Received: 20 January 2022 | Revised: 31 March 2022 | Accepted: 29 April 2022

DOI: 10.1002/wics.1588

ADVANCED REVIEW

WIREs COMPUTATIONAL STATISTICS WILEY

Document clustering

Irene Cozzolino | **Maria Brigida Ferraro**

Department of Statistical Sciences, Sapienza University of Rome, Rome, Italy

Correspondence
Maria Brigida Ferraro, Department of Statistical Sciences, Sapienza University of Rome, P.le A. Moro 5, 00185, Rome, Italy.
Email: mariabrigida.ferraro@uniroma1.it

Edited by: Yuichi Mori, Commissioning Editor and David Scott, Review Editor and Co-Editor-in-Chief

Abstract

Nowadays, the explosive growth in text data emphasizes the need for developing new and computationally efficient methods and credible theoretical support tailored for analyzing such large-scale data. Given the vast amount of this kind of unstructured data, the majority of it is not classified, hence unsupervised learning techniques show to be useful in this field. Document clustering has proven to be an efficient tool in organizing textual documents and it has been widely applied in different areas from information retrieval to topic modeling. Before introducing the proposals of document clustering algorithms, the principal steps of the whole process, including the mathematical representation of documents and the preprocessing phase, are discussed. Then, the main clustering algorithms used for text data are critically analyzed, considering prototype-based, graph-based, hierarchical, and model-based approaches.

This article is categorized under:

Statistical Learning and Exploratory Methods of the Data Sciences > Clustering and Classification

Statistical Learning and Exploratory Methods of the Data Sciences > Text Mining

Data: Types and Structure > Text Data

KEYWORDS

document clustering, document representation, graph-based methods, hierarchical methods, model-based methods, prototype-based methods, text data

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 et al., 1981). Nowadays, due to the increasing number of text data, document clustering is used for different applications: 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 is useful for improving the efficiency of search engines since the user query are initially compared with the clusters' content 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, such as Jaccard correlation coefficient, Kullback–Leibler divergence, and 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 $D_{KL}\left(P\middle\|Q\right) = \sum_{x \in \mathcal{X}} P(x) \cdot ln\left(\frac{P(x)}{Q(x)}\right)$. 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 emphasizing 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 preprocessing, 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 in 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) analyze also the frequent itemset based clustering approach which consists of a set of techniques that do not require the vector space model representation of the corpus.

In this article, 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 study, we decided to focus on unsupervised classification techniques. However, there are also many supervised text categorization proposals in the literature (for a detailed survey on the main text classification algorithms see, for instance, Aggarwal & Zhai, 2012). In a supervised framework, Support Vector Machines (SVM) (Cortes & 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, and few irrelevant features) and also their robustness, outperforming in this regard other existing methods.

The rest of this article is organized as follows: in-depth studies on the preprocessing and the representation of documents into a multidimensional space are provided in Section 2; the main document clustering techniques for each of the aforementioned categories are introduced and described in Section 3. Finally, Section 4 contains some comments and concluding remarks.

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 analyzed in Section 2.2. Nevertheless, the construction of a VSM requires a preprocessing step that takes place directly on the documents written in natural language. The preprocessing phase aims at removing the noise from text data (e.g., the nonmeaningful terms) and hence reducing the dimensions of the feature-space.

2.1 | Preprocessing

Preprocessing plays a key part in document clustering techniques since it is the very first phase of the entire process. The main steps of preprocessing are: tokenization, filtering, pruning, stemming, and& 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, the complexity of tokenization in languages such as Chinese, characterized by the absence of words, is also discussed.
- 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 et al., 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 <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. Porters stemming algorithm (Porter, 1980, 2001), is one of the most famous stemming technique used for text mining applications. Lemmatization (see, e.g., Korenius et al., 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 preprocessing techniques for text mining applications see Vijayarani et al. (2015).

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 preprocessing). 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 (DTM). Sometimes, even if more rarely, the transposed of the DTM, the Term Document Matrix (TDM), is also considered as the mathematical representation of the collection. In the remaining part of the article, 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 ith document and the jth term is calculated as follows:

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

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

As reported in (Salton & 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 $log\left(tf_{ij}+1\right)$, 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 jth term and the ith document is calculated as follows:

$$w_{ij} = IDF_j \times \frac{tf_{ij} \times (k_1+1)}{tf_{ij} + k_1 \times \left(1 - b + b \times \frac{|\mathbf{d}_i|}{avgLen}\right)}, \tag{2}$$

$$IDF_j = ln\left(\frac{N - df_j + 0.5}{df_j + 0.5} + 1\right), \tag{3}$$

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. k_1 and b are two free parameters that should be properly chosen. Following Manning et al. (2008), k_1 is a nonnegative 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 equal to 0, then no normalization is performed.

In classification problems, where a train-test split of the data is carried out, they 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 obtained 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 et al. (2008), where the authors have investigated the effectiveness of different supervised and unsupervised weighting schemes on two popular benchmark data corpus.

Other approaches consists 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).

However, the vector representation of documents under the VSM suffers from certain challenges. The first one is the high number of features of the DTM, since the size of the vocabulary tends to be quite large even for a moderate number of documents in the collection. A good preprocessing can help in reducing this issue. Second, the DTM is likely to be extremely sparse. In this case, several techniques used to reduce the feature space can be applied on the DTM.

3 | DOCUMENT CLUSTERING METHODS

The document clustering problem consists in partitioning the corpus of N documents, $\mathcal{C} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N\}$ 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 divides 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 observations are grouped around the prototypes. The most widely applied prototype-based algorithms (crisp and soft, respectively) are K-means (MacQueen, 1967) and Fuzzy K-Means (FKM) (Bezdek, 1981).

Despite K-means is considered one of the top 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 centers 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 nonoptimum solutions. 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 calculating 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 et al. (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 makes a cluster. Spectral clustering methods (Ng et al., 2002) are representative of graph-based class. These methodologies rely on the use of an affinity matrix, determining a connection between kernel methods and spectral clustering (see Dhillon et al., 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 3.2 specific kernel functions, used to define affinities between documents, are analyzed.

Hierarchical algorithms aim at identifying a hierarchical set of partitions. A hierarchy is a sequence of partitions such that the sets on lower levels are partitions of the sets of the higher levels. Hierarchies can be visualized as trees (known as *dendograms*). Agglomerative hierarchical clustering (AHC) algorithm (Tan et al., 2006) is representative of this category. For an exhaustive review of hierarchical methods 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 et al. (2019).

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

3.1 | Prototype-based methods

Compared with 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 nonsuitable 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 et al., 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. As the name suggests, spherical K-means takes into account the Euclidean dissimilarities calculated between the vector's projections on the unit sphere.

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 cosine similarity is widely applied in document clustering and it returns better results compared with 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 $\Phi(\mathbf{U},\mathbf{H})$:

$$\Phi(\mathbf{U},\mathbf{H}) = \sum_{i=1}^{N} \sum_{g=1}^{K} u_{ig} \left(1 - cos\left(\mathbf{d}_i, \mathbf{h}_g\right)\right), \tag{4}$$

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:

$$u_{ig} = \begin{cases} 1 & \text{if } i \text{ is allocated to cluster } g, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

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

$$\mathbf{h}_g = \sum_{i=1}^{N} u_{ig} \frac{\mathbf{d}_i}{\left\| \mathbf{d}_i \right\|}. \tag{6}$$

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 (4) 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 et al. (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, in Krishna and Murty (1999) the authors presented genetic K-means algorithm (GKA) for clustering textual documents by identifying 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 et al. (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 four different 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.

3.2 | Graph-based methods

Graph partitioning methods convert the data clustering problem into a graph partitioning problem (Ding et al., 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, the corresponding spectral clustering algorithm has been extensively used when analyzing 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 et al. (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**:

$$\mathbf{L} = \mathbf{D} - \mathbf{S}, \tag{7}$$

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:

$$\mathbf{L}_{\text{norm}} = \mathbf{D}^{-\frac{1}{1}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{1}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}. \tag{8}$$

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 et al., 2002) are 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 p characters appearing one after the other in the text, even though not necessarily contiguously. Consider, for instance, the following three 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:

$$k(\mathbf{d}_i, \mathbf{d}_q) = \sum_{s \in A^*} \text{num}_s(\mathbf{d}_i)\text{num}_s(\mathbf{d}_q)\lambda_s, \tag{9}$$

where A^* is the set of all strings of length p, *num* counts how many times the substrings in A^* appear in the documents $\mathbf{d}_i$ and $\mathbf{d}_q$, and λ_s is a decay factor associated to s 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 p 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 p 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).

A fuzzy version of spectral clustering algorithm to use in combination with text data is provided in Cozzolino et al. (2021) where the standard K-means clustering algorithm is replaced by the corresponding fuzzy counterpart.

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.

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 et al., 2000): all the documents initially belong to a single cluster, then the methodology 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 a single cluster containing all the observations is obtained (Sneath & Sokal, 1973).

A detailed review on hierarchical methods for text data is present in Zhao et al. (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 analyzed.

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 et al., 2000). It is expressed as follows:

$$I_1 = \sum_{g=1}^{K} n_g \left(\frac{1}{n_g^2} \sum_{\mathbf{d}_i, \mathbf{d}_q \in C_g} cos(\mathbf{d}_i, \mathbf{d}_q) \right), \tag{10}$$

where C_g represents the gth 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 et al., 2000)

$$E_1 = \sum_{g=1}^{K} n_g \ \cos(\mathbf{h}_g, \mathbf{h}). \tag{11}$$

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 one measures the similarity of two generic clusters by calculating the maximum of the cosine distance, $\cos_{\text{dist}}$, between the documents for each of the two clusters

$$\Phi_{\text{single.link}}(C_g, C_f) = \max_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} \cos_{\text{dist}}(\mathbf{d}_i, \mathbf{d}_q). \tag{12}$$

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

$$\Phi_{\text{complete.link}}(C_g, C_f) = \min_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} \cos_{\text{dist}}(\mathbf{d}_i, \mathbf{d}_q). \tag{13}$$

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

$$\Phi_{\text{average.link}}(C_g, C_f) = \frac{1}{n_g \cdot n_f} \sum_{\mathbf{d}_i \in C_g, \mathbf{d}_q \in C_f} \cos_{\text{dist}}(\mathbf{d}_i, \mathbf{d}_q). \tag{14}$$

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

3.4 | Model-based methods

Model-based clustering methods rely on the assumption that the population is composed by a mixture of different subpopulations, 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 et al., 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 with 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:

$$f(\mathbf{d}_i, \Psi) = \sum_{g=1}^{K} \pi_g f_g(\mathbf{d}_i|\theta_g), \tag{15}$$

where $\Psi = (\pi_1, ..., \pi_K, \theta_1, ..., \theta_K)$ denotes the global vector of unknown parameters and $f_g(\mathbf{d}_i|\theta_g)$ are the component densities. $\pi = \{\pi_1, ..., \pi_g, ..., \pi_K\}$ represents the vector of prior probabilities for each mixture component (such that $\pi_g > 0 \quad \forall g = 1, ..., K$ 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):

$$\pi(g|\mathbf{d}_i) = \frac{\pi_g f(\mathbf{d}_i|\theta_g)}{\sum_{l=1}^{K} \pi_l f(\mathbf{d}_i|\theta_l)} \quad \forall g = 1, ..., K \quad \forall i = 1, ..., N. \tag{16}$$

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 et al. (2010) and Liu et al., (2010) proposed to add, when analyzing the likelihood function of GMM, a Laplacian regularizer (Belkin et al., 2006) in order to model the underlying submanifold structure. The manifold is modeled 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) (Liu et al., 2010) have been introduced, improving the performance of GMM on text data.

In Nigam et al. (2000), a new algorithm based on the interaction between the EM and the naive Bayes classifier is proposed, considering both labeled and unlabeled 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 modeling 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.

4 | CONCLUSIONS

Most of the document clustering algorithms have been analyzed in this review. 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 documents, taking into consideration the preprocessing 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.

AUTHOR CONTRIBUTIONS

Irene Cozzolino: Conceptualization (equal); methodology (equal); writing – original draft (lead). **Maria Brigida Ferraro:** Conceptualization (equal); methodology (equal); supervision (lead).

ACKNOWLEDGEMENT

Open Access Funding provided by Universita degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study

ORCID

Maria Brigida Ferraro https://orcid.org/0000-0002-7686-5938

RELATED WIREs ARTICLE

Soft Clustering

REFERENCES

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

Arthur, D., & Vassilvitskii, S. (2006). *k*-means++: The advantages of careful seeding (Technical Report). 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. *Lecture Notes on Software Engineering*, *2*(3), 262–267.

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). Springer.

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). MIT Press.

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), 26–33.

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., & Winker, P. (2021). A fuzzy clustering approach for textual data. In *Book of short papers SIS 2021* (pp. 770–776). Pearson.

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). IEEE.

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., 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). Association for Computing Machinery.

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). IEEE.

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). IEEE.

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.

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.

Guattery, S., & Miller, G. L. (1994). On the performance of spectral graph partitioning methods. (Technical Report). Department of Computer Science, Carnegie-Mellon University, Pittsburgh PA.

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 (NZCSRSC 2008), Christchurch, New Zealand* (Vol. 4, pp. 9–56). Association for Computing Machinery.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Comput Surv*, *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). Springer.

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). Springer.

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). Association for Computing Machinery.

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). Association for Computing Machinery.

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). Omnipress.

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.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning, PMLR* (pp. 1188–1196).

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

Liu, J., Cai, D., & He, X. (2010). Gaussian mixture model with local consistency. In *Proceedings of the AAAI conference on artificial intelligence*. Association for the Advancement of Artificial Intelligence.

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*, 419–444.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). University of California.

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.

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). Newswood Limited.

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). Bradford Books.

Nguyen, E. (2013). Text mining and network analysis of digital libraries in r. In Y. Zhao & Y. Cen (Eds.), *Data mining applications with r* (pp. 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.

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

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), 333–389.

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.

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.

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 (Technical Report)*. University of Minnesota.

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

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.

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.

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). Association for Computational Linguistics.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37.

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.

How to cite this article: Cozzolino, I., & Ferraro, M. B. (2022). Document clustering. *WIREs Computational Statistics*, e1588. https://doi.org/10.1002/wics.1588