectral bi-clustering algorithms on real data.

Indexes	*Dhillon*	*Kluger*
Silhouette (docs)	0.61	0.50
Silhouette (word)	0.58	0.65

What emerges is that, the *Joint Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means and fuzzy K-medoids return the same values of

the fuzzy Silhouette index for both the document and the word partitions. More generally, it is possible to observe that for each version of the two fuzzy spectral bi-clustering methods, the fuzzy Silhouette indexes calculated on the document partitions are always higher than 0.7.
The *Joint Fuzzy Spectral Bi-clustering* algorithm returns always an average Silhouette index value of 0.94 for all the document partitions, identifying a better behaviour than the *Sequential Fuzzy Spectral Bi-clustering*.
A reverse situation can be noticed for the word partitions where the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means and fuzzy K-medoids returns values of the average fuzzy Silhouette index higher than the corresponding ones from the *Joint Fuzzy Spectral Bi-clustering* method (0.94 vs. 0.89 with fuzzy K-means and 0.93 vs. 0.89 with fuzzy K-medoids).
More generically, all the versions of the two methods present a very similar behaviour (especially when combined with fuzzy K-means and fuzzy K-medoids).
The crisp Dhillon's and Kluger's methods can not be evaluated in terms of the average fuzzy Silhouette index; however, the corresponding average Silhouette indexes for the document and the word partitions assume lower values compared to their fuzzy counterparts.

Each version of the *Joint Fuzzy Spectral Bi-clustering* algorithm and the *Sequential Fuzzy Spectral Bi-clustering* algorithm identifies overlapping structures for both the document and the word partitions, whose results are reported in Table 4.38, Table 4.39, Table 4.40, Table 4.41, Table 4.42 and Table 4.43.
In this context, the analysis highlights that all the versions of both methods return document partitions characterized by average membership degrees almost always higher than 0.75. In some cases, the average membership degrees for the identified partitions are even higher than 0.9. Concerning the word partitions, the *Joint Fuzzy Spectral Bi-clustering* algorithm is mainly characterized by higher average membership degrees compared to the ones from the *Sequential Fuzzy Spectral Bi-clustering*.

Fuzzy K-means

Table 4.38: Main statistics of the membership degrees for the **document** partitions obtained on real data by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.53	0.99	0.98
Cluster 2	0.52	0.99	0.82
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.54	1	0.86
Cluster 2	0.53	1	0.90

Table 4.39: Main statistics of the membership degrees for the **word** partitions obtained on real data by using fuzzy K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.95
Cluster 2	0.50	1	0.93
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.75
Cluster 2	0.50	1	0.97

Fuzzy K-medoids

Table 4.40: Main statistics of the membership degrees for the **document** partitions obtained on real data by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.52	0.99	0.97
Cluster 2	0.53	0.99	0.78
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.94
Cluster 2	0.51	1	0.91

Table 4.41: Main statistics of the membership degrees for the **word** partitions obtained on real data by using fuzzy K-medoids in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.93
Cluster 2	0.50	1	0.91
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.76
Cluster 2	0.50	1	0.77

Fuzzy spherical K-means

Table 4.42: Main statistics of the membership degrees for the **document** partitions obtained on real data by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.54	0.99	0.97
Cluster 2	0.51	0.99	0.95
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.48	0.99	0.87
Cluster 2	0.47	0.99	0.88
Cluster 3	0.39	0.99	0.76

Table 4.43: Main statistics of the membership degrees for the **word** partitions obtained on real data by using fuzzy spherical K-means in, respectively, *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* algorithms.

Joint Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	1	0.94
Cluster 2	0.50	1	0.95
Sequential Fuzzy Spectral			
Cluster	Min.memb.deg.	Max.memb.deg.	Av.memb.deg.
Cluster 1	0.50	0.99	0.87
Cluster 2	0.51	0.99	0.78

Then, an inspection of the content conveyed by the identified partitions is carried out. Against this background, the first ten most frequent words in each cluster are analysed. What emerges from the results of the fuzzy spectral bi-clustering algorithms is that at least one cluster in each partition is characterized by the topic of "propaganda", mainly identifiable by the words "country", "great", "vote", "campaign", "election", "together", "president", "future", "support", "America", "change".
The other contents emerging from the first ten most frequent words can be associated to "trade and international relationships", "education system", "internal affairs" and "military and economic affairs".
In order to better visualize the content of each cluster emerging from our analysis, for all the versions of the proposed fuzzy spectral bi-clustering algorithms, a summary of them is provided in Table 4.44.

Table 4.44: Main concepts conveyed by the most ten frequent terms for each cluster in the identified partitions of the fuzzy spectral bi-clustering algorithms.

Fuzzy K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	"propaganda"	"Trade and international relationships"
Cluster 2	"education system"	"propaganda"
Fuzzy K-medoids		
Cluster	*Joint*	*Sequential*
Cluster 1	"propaganda"	"Trade and international relationships"
Cluster 2	"internal affairs"	"military and economic affairs"
Fuzzy spherical K-means		
Cluster	*Joint*	*Sequential*
Cluster 1	"education system"	"propaganda "
Cluster 2	"propaganda"	"Trade and international relationships"

The detailed lists of the first ten most frequent words for each word cluster are reported in Table 4.45, Table 4.46, Table 4.47, Table 4.48 and Table 4.49.

Fuzzy K-means

Table 4.45: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with K-means algorithm on real data.

Joint Fuzzy Spectral		*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 1	Cluster 2
countri	economi	peopl	presid
great	young	job	work
job	hope	year	campaign
back	kid	world	togeth
vote	colleg	win	election
unit	nuclear	big	famili
year	debt	trade	hard
america	student	monei	future
support	value	militari	support
histori	gun	wall	life

Fuzzy K-medoids

Table 4.46: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with fuzzy K-medoids algorithm on real data.

Joint Fuzzy Spectral		*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 1	Cluster 2
peopl	election	countri	peopl
countri	famili	great	work
great	economi	job	care
job	right	obama	business
back	senat	change	veteran
america	service	govern	war
time	student	stop	chance
vote	coorporation	mexico	economi
change	convention	disaster	republican
important	deserve	tax	nation

Fuzzy spherical K-means

Table 4.47: Most frequent terms of each word cluster returned by *Joint Fuzzy Spectral Bi-clustering* and *Sequential Fuzzy Spectral Bi-clustering* in combination with the fuzzy spherical K-means algorithm on real data.

Joint Fuzzy Spectral		*Sequential Fuzzy Spectral*	
Cluster 1	Cluster 2	Cluster 1	Cluster 2
america	peopl	america	peopl
campaign	vote	work	great
pay	world	campaign	job
futur	win	togeth	world
business	audience	election	monei
support	care	famili	audience
school	trade	futur	change
education	remember	support	isi
children	tax	women	militari
chance	disaster	health	bill

Dhillon's and Kluger's spectral bi-clustering algorithms

Table 4.48: Most frequent terms of each word cluster returned by Dhillon's spectral bi-clustering algorithm on real data.

Dhillon			
Cluster 1	Cluster 2	Cluster 3	Cluster 4
audience	presid	colleg	win
care	work	right	monei
thing	campaign	equal	trade
love	togeth	black	wall
remember	america	challenge	deal
day	election	grateful	countri
job	futur	church	mexico
obama	famili	universiti	disaster
militari	women	muslim	china
change	economi	biden	border

Table 4.49: Most frequent terms of each word cluster returned by Kluger's spectral bi-clustering algorithm on real data.

Kluger				
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
america	vote	good	peopl	presid
life	audience	pay	countri	togeth
secretari	change	job	great	election
candidat	plan	important	job	famili
house	govern	tax	back	hard
honor	nation	business	time	futur
leadership	fight	veteran	world	economi
men	energi	manufactur	big	nuclear
presid	administration	war	monei	opportuniti
attack	education	power	care	weapon

In conclusion, since the document and the word partitions are characterized by the same number of clusters, we have tried to understand if there is a link between the document and the word clusters. In other words, we tried to understand if, for instance, the documents within the first cluster are somehow connected with the "concepts" conveyed by the first word cluster.
However, from the inspection of the documents and the corresponding word clusters we cannot reach any general conclusion, especially when there is not a one to one correspondence between the word and the document clusters.
Nevertheless, for the documents for which is possible to identify a connection with the corresponding word clusters, it is interesting to observe the behaviour of the membership degrees.
Two illustrative examples, one for the *Joint Fuzzy Spectral Bi-clustering* algorithm and the other one for the *Sequential Fuzzy Spectral Bi-clustering*, are reported below.

In particular, the first example is about the assignment of three documents under the *Joint Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means: the first document is assigned to the second cluster with a membership degree of 0.97, the second document is assigned to the first cluster with a membership degree of 0.99 and the third document is assigned to the first cluster with a lower membership degree equal to 0.50.
Figure 4.2, Figure 4.3 and Figure 4.4 report the wordclouds of the three aforementioned documents.

Figure 4.2: Wordcloud of the document assigned to the second cluster with a membership degree of 0.97 under the *Joint Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "education system".

Figure 4.3: Wordcloud of the document assigned to the first cluster with a membership degree of 0.99 under the *Joint Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "propaganda".

Figure 4.4: Wordcloud of the document assigned to the first cluster with a membership degree of 0.50 under the *Joint Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "propaganda".

How it is possible to notice, the first document clearly conveys the concept of "education system", as well as the second document clearly conveys the concept of "propaganda". On the other hand the third document is less easily identifiable within one of the two categories. In this example the "strength" of the conveyed concepts is well represented by the corresponding membership degrees of the document partitions. Indeed, the third document is assigned to the first cluster (that could be associated to the first word cluster "propaganda") with a membership degree equals to 0.5. On the contrary the first two documents, which are more easily understandable from the wordclouds, have less fuzzier membership degrees (respectively, 0.97 and 0.99).

The second example considers three documents classified under the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination always with fuzzy K-means. Also in this example the membership degrees behaviour of the document partition reflect the "strenght" of the concepts conveyed by the corresponding word clusters. Indeed, the first document is assigned to the second cluster with a membership degree of 0.85. As it is possible to notice from its wordcloud in Figure 4.5, it well reflects the concept of "propaganda".

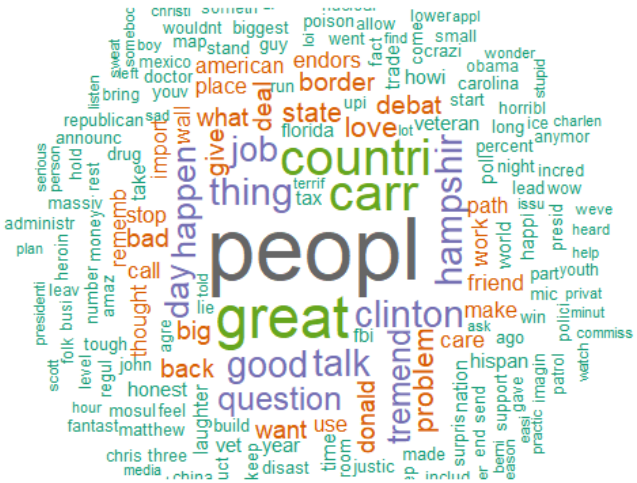


Figure 4.5: Wordcloud of the document assigned to the second cluster with a membership degree of 0.85 under the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "propaganda".

The second document is assigned to the first cluster with a membership degree of 0.97. The corresponding word cluster is about "trade and international relationships". Its wordcloud is reported in Figure 4.6.

To conclude, the third and last document is the one characterized by a fuzzier membership degree: indeed, it is assigned to the first cluster with a membership degree of 0.68. Its wordcloud is displayed in Figure 4.7.

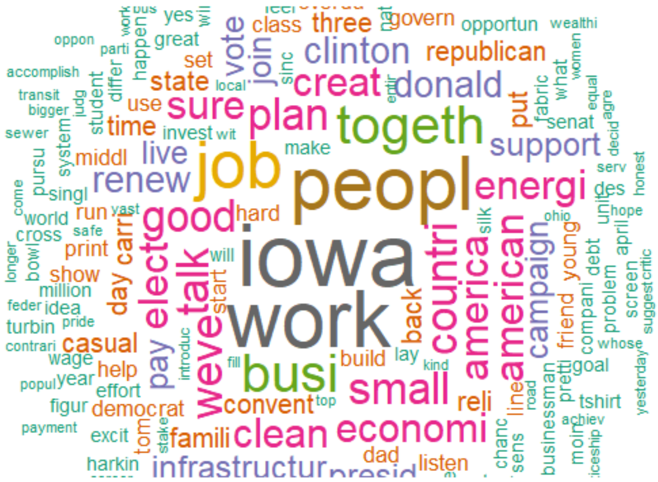


Figure 4.6: Wordcloud of the document assigned to the first cluster with a membership degree of 0.97 under the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "trade and international relationships".

Figure 4.7: Wordcloud of the document assigned to the first cluster with a membership degree of 0.68 under the *Sequential Fuzzy Spectral Bi-clustering* algorithm in combination with fuzzy K-means. The corresponding word cluster conveys the concept of "trade and international relationships".

4.4 Concluding remarks

In this chapter, starting from the state-of-art bi-clustering methods introduced by Dhillon and Kluger, two novel fuzzy spectral bi-clustering algorithms are presented and discussed. From the experiments carried out on both benchmark and real data sets, the new methods lead to the identification of fuzzy partitions for both the documents and the words. Moreover, the two proposed methods are characterized by an higher accuracy, in terms of the Purity index, compared to their crisp counterparts.

Chapter 5

Conclusions and open problems

The development of unsupervised classification algorithms for analysing text documents represents an interesting field of research for several text mining tasks. This thesis wanted to give a contribution in this direction, focusing mainly on the construction of fuzzy spectral clustering methods for the classification of text data.

Indeed, nowadays societies are developing a huge amount of text data (e.g. social networks, web pages, electronic document archives...) that if are not elaborated and analysed correctly, there may be the risk of loosing important and precious information which can be used for further analyses.
The aim of this work consisted in providing a further and little step in the field of research of text classification, presenting different novel fuzzy (bi-)clustering methods which could be of support in decision making processes.

Starting off from this base, the thesis begins with an introduction (Chapter 1) and continues with a review of the main approaches already available in literature in the document clustering framework, providing an overview of the state-of-art methods (Chapter 2).
The following chapter (Chapter 3) contains both methodological proposals and empirical evaluations on benchmark and real data sets.
More specifically, a novel fuzzy spectral clustering algorithm for unsupervised classification of text data is presented. The new method exploits the use of fuzzy K-medoids and allows for an overlapping between clusters. The novel method has been initially used in combination with the Spectrum string kernel and then with KS^2M similarity, giving rise to a (second) novel fuzzy spectral clustering algorithm. Nevertheless, in order to improve the accuracy of the clustering results, a new similarity measure for text documents, $\mathbf{S}^*$, to use in combination with the novel aforementioned fuzzy spectral clustering algorithm is presented in Section 3.5.

The second part of the thesis (Chapter 4), focuses on Dhillon's and Kluger's state-of-arts bi-clustering algorithms. In particular, starting from them, two novel bi-clustering methods developed under a fuzzy set-up, the *Joint Fuzzy Spectral Bi-clustering* and the *Sequential Fuzzy Spectral Bi-clustering*, are proposed and discussed. Chapter 4 ends with applications involving benchmark and real data sets.
Finally, the last chapter (Chapter 5) holds the main conclusions.

Therefore, our main contributions in this thesis are: 1) the identification of three novel fuzzy spectral clustering algorithms for text data, 2) the development of a novel similarity measure for textual documents and 3) the proposal of two novel fuzzy bi-clustering algorithms using spectral methods.

As previously mentioned, this thesis seeks to produce a little contribution in the desired direction, but there is still plenty of room for improvement. Considering all the possible open problems, one that certainly will interest our attention in future is related to the inclusion, during the clustering approach of the novel fuzzy spectral bi-clustering algorithm with $\mathbf{S}^*$ similarity, of the semantic content conveyed by the words.
For this purpose an English thesaurus, such as the publicly available WordNet (Miller, 1995), can be employed. Indeed, this lexical database provides valuable information on the semantic of verbs, adjectives, nouns and adverbs. These items are clustered into groups, called *synsets* (cognitive synonyms), conveying clear and distinct concepts. This information can be used as metadata in the clustering process, providing further insights for an accurate classification of the documents.

On the other hand, concerning the fuzzy spectral bi-clustering methods, the potential for improvements is even wider than in the one-mode clustering setup. Indeed, even if the already available methods provide a way to identify efficiently the co-clustering structures of both words and documents, in our future works we plan to investigate the development of a new version of *Joint Fuzzy Spectral Bi-clustering*. The new algorithm should be characterized by a simultaneous identification of the document and the word partitions but, at the same time, should provide the selection of a distinct number of clusters for the two partitions. Furthermore, even in the bi-clustering setup, another issue that will be subject of future exploration is the inclusion of the semantic content of the terms for the identification of the word clusters in order to improve the clustering performance.

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163–222).

Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding* (Tech. Rep.). Stanford.

Babu, G. P., & Murty, M. N. (1993). A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm. *Pattern recognition letters*, *14*(10), 763–769.

Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances. In *Proceedings of scei seoul conferences* (pp. 174–179).

Bao, L., Tang, S., Li, J., Zhang, Y., & Ye, W.-p. (2008). Document clustering based on spectral clustering and non-negative matrix factorization. In *International conference on industrial, engineering and other applications of applied intelligent systems* (pp. 149–158).

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems 14 (nips 2001)* (Vol. 14, pp. 585–591).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, *7*(85), 2399–2434.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Springer.

Bisht, S., & Paul, A. (2013). Document clustering: a review. *International Journal of Computer Applications*, *73*(11).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Campello, R. J. (2007). A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, *28*(7), 833–841.

Campello, R. J., & Hruschka, E. R. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, *157*(21), 2858–2875.

Cano, C., Adarve, L., López, J., & Blanco, A. (2007). Possibilistic approach for biclustering microarray data. *Computers in biology and medicine*, *37*(10), 1426–1436.

Casella, G., & George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, *46*(3), 167–174.

Chung, F. R. (1997). *Spectral graph theory* (Vol. 92). American Mathematical Soc.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cozzolino, I., & Ferraro, M. B. (2022). Document clustering. *Wiley Interdisciplinary Reviews: Computational Statistics*, e1588.

Cozzolino, I., Ferraro, M. B., & Winker, P. (2021). A fuzzy clustering approach for textual data. In *Book of short papers sis 2021* (pp. 770–776).

Cui, X., Potok, T. E., & Palathingal, P. (2005). Document clustering using particle swarm optimization. In *Proceedings 2005 ieee swarm intelligence symposium, 2005. sis 2005.* (pp. 185–191).

de Finetti, B. (1969). Sulla proseguibilità di processi aleatori scambiabili.

De Finetti, B. (1972). Probability, induction and statistics: The art of guessing.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh acm sigkdd international conference on knowledge discovery and data mining* (pp. 269–274).

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 551–556).

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine learning*, *42*(1), 143–175.

Ding, C. H., He, X., Zha, H., Gu, M., & Simon, H. D. (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 ieee international conference on data mining* (pp. 107–114).

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Mhs'95. proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43).

Fahad, S. A., & Yafooz, W. M. (2017). Review on semantic document clustering. *International Journal of Contemporary Computer Research*, *1*(1), 14–30.

Ferraro, M. B., & Giordani, P. (2020). Soft clustering. *Wiley Interdisciplinary Reviews: Computational Statistics*, *12*(1), e1480.

Ferraro, M. B., Giordani, P., & Serafini, A. (2019). fclust: An r package for fuzzy clustering. *R J.*, *11*(1), 198.

Ferraro, M. B., Giordani, P., & Vichi, M. (2021). A class of two-mode clustering algorithms in a fuzzy setting. *Econometrics and Statistics*, *18*, 63–78.

Fraley, C., & Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, *41*(8), 578–588.

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, *97*(458), 611–631.

Gelfand, A. E., & Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, *85*(410), 398–409.

Green, N., Rege, M., Liu, X., & Bailey, R. (2011). Evolutionary spectral co-clustering. In *The 2011 international joint conference on neural networks* (pp. 1074–1081).

Guan, J., Qiu, G., & Xue, X.-Y. (2005). Spectral images and features co-clustering with application to content-based image retrieval. In *2005 ieee 7th workshop on multimedia signal processing* (pp. 1–4).

Guattery, S., & Miller, G. L. (1994). *On the performance of spectral graph partitioning methods.* (Tech. Rep.). CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.

Hagen, L., & Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, *11*(9), 1074–1085.

Hart, P. E., Stork, D. G., & Duda, R. O. (2000). *Pattern classification.* Wiley Hoboken.

He, X., Cai, D., Shao, Y., Bao, H., & Han, J. (2010). Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering*, *23*(9), 1406–1418.

Hornik, K., Feinerer, I., Kober, M., & Buchta, C. (2012). Spherical k-means clustering. *Journal of statistical software*, *50*(10), 1–22.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (nzcsrsc2008), christchurch, new zealand* (Vol. 4, pp. 9–56).

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, *37*, 547–579.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264–323.

Janani, R., & Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, *134*, 192–200.

Jivani, A. G., et al. (2011). A comparative study of stemming algorithms. *International journal of computer technology and applications*, *2*(6), 1930–1938.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137–142).

Joachims, T., Cristianini, N., & Shawe-Taylor, J. (2001). Composite kernels for hypertext categorisation. In *Icml* (Vol. 1, pp. 250–257).

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11–21.

Karatzoglou, A., & Feinerer, I. (2007). Text clustering with string kernels in r. In *Advances in data analysis, proceedings of the 30th annual conference of the gesellschaft fur klassifikation e.v., freie universitat berlin* (pp. 91–98).

Karatzoglou, A., Smola, A., Hornik, K., & Karatzoglou, M. A. (2019). Package 'kernlab'. *CRAN R Project*.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab-an s4 package for kernel methods in r. *Journal of statistical software*, *11*(9), 1–20.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kluger, Y., Basri, R., Chang, J. T., & Gerstein, M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, *13*(4), 703–716.

Knittel, J., Koch, S., & Ertl, T. (2021). Efficient sparse spherical k-means for document clustering. In *Proceedings of the 21st acm symposium on document engineering* (pp. 1–4).

Korenius, T., Laurikkala, J., Järvelin, K., & Juhola, M. (2004). Stemming and lemmatization in the clustering of finnish text documents. In *Proceedings of the thirteenth acm international conference on information and knowledge management* (pp. 625–633).

Krishna, K., & Murty, M. N. (1999). Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *29*(3), 433–439.

Krovetz, R. (2000). Viewing morphology as an inference process. *Artificial intelligence*, *118*(1-2), 277–294.

Kumar, A., & Daumé, H. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 393–400).

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2008). Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, *31*(4), 721–735.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995* (pp. 331–339). Elsevier.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Lenz, D., & Winker, P. (2020). Measuring the diffusion of innovations with paragraph vector topic models. *PLoS ONE*, *15*(1), e0226685.

Lewis, D. (1997). Reuters-21578 text categorization test collection, distribution 1.0. *http://www. research/. att. com*.

Liu, J., Cai, D., & He, X. (2010). Gaussian mixture model with local consistency. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 24, pp. 512–517).

Liu, N., Chen, F., & Lu, M. (2013). Spectral co-clustering documents and words using fuzzy k-harmonic means. *International Journal of Machine Learning and Cybernetics*, *4*(1), 75–83.

Lo, R. T.-W., He, B., & Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. *Journal on Digital Information Management*, *3*, 3–8.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, *2*(Feb), 419–444.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th berkeley symp. math. statist. probability* (pp. 281–297).

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. *Annual review of statistics and its application*, *6*(1), 355–378.

Miao, D., Duan, Q., Zhang, H., & Jiao, N. (2009). Rough set based hybrid algorithm for text classification. *Expert Systems with Applications*, *36*(5), 9168–9174.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Mohar, B. (1997). Some applications of laplace eigenvalues of graphs. In *Graph symmetry* (pp. 225–275). Springer.

Nazeer, K. A., & Sebastian, M. (2009). Improving the accuracy and efficiency of the k-means clustering algorithm. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 1–3).

Ng, Jordan, & Weiss. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, *14*.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems 14 (nips 2001)* (pp. 849–856).

Nguyen, E. (2013). Text mining and network analysis of digital libraries in r. In Y. Zhao & Y. Cen (Eds.), *Data mining applications with r.* (p. 201-213). Academic Press.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification

from labeled and unlabeled documents using em. *Machine learning*, *39*(2), 103–134.

Oellermann, O. R., & Schwenk, A. J. (1991). The laplacian spectrum of graphs. *Graph Theory, c, Appl*, *2*, 871–898.

Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy systems*, *3*(3), 370–379.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems*, *14*(3), 130–137.

Porter, M. F. (2001). *Snowball: A language for stemming algorithms.* Retrieved from `http://snowball.tartarus.org/texts/introduction.html`

Pothen, A., Simon, H. D., & Liou, K.-P. (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, *11*(3), 430–452.

Premalatha, K., & Natarajan, A. (2010). A literature review on document clustering. *Information Technology Journal*, *9*(5), 993–1002.

Puzicha, J., Hofmann, T., & Buhmann, J. M. (2000). A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, *33*(4), 617–634.

Rencher, A. C. (2005). A review of "methods of multivariate analysis, second edition". *IIE Transactions*, *37*(11), 1083-1085.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, *60*, 503–520.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, *3*(4).

Rodrigues, M. M., & Sacks, L. (2004). A scalable hierarchical fuzzy clustering algorithm for text mining. In *Proceedings of the 5th international conference on recent advances in soft computing* (pp. 269–274).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Salton, G. (1971). *The smart retrieval system—experiments in automatic document processing.* Prentice-Hall, Inc.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* McGraw-Hill.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press Cambridge.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1–47.

Shah, N., & Mahajan, S. (2012). Document clustering: a detailed review. *International Journal of Applied Information Systems*, *4*(5), 30–38.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, *5*(1), 3–55.

Sharma, D., & Cse, M. (2012). Stemming algorithms: a comparative study and their analysis. *International Journal of Applied Information Systems*, *4*(3), 7–12.

Shawe-Taylor, J., Cristianini, N., et al. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Shi, & Malik. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 888–905.

Shi, L., Weng, M., Ma, X., & Xi, L. (2010). Rough set based decision tree ensemble algorithm for text classification. *Journal of Computational Information Systems*, *6*(1), 89–95.

Smola, A. J., & Schölkopf, B. (1998). *Learning with kernels* (Vol. 4). Citeseer.

Sneath, P. H., & Sokal, R. R. (1973). *Numerical taxonomy. the principles and practice of numerical classification.* W H Freeman & Co.

Steinbach, M., Karypis, G., & Kumar, V. (2000). *A comparison of document clustering techniques* (Tech. Rep.). University of Minnesota.

Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic author-topic models for information discovery. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 306–315).

Still, S., & Bialek, W. (2004). How many clusters? an information-theoretic perspective. *Neural computation*, *16*(12), 2483–2506.

Stoer, M., & Wagner, F. (1997). A simple min-cut algorithm. *Journal of the ACM (JACM)*, *44*(4), 585–591.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining.* Pearson education.

Tripathy, B., et al. (2019). Fuzzy clustering of sequential data. *International Journal of Intelligent Systems and Applications*, *11*(1), 43.

Van Rijsbergen, C., Harper, D. J., & Porter, M. F. (1981). The selection of good search terms. *Information Processing & Management*, *17*(2), 77–91.

Vapnik, V. (1999). *The nature of statistical learning theory.* Springer science & business media.

Vichi, M. (2001). Double k-means clustering for simultaneous classification of objects and variables. In *Advances in classification and data analysis* (pp. 43–52). Springer.

Vijayarani, S., Ilamathi, M. J., Nithya, M., et al. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Vijaymeena, M., & Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*,

3(2), 19–28.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*(4), 395–416.

Wagner, D., & Wagner, F. (1993). Between min cut and graph bisection. In *International symposium on mathematical foundations of computer science* (pp. 744–750).

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in nlp. In *Coling 1992 volume 4: The 14th international conference on computational linguistics* (pp. 1106–1110).

Wieling, M., & Nerbonne, J. (2009). Bipartite spectral graph partitioning to co-cluster varieties and sound correspondences in dialectology. In *Proceedings of the 2009 workshop on graph-based methods for natural language processing (textgraphs-4)* (pp. 14–22).

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., ... others (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1–37.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, *13*(8), 841–847.

Xu, G., Zong, Y., Dolog, P., & Zhang, Y. (2010). Co-clustering analysis of weblogs using bipartite spectral projection approach. In *International conference on knowledge-based and intelligent information and engineering systems* (pp. 398–407).

Zha, H., He, X., Ding, C., Simon, H., & Gu, M. (2001). Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on information and knowledge management* (pp. 25–32).

Zhang, J.-S., & Leung, Y.-W. (2004). Improved possibilistic c-means clustering algorithms. *IEEE transactions on fuzzy systems*, *12*(2), 209–217.

Zhao, Y., & Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine learning*, *55*(3), 311–331.

Zhao, Y., Karypis, G., & Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data mining and knowledge discovery*, *10*(2), 141–168.