Performance evaluation of density-based clustering methods

Ramiz M. Aliguliyev *

Institute of Information Technology of Azerbaijan National Academy of Sciences, Department of Artificial Intelligence and Computer Sciences, 9, F. Agayev Street, Baku AZ1141, Azerbaijan

Abstract

With the development of the World Wide Web, document clustering is receiving more and more attention as an important and fundamental technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering. A good document clustering approach can assist computers in organizing the document corpus automatically into a meaningful cluster hierarchy for efficient browsing and navigation, which is very valuable for complementing the deficiencies of traditional information retrieval technologies. In this paper, we study the performance of different density-based criterion functions, which can be classified as internal, external or hybrid, in the context of partitional clustering of document datasets. In our study, a weight was assigned to each document, which defined its relative position in the entire collection. To show the efficiency of the proposed approach, the weighted methods were compared to their unweighted variants. To verify the robustness of the proposed approach, experiments were conducted on datasets with a wide variety of numbers of clusters, documents and terms. To evaluate the criterion functions, we used the WebKb, Reuters-21578, 20Newsgroups-18828, WebACE and TREC-5 datasets, as they are currently the most widely used benchmarks in document clustering research. To evaluate the quality of a clustering solution, a wide spectrum of indices, three internal validity indices and seven external validity indices, were used. The internal validity indices were used for evaluating the within-cluster scatter and between cluster separations. The external validity indices were used for comparing the clustering solutions produced by the proposed criterion functions with the “ground truth” results. Experiments showed that our approach significantly improves clustering quality. In this paper, we developed a modified differential evolution (DE) algorithm to optimize the criterion functions. This modification accelerates the convergence of DE and, unlike the basic DE algorithm, guarantees that the received solution will be feasible.

1. Introduction

With the rapid development of the World Wide Web, we are facing an increasing volume of electronic documents, such as news articles and scientific papers. This explosion of electronic documents has made it difficult for users to extract useful information from them. Document clustering is receiving more and more attention as an important and fundamental technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering [30,34,49]. It is an effective tool to manage information overload. By clustering similar documents together, we can quickly browse document collections, easily grasp their distinct topics and subtopics, and efficiently query them, among many other applications.

* Tel.: +994 12 439 01 67; fax: +994 12 439 61 21.
E-mail addresses: a.ramiz@science.az, aramiz@iit.ab.az

© 2009 Elsevier Inc. All rights reserved.

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.06.012
Generally, clustering can be defined as partitioning of a given dataset into clusters, in such a way that data points belonging to the same cluster are as similar to each other as possible, whereas data points from two different clusters are separated by the maximum difference [18–20]. Document clustering has been investigated as a fundamental operation in many areas, such as data mining [29], information retrieval (IR) [10,27,33] topic detection [8], data streams [16], and as a preprocessing step for other tasks, such as text summarization [2,3,5,7,22].

Clustering can be performed in two different modes [18,23,24], hard and soft. In hard clustering the clusters are disjointed and non-overlapping in nature. Any pattern may belong to one and only one class in this case. In the case of soft clustering, a pattern may belong to any or all of the classes with various membership grades. In this paper, we deal with the hard clustering problem.

We introduce a number of weighted criterion functions, which can be classified into three groups: internal, external and hybrid. In particular, we evaluate a total of twelve criterion functions that measure various aspects of intra-cluster similarity, inter-cluster dissimilarity, and their combinations. We developed a modified DE algorithm to optimize the criterion functions. The proposed methods were experimentally evaluated on different datasets using various validity indices and metrics. For the evaluation, we used the WebKb, Reuters-21578, 20Newsgroups-18828, WebACE and TREC-5 datasets, as they are currently the most widely used benchmarks in document clustering research.

The rest of the paper is organized as follows. Section 2 describes a review of the existing clustering methods. Section 3 introduces the proposed clustering methods. Section 4 presents the modified DE to optimize the clustering methods. Section 5 describes the validity indices that were used for evaluation of the quality of the clustering results. We report experimental results in Section 6, and conclude the paper in Section 7.

2. Brief review of the existing clustering methods

Generally, clustering methods can be categorized into hierarchical methods and partitioning methods. Within each of the types, there exist a wealth of subtypes and different algorithms for constructing the clusters. An extensive survey of various clustering techniques can be found in the literature [25,29,32]. Here, we focus on reviewing the partitional clustering techniques, which are the most directly related to our work.

Partitional clustering algorithms decompose the dataset into a number of disjointed clusters that are usually optimal in terms of some predefined criterion functions. For instance, k-means is a typical partitioning method that aims to minimize the sum of the squared distance between the data points and their corresponding cluster centers [25,29,32]. Many criterion functions have been proposed in the literature [1,4,6,48,51] for producing balanced partitions. Their objective is to maximize the intra-cluster connectivity (compactness) while minimizing inter-cluster connectivity (separability). Zhao and Karypis [51] evaluated the performance of different criterion functions in the context of partitional clustering algorithms of document datasets. This study involved a total of seven different criterion functions. The methods proposed in [4,6] satisfy homogeneity within-clusters as well as separability between the clusters. A novel mixed-integer nonlinear programming-based clustering algorithm, the global optimal search with enhanced positioning, is presented in [48]. This algorithm is significant in that it is able to progressively identify and weed out outlier data points.

The k-means method [25,29,32] is a commonly used partitioning algorithm in document clustering and other related research areas. This method is based on the idea of a center point that can represent a cluster. Given n objects, the method first selects k objects as the initial k clusters. Then it iteratively assigns each object to the most similar cluster based on the mean value of the objects in each cluster. There are many variations of the k-means method [11,31,36]. Bagirov [11] proposed a new version of the global k-means algorithm, a modified global k-means algorithm that computes clusters incrementally, and computes the k-partition of a data set by using k – 1 cluster centers from the previous iteration. An important step in this algorithm is the computation of a starting point for the kth cluster center. This starting point is computed by minimizing the so-called auxiliary cluster function. A new k-means type algorithm called W-k-means [31] automatically weights the variables based on their importance in clustering. W-k-means adds a new step to the basic k-means algorithm to update the variable weights on the current partition of data. Based on the current partition in the iterative k-means clustering process, the algorithm calculates a new weight for each variable based on the variance of the within-cluster distances. The new weights are used to decide the cluster memberships of the objects in the next iteration. The weights can be used to identify important variables for clustering and the variables that are likely to contribute noise to the clustering process and can be removed from the data in the future analysis.

Two new text document clustering algorithms, called clustering based on frequent word sequences (CFWS) and clustering based on frequent word meaning sequences (CFWMS), are proposed in [38]. Unlike the traditional VSM, these models utilize the sequential patterns of the words in the document. Frequent word sequences discovered from the document set can represent the topics covered by the documents very well, and the documents containing the same frequent word sequences are clustered together in these algorithms. Li et al. [40] proposed a new text-clustering algorithm, called text clustering with feature selection (TCFS), that performs a supervised feature selection during the clustering process. The selected features improve the quality of clustering iteratively, and as the clustering process converges, the clustering result has higher accuracy.

The problem of partitional clustering has been approached from diverse fields of knowledge such as graph theory [15,21,37,49], neural networks [31], genetic algorithms (GA) [4,6,12,36,42] particle swarm optimization (PSO) [19], ant algorithm [9], and differential evolution [1,18,19]. In the evolutionary approach, clustering a dataset is viewed as an optimization
problem and solved using an evolutionary search heuristic such as GA [4,12,36,42], DE [1,18,19] or PSO [20]. Das et al. [20] present a novel, modified PSO-based strategy for the hard clustering of complex data. A new DE-based strategy for hard clustering of real-world data sets was presented in [19]. An important feature of the proposed techniques [18–20] is that they are able to find the optimal number of clusters automatically (that is, the number of clusters does not have to be known in advance) for complex and linearly non-separable datasets. A clustering algorithm called clustering with local and global regularization (CLGR) [49] preserves the merit of local-learning algorithms and spectral clustering. Spectral clustering formulates clustering as a graph partitioning problem. The optimal partition is approximated by eigenvectors of a properly normalized affinity matrix of the graph. The relationships between spectral partitioning methods and kernel k-means are discussed in [21]. This study showed that a weighted form of the kernel k-means objective is mathematically equivalent to a general, weighted graph partitioning objective. The key contribution of the graph-based relaxed (GBR) algorithm [37] is its very simple implementation using the existing optimization packages. In [9], a new model (called AntTree) was presented for data clustering, which was inspired by the self-assembly behavior of real ants. In [4], a fast GA was proposed for document clustering, in which the penalty function was introduced to accelerate the convergence. In [12], a new symmetry based genetic clustering algorithm, VGAPS, was proposed, which automatically evolves the number of clusters as well as the proper partitioning from a data set. The algorithm genetic algorithm k-means logarithmic regression expectation maximization (GAKREM) [42] combines the best characteristics of the k-means and EM algorithms but avoids their weaknesses. The novelty of GAKREM is that in each evolving generation it efficiently approximates the log-likelihood for each chromosome using logarithmic regression, instead of running the conventional EM algorithm until it converges.

3. Clustering documents

The standard clustering technique consists of the following steps: (1) feature selection and data representation model, (2) similarity measure selection, (3) clustering model, (4) clustering algorithm that generates the clusters using the data model and the similarity measure, (5) validation [27].

3.1. Document representation and term weighting scheme

Let \( \mathbf{D} = (D_1, D_2, \ldots, D_n) \) be a collection of documents and \( \mathbf{T} = (T_1, T_2, \ldots, T_m) \) be the complete vocabulary set of the document collection \( \mathbf{D} \), where \( n \) is the number of documents and \( m \) is the number of unique terms. There are several ways to model a text document. We apply the vector space model, widely used in IR and text mining [10], to represent the text documents. In this model each document \( D_i \) is represented by a point in an \( m \) dimensional vector space, \( D_i = (w_{i1}, w_{i2}, \ldots, w_{im}) \), \( i = 1, \ldots, n \), where the dimension is the same as the number of terms in the document collection. Each component of such a vector reflects a term connected with the given document. The value of each component depends on the degree of relationship between its associated term and the respective document. Many schemes have been proposed for measuring this relationship. Term weighting is the process of calculating the degree of relationship (or association) between a term and a document. One of the more advanced term weighting schemes is the tf-idf (term frequency-inverse document frequency) [10]. The tf-idf scheme aims at balancing the local and the global term occurrences in the documents. In this scheme

\[
w_{ij} = n_i \times \log \left( \frac{n}{n_{ij}} \right),
\]

where \( n_i \) is the term frequency, and \( n_{ij} \) denotes the number of documents in which term \( T_j \) appears. The term \( \log(n/n_i) \), which is often called the idf factor, defines the global weight of the term \( T_j \). Indeed, when a term appears in all documents in the collection, then \( n_{ij} = n \), and thus the balanced term weight is 0, indicating that the term is useless as a document discriminator. The idf factor has been introduced to improve the discriminating power of terms in the traditional information retrieval.

3.2. The cosine measure

As mentioned above, clustering is the process of recognizing natural groups or clusters in multi-dimensional data based on some similarity measures. Hence, defining an appropriate similarity measure plays a fundamental role in clustering [30,35]. A variety of similarity or distance measures have been proposed and widely applied, such as cosine similarity, Euclidean distance and the Jaccard correlation coefficient [30,35].

The cosine measure is one of the most popular similarity measures applied to text documents. This measure computes the cosine of the angle between two feature vectors and is used frequently in text mining. The cosine similarity between two documents \( D_i \) and \( D_l \) is calculated as

\[
sim(D_i, D_l) = \cos(D_i, D_l) = \frac{\sum_{j=1}^{m} w_{ij} w_{lj}}{\sqrt{\sum_{j=1}^{m} w_{ij}^2 \cdot \sum_{j=1}^{m} w_{lj}^2}}, \quad i, l = 1, \ldots, n.
\]
3.3. The proposed clustering methods

The hard clustering problem can be defined as follows [25,29,32]. A clustering \( C \) is a partition of a dataset \( D \) into mutually disjointed subsets \( C_1, C_2, \ldots, C_k \), called clusters, such that \( C_p \cap C_q = \emptyset, p \neq q \) (i.e., two different clusters should have no documents in common) and \( \bigcup_{p=1}^{k} C_p = D \) (i.e., each document should definitely belong to a cluster). We also assume that for all \( q = 1, \ldots, k \) \( C_q \neq \emptyset \) and \( C_q \in D \), i.e., each cluster should have at least one document assigned and it must not contain all documents. In other words, \( k \) represents the number of non-empty clusters.

As is well known, the clustering quality depends on the clustering technique and the data structure. Different methods may show different qualities of compactness and separability of clusters; some methods can provide a higher level of compactness and a lower level of separability, and others vice versa, and some may balance compactness and separability of clusters.

The main drawback most of the clustering methods is that they do not consider data structure. They assume that the data are independent and are identically distributed samples generated from an unknown probability density function. Therefore, they cannot find the clusters of an arbitrary shape. In this paper, there is an attempt to find the clusters of arbitrary shape. To find the clusters of arbitrary shape, it is necessary to have some prior knowledge about the distribution of points. In this paper, in order to have a priori knowledge about the distribution of points, we define their relative positions in a dataset. The relative position of a point is defined by a measure of affinity between the point and the center of a dataset. It defines a concentration degree of points around the center of the entire collection. In this paper, to verify the robustness of the suggested approach, experiments were conducted on the datasets with varying numbers of clusters, documents and terms. To evaluate the quality of a clustering solution, a wide spectrum of indices – three internal validity indices and seven external validity indices – were used. Internal indices evaluate the compactness and separation of the clusters, and external validity indices compare a clustering solution to a true clustering. Experiments showed that our approach significantly improves the clustering quality.

In general, our study involves the twelve criterion functions that are given below. These criterion functions have been proposed in the context of partition clustering algorithms. Our goal is to find such a clustering solution that will satisfy not only the homogeneity and separability of clusters, but also the separability of clusters from all collections. For this purpose, in our clustering methods a weight is assigned to each document defining its position in a document collection.

Thus, let each document \( D_i \) be associated with a positive weight \( a_i \), which is defined as follows:

\[
\alpha_i = \frac{\text{sim}(D_i, O)}{\sum_{i=1}^{n} \text{sim}(D_i, O)}, \quad i = 1, \ldots, n, \tag{3}
\]

where \( O \) is the center of the document collection \( D \), and \( j \)th coordinate \( o_j \) of the center \( O \) is calculated as: 

\[
o_j = \frac{1}{k} \sum_{i=1}^{n} w_i (j = 1, \ldots, m). \tag{4}
\]

This weight defines the degree of relative similarity of the document \( D_i \) to the centre of the \( D \), i.e., this weight defines the position of the document \( D_i \) relative to the center of the entire collection \( D \). It is not difficult to see that

\[
\sum_{i=1}^{n} \alpha_i = 1. \tag{5}
\]

For each cluster \( C_p \), we define the weighted cardinality \(|C_p|_\alpha\) to be

\[
|C_p|_\alpha = \sum_{D_i \in C_p} \alpha_i. \tag{5}
\]

If the weights \( \alpha \) are equal to 1, then (5) is identical to the standard definition of set cardinality, i.e., the size of the cluster \( C_p \).

From the formula (4), it follows that \( \sum_{p=1}^{k} |C_p|_\alpha = 1 \).

The weighted criterion functions introduced in this section can be classified into three groups: internal, external and hybrid.

The internal criterion functions focus on producing a clustering solution that optimizes a function defined only over the documents of each cluster, without taking into account the documents assigned to different clusters:

\[
\mathcal{F}_1 = \sum_{p=1}^{k} \sum_{D_i \in C_p} \frac{\text{sim}(D_i, D_j)}{\alpha_i \alpha_j} \rightarrow \max, \tag{6}
\]

\[
\mathcal{F}_2 = \sum_{p=1}^{k} \sum_{D_i \in C_p} \frac{\text{sim}(D_i, O_p)}{\alpha_i} \rightarrow \max. \tag{7}
\]

where \( O_p \) denotes the centroid of the cluster \( C_p, O_p = \frac{1}{|C_p|} \sum_{D_i \in C_p} D_i \), and \(|C_p|\) is the number of documents belonging to cluster \( C_p \).

The \( \mathcal{F}_1 \) criterion function (6) maximizes the sum of the pairwise similarities between the documents assigned to each cluster, taking into account their positions in a dataset defined by (3). The \( \mathcal{F}_2 \) criterion function (7) is a weighted version of the k-means algorithm. Comparing \( \mathcal{F}_2 \) and the k-means algorithm, we see that the essential difference between them
is that in this algorithm, the contribution of each cluster is weighted proportionally to its weighted cardinality and the contribution of each document is weighted inversely to its weight. In this algorithm, each cluster is represented by its centroid vector, and the goal is to find the solution that maximizes the similarity between each document and the centroid of the cluster to which it is assigned, taking into account the position of the document in a dataset. The functions \( F_2 \) and \( F_2^* \) not only provide compactness of the clusters but also separation of the clusters from the entire collection.

The external criterion functions derive the clustering solution by focusing on optimizing a function that is based on how the various clusters are different from the entire collection and from each other:

\[
F_2^* = \sum_{p=1}^{k} |C_p| \sum_{D_i \in C_p} \text{sim}(D_i, O_p) \rightarrow \min, \\
F_2^* = \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} |C_p| |C_q| \text{sim}(O_p, O_q) \rightarrow \min. 
\]

The \( F_2^* \) criterion function (8) computes the clustering by finding a solution that separates the documents of each cluster from the entire collection. Specifically, the \( F_2^* \) criterion function tries to minimize the similarity between the centroid vector of each cluster and the centroid vector of the entire collection. The contribution of each cluster is weighted proportionally to its weighted cardinality, so that larger clusters will be weighted higher in the overall clustering solution. The \( F_2^* \) criterion function (9) computes the clustering by finding a solution that separates each cluster from other clusters.

The following hybrid criterion functions simultaneously optimize multiple individual criterion functions:

\[
F_5 = \frac{F_3 + F_4}{F_1} \rightarrow \min, \\
F_6 = \frac{F_3 + F_4}{F_2} \rightarrow \min. 
\]

The \( F_5, F_6 \) criterion functions (10), (11) are obtained by combining the criterion functions (6)–(9). Since \( F_2^* \) and \( F_2^* \) are maximized, the \( F_2^* \) and \( F_2^* \) criterion functions need to be minimized as they are inversely related to \( F_2^* \) and \( F_2^* \), respectively. The \( F_5, F_6 \) criterion functions (10), (11) measure the quality of the overall clustering solution by taking into account the separation between clusters and the entire collection, the separation between clusters, and the tightness of each cluster.

To show the efficiency of the assignment of weights to documents, we shall compare the criterion functions (6)–(11) with the following criterion functions:

\[
F_1 = \sum_{p=1}^{k} |C_p| \sum_{D_i \in C_p} \text{sim}(D_i, O_p) \rightarrow \max, \\
F_2 = \sum_{p=1}^{k} |C_p| \sum_{D_i \in C_p} \text{sim}(D_i, O_p) \rightarrow \max, \\
F_3 = \sum_{p=1}^{k} |C_p| \text{sim}(O_p, O) \rightarrow \min, \\
F_4 = \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} |C_p| |C_q| \text{sim}(O_p, O_q) \rightarrow \min. \\
F_5 = \frac{F_3 + F_4}{F_1} \rightarrow \min, \\
F_6 = \frac{F_3 + F_4}{F_2} \rightarrow \min. 
\]

We shall call the criterion functions (12)–(17) unweighted versions of the criterion functions (6)–(11). The criterion functions (12)–(17) can be received from the (6)–(11) under the assumption \( x_i = x > 0 \) for any \( i \in \{1, \ldots, n\} \). Note that \( F_3 \) is the criterion function \( \mathcal{E}_3 \) proposed in [51]. Comparing \( F_2 \) and the \( k \)-means algorithm, we see that the essential difference between them is that in method \( F_2 \) the contribution of each cluster is weighted proportionally to its cardinality. We shall also call this function the weighted version of the \( k \)-means method.

4. DE-based clustering algorithm

In our study, the criterion functions are optimized using the DE [45]. In clustering research, it is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than finding an optimal partition. The evolutionary algorithms differ mainly in their representations of parameters (usually binary strings are used for genetic algorithms while parameters are real-valued for evolution strategies and DE) and in their evolutionary operators.
4.1. Chromosome representation

DE, like other evolutionary algorithms, begins with a randomly initialized population of multi-dimensional real-coded chromosomes. To represent the $a$th chromosome of the population at the current generation (at time $t$), we use the following notation:

$$X_a(t) = [x_{a,1}(t), x_{a,2}(t), \ldots, x_{a,m_k}(t)].$$

where $m_k = m \cdot k$, $a = 1, \ldots, N$, $N$ is the size of the population.

Each chromosome forms a candidate solution to the multi-dimensional optimization problem. It is a sequence of real numbers representing the $k$ cluster centers. For an $m$-dimensional space, the length of a chromosome is $m \cdot k$, where the first $m$ positions (or genes) represent the $m$ dimensions of the first cluster center, the next $m$ genes represent the second cluster center, and so on. For example, let $m = 2$ and $k = 4$, i.e., the space is two-dimensional and the number of clusters is four. Then the chromosome $X = [0.16, 0.37, 0.23, 0.75, 0.82, 0.26, 0.94, 0.68]$ represents the four cluster centers $(0.16, 0.37), (0.23, 0.75), (0.82, 0.26)$ and $(0.94, 0.68)$.

4.2. Population initialization

At the initial stage each chromosome randomly chooses $k$ different document vectors from the document collection $D = [D_1, D_2, \ldots, D_n]$ as the initial cluster centroid vectors. This process is repeated for each of the $N$ chromosomes in the population.

4.3. Fitness functions

We define the fitness functions according to the objective functions (6)–(11) as follows:

$$f_1^a(X) = \frac{1}{F_1^a(X)},$$
$$f_2^a(X) = \frac{1}{F_2^a(X)},$$
$$f_3^a(X) = F_3^a(X),$$
$$f_4^a(X) = F_4^a(X),$$
$$f_5^a(X) = F_5^a(X),$$
$$f_6^a(X) = F_6^a(X),$$

so that their minimization leads to the maximization (or minimization) of the criterion functions (6)–(11), respectively. We define the fitness functions corresponding to the criterion functions (12)–(17) in a similar way.

4.4. The basic DE algorithm

DE is based on a mutation operator, which adds an amount obtained by the difference of two randomly chosen individuals of the current population, in contrast to most of the evolutionary algorithms, in which the mutation operator is defined by a probability function. The basic algorithm of DE [45] is shown in Fig. 1.

![Pseudo-code of the basic DE algorithm.](image-url)
The scaling factor $\lambda \in [0, 1]$, and the crossover probability $pr \in [0, 1]$, are the control parameters of DE, which are set by the user. The values $rnd$, are uniformly distributed random numbers within the range $[0,1]$, chosen once for each $s \in \{1, \ldots, m_k\}$, $x_{rs}(t)$ is the $s$ th decision variable of the $r$ th chromosome in the population. $\mathcal{F}(\cdot)$ is the objective function to be minimized.

4.5. Crossover operator

The crossover operator for the chromosome of the current best solution $X_b(t)$ randomly chooses two other chromosomes $X_a(t)$ and $X_c(t)$ from the same generation. Then it calculates the weighted difference $\pi(X_b(t) - X_a(t)) + (1 - \pi)(X_b(t) - X_c(t))$ and creates a trial offspring chromosome by adding the result to the chromosome $X_b(t)$ scaled by the factor $\lambda_c$. Thus, for the $s$ th gene $(s = 1, 2, \ldots, m_k)$ of $X_b(t + 1)$, we have

$$y_{bs}(t + 1) = \begin{cases} \lambda_c x_{bs}(t) + \pi_s(x_{bs}(t) - x_{as}(t)) + (1 - \pi_s)(x_{bs}(t) - x_{cs}(t)), & \text{if } \theta_s < pr, \\ x_{bs}(t), & \text{otherwise}. \end{cases}$$

(25)

The scaling factor $\lambda_c \in [0.5, 1.0]$ and the crossover constant, $pr_c \in [0, 1]$, are control parameters, which are set by the user. The values $\pi_s$ and $\theta_s$ are uniformly distributed random numbers within the range $[0,1]$, chosen once for each $s \in \{1, \ldots, m_k\}$.

We define the function $MaxMin(x_i(t))$ in (25) as

$$MaxMin(x_i(t)) = \begin{cases} x_i^{\text{min}}(t) + \delta_i(t), & \text{if } x_i(t) \leq x_i^{\text{min}}(t), \\ x_i(t), & \text{if } x_i^{\text{min}}(t) < x_i(t) < x_i^{\text{max}}(t), \\ x_i^{\text{max}}(t) - \delta_i(t), & \text{if } x_i(t) \geq x_i^{\text{max}}(t) \end{cases}$$

(26)

where $x_i^{\text{min}}(t) = \min_{3b_1(12\ldots N)} \{x_{bs}(t)\}$, $x_i^{\text{max}} = \max_{3b_1(12\ldots N)} \{x_{bs}(t)\}$, and $\delta_i(t) = \frac{x_i^{\text{max}}(t) - x_i^{\text{min}}(t)}{n}$. DE uses the principle of “survival of the fittest” in its selection process, which may be expressed as:

$$X_b(t + 1) = \begin{cases} Y_{bs}(t + 1), & \text{if } f_z(Y_{bs}(t + 1)) < f_z(X_b(t)), \\ X_b(t), & \text{otherwise} \end{cases}$$

(27)

where the fitness functions $f_z(x), z = 1, 2, \ldots, 6$ are defined by (19)–(24).

4.6. Mutation operator

The mutation operation for the target chromosome $X_b(t)$ is performed according to the following rule:

$$y_{bs}(t + 1) = \begin{cases} MaxMin(\lambda_m x_{bs}(t) + \rho_s(x_{bs}(t) - x_{bs}(t))), & \text{if } \eta_s < pr_m, \\ x_{bs}(t), & \text{else} \end{cases}$$

(28)

with distinct random integer indices $s, q, r \in \{1, \ldots, m_k\}, s \neq q \neq r$. The control parameters – the scaling factor $\lambda_m \in [0.5, 1.0]$ and the mutation constant, $pr_m \in [0, 1]$ – are set by the user. The values $\rho_s$ and $\eta_s$ are uniformly distributed random numbers within the range $[0,1]$, chosen once for each $s \in \{1, \ldots, m_k\}$. If the mutant chromosome yields a better value of the fitness function, it replaces its parent in the next generation; otherwise the parent remains in the population.

We explain the objective of using the $MaxMin(x_i(t))$ function in the crossover (25) and mutation (28) operations. Given the nature of the problem, after applying the algorithm we must have a solution such that the coordinates belong to the interval $(x_i^{\text{min}}, x_i^{\text{max}})$, $s \in \{1, \ldots, m_k\}$ (we call such a solution feasible). As can be seen from Fig. 1, a direct application of the basic DE algorithm does not guarantee that the obtained solution will be feasible. However such a solution can also be obtained from the basic DE algorithm. In this case, the feasibility of solution must be verified at every step. Obviously, it requires additional computational efforts and as a result the convergence of the algorithm is delayed. Therefore, we introduce the function $MaxMin(x_i(t))$ that ensures the feasibility of the solution and accelerates the convergence of the algorithm.

4.7. Halting criterion

The algorithm terminates when the maximum number of fitness calculations $t_{\text{max}}$ is achieved.

4.8. The pseudo-code of the proposed DE algorithm

The pseudo-code of the clustering algorithm is given below:

**Step 1** *(Input. Create initial population)*. At the initial stage each chromosome randomly chooses $k$ different document vectors from the document collection $D = [D_1, D_2, \ldots, D_n]$ as the initial cluster centroid vectors.

**Step 2** *(Form initial clusters)*. Assign each document vector $D_i$ ($i = 1, 2, \ldots, n$) in the document collection to the cluster $C_p$, $p \in \{1, 2, \ldots, k\}$ if $\text{sim}(D_i, O_{p}) > \text{sim}(D_i, O_{q})$, $q \in \{1, 2, \ldots, k\}$, and $p \neq q$.

**Step 3** *(Evaluate initial population)*. Calculate the fitness value of each chromosome in the population based on (19)–(24).

**Step 4** *(Select best chromosome)*. Select the chromosome with current best solution.
5. Measures for evaluation of the clustering quality

Cluster validation refers to the quantitative evaluation of the quality of a clustering solution. In general, there are three approaches to investigate cluster quality. The first approach is based on internal criteria. Internal criteria assess the clusters against their own structural properties. Internal cluster validation aims at measuring the quality of a clustering in real-life settings when there is no knowledge of the real clustering. The second approach is based on external criteria. External cluster validation refers to comparing a clustering solution to a true clustering. This is important in evaluating the performance of a clustering algorithm on datasets. The third approach is based on relative criteria for the investigation of cluster quality. Here the basic idea is the evaluation of a clustering structure by comparing it to other clustering schemes produced by the same method but with different input parameter values. Various cluster validity indices are available in the literature [13,23,26,41,43,46,50]. In this paper, we use indices for the internal and external approaches.

5.1. Measures for internal cluster validation

In this subsection, we discuss suitable methods for the quantitative evaluation of a clustering result, known as internal cluster validity indices. Ideally, a validity index should measure the two aspects of partitioning:

1. **Cohesion**: The documents in one cluster should be as similar to each other as possible. The fitness variance of the documents in a cluster is an indication of the cluster’s cohesion or compactness.
2. **Separation**: Clusters should be well separated. The similarity among the cluster centers gives an indication of cluster separation.

Many internal validity measures have been proposed for evaluating clustering results. Most of these popular validity measures do not work well for clusters with different densities and/or sizes. They usually have a tendency to ignore clusters with low densities. A validity measure that can deal with this situation is studied in [1,17–19]. This measure is the ratio of the sum of intra-cluster scatter to inter-cluster separation:

\[
CS_1(k) = \frac{\sum_{p=1}^{k} \left\{ \frac{1}{|C_p|} \sum_{D_i \in C_p} \min\{\text{sim}(D_i, D_j)\} \right\}}{\sum_{p=1}^{k} \max_{q=1...k, q \neq p} \{\text{sim}(O_p, O_q)\}}.
\] (29)

This cluster validity index is inspired by the work reported in [17], and has been suitably modified for clustering different datasets [1,18,19]. It simultaneously takes the cohesion and separation factors into account while dealing with complex structure data sets. The denominator in (29) computes the largest similarity between cluster centers. The numerator measures the average smallest similarity between two documents lying in the same cluster, whereas the latter uses the smallest similarity between two documents lying in the same cluster to measure the scatter volume.

Similarly, the second measure is defined as

\[
CS_2(k) = \frac{\sum_{p=1}^{k} \left\{ \frac{1}{|C_p|} \min\{\text{sim}(D_i, O_p)\} \right\}}{\sum_{p=1}^{k} \max_{q=1...k, q \neq p} \{\text{sim}(O_p, O_q)\}}.
\] (30)

We define the third measure as

\[
CS_3(k) = \frac{\sum_{p=1}^{k} \frac{1}{|C_p|} \sum_{D_i \in C_p} \text{sim}(D_i, O_p)}{\sum_{p=1}^{k} \text{sim}(O_p, O)}.
\] (31)

The numerator in (31) measures the average similarity of documents to the cluster centers. The denominator in (31) computes the sum of the similarity between cluster centers and the center of the entire collection.
The validity indices (29)–(31) simultaneously take the compactness and separation factors into account while dealing with complex structure datasets. A large value of these measures (29)–(31) indicates a valid optimal partition.

5.2. Measures for external cluster validation

The second set of experiments will be focused on comparing of clustering results produced by the proposed criterion functions with the “ground truth” results. The quality of the clustering solution will be measured by using different metrics. These metrics measure the matching of clusters computed by each method to the “ground truth” classes. In situations where documents are already labeled, we can compare the clusters with the “true” class labels.

Assume that the dataset \( D \) is composed of the classes \( C^+ = (C_1^+, \ldots, C_k^+) \) (true clustering), and we apply a clustering procedure to find clusters \( C = (C_1, \ldots, C_k) \) in this dataset. We present various indices to compare the two partitions \( C = (C_1, \ldots, C_k) \) and \( C^* = (C_1^*, \ldots, C_k^*) \).

Important classes of criteria for comparing clustering solutions are based on counting the pairs of points on which two clustering agree/disagree. The best-known clustering distances based on point pairs are the purity, the precision, the recall, the clustering accuracy, and the F-measure.

Purity. The purity of the cluster \( C_p \) is defined as follows [14,47]:

\[
\text{purity}(C_p) = \frac{1}{|C_p|} \max_{p' \neq p} |C_p \cap C_{p'}^+|, \quad p = 1, \ldots, k.
\]

The purity gives the ratio of the dominant class size in the cluster to the cluster size itself. The value of the purity is always in the interval \( [0,1] \). A large purity value implies that the cluster is a “pure” subset of the dominant class. The purity of the entire collection of clusters was evaluated as a weighted sum of the individual cluster purities:

\[
\text{purity}(C) = \frac{1}{n} \sum_{p=1}^{k} \text{purity}(C_p) = \frac{1}{n} \sum_{p=1}^{k} \max_{p' \neq p} |C_p \cap C_{p'}^+|.
\]

According to this measure, a higher purity value indicates a better clustering solution.

Mirkin metric. The Mirkin metric is defined as follows [41]:

\[
M(C, C^+) = \frac{1}{n^2} \left( \sum_{p=1}^{k} |C_p|^2 + \sum_{p'=1}^{k} |C_{p'}^+|^2 - 2 \sum_{p=1}^{k} \sum_{p'=1}^{k} |C_p \cap C_{p'}^+|^2 \right).
\]

The Mirkin metric (34) is scaled with the factor \( 1/n^2 \) in order to restrict its range to the interval [0,1]. This metric is obviously 0 for identical clusterings, and positive otherwise.

F-measure. Another frequently used external validation measure is commonly called the “clustering accuracy”. The calculation of this accuracy is inspired by the information retrieval metric known as the F-measure. If we want to compare a clusters \( C \) to a set of classes \( C^+ \), a simple approach would be to calculate the precision (\( P \)), recall (\( R \)) and the F-measure, used widely in the information retrieval literature, to measure the success of the retrieval task.

Using our clustering notation, the precision is computed as follows [46]:

\[
P(C_p, C_{p'}) = \frac{|C_p \cap C_{p'}^+|}{|C_p|}.
\]

The precision is calculated as the portion of cluster \( C_p \) that includes the documents of class \( C_{p'}^+ \), thus measuring how homogenous the cluster \( C_p \) is with respect to the class \( C_{p'} \).

Similarly, the recall is calculated as the proportion of documents from class \( C_{p'}^+ \) that are included in cluster \( C_p \), thus measuring how complete the cluster \( C_p \) is with respect to the class \( C_{p'}^+ \):

\[
R(C_p, C_{p'}) = \frac{|C_p \cap C_{p'}^+|}{|C_{p'}^+|}.
\]

Then F value of the cluster \( C_p \) and the class \( C_{p'}^+ \) is the harmonic mean of the precision and the recall:

\[
F(C_p, C_{p'}) = \frac{2}{\frac{1}{P(C_p, C_{p'})} + \frac{1}{R(C_p, C_{p'})}} = \frac{2P(C_p, C_{p'})R(C_p, C_{p'})}{P(C_p, C_{p'}) + R(C_p, C_{p'})}.
\]
\[ F(C_p) = \max_{C_p' \in C} F\left(C_p, C_p'\right), \quad p = 1, 2, \ldots, k. \]  

(38)

The \textit{F-measure} of the entire collection is considered to be the sum of the individual cluster specific \textit{F-measures}, weighted according to cluster size. That is,

\[ F(C) = \sum_{p=1}^{k} \frac{|C_p|}{n} F(C_p). \]  

(39)

The higher the \textit{F-measure}, the better the clustering solution. This measure has a significant advantage over the \textit{purity} and the \textit{entropy}, because it measures both the homogeneity and the completeness of a clustering solution [46].

Now we describe information-based methods to compare partitions \( C = (C_1, \ldots, C_k) \) and \( C^+ = (C_1^+, \ldots, C_k^+) \). The commonly used external validity indices based on information are the \textit{partition coefficient} [13], the \textit{entropy} [13,14], the \textit{variation of information} [43] and the \textit{V-measure} [46]. These measures represent plausible ways to evaluate the homogeneity of a clustering solution.

\textbf{Partition coefficient.} The partition coefficient (PC) was introduced by Bezdek [13]. It measures the amount of overlap between clusters. Considering a cluster \( C_p \), the PC is defined as follows:

\[ \text{PC}(C_p) = \frac{1}{k'} \sum_{p'=1}^{k'} \left( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \right)^2. \]  

(40)

\( \text{PC}(C_p) \) is a value between \( \frac{1}{k} \) and 1. If almost all documents of \( C_p \) belong to the same cluster in \( C_p^+ \), then \( \text{PC}(C_p) \) is close to 1. On the other hand, if the documents of \( C_p \) are randomly divided into all clusters of \( C^+ \) then \( \text{PC}(C_p) \) is close to \( \frac{1}{k} \).

A global partition coefficient is computed using the following formula:

\[ \text{PC}(C, C^+) = \frac{1}{k} \sum_{p=1}^{k} \text{PC}(C_p) = \frac{1}{kk'} \sum_{p=1}^{k} \sum_{p'=1}^{k'} \left( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \right)^2. \]  

(41)

\( \text{PC}(C, C^+) \) also takes values between \( \frac{1}{k} \) and 1. Now, if \( \text{PC}(C, C^+) \) is close to \( \frac{1}{k} \), then \( C \) and \( C^+ \) are almost independent. Moreover, if \( \text{PC}(C, C^+) \) is close to 1, then \( C \) is close to \( C^+ \).

\textbf{Entropy.} An entropy measure based on information-theoretic considerations can be also used. In the same way as a \textit{partition coefficient}, Bezdek defined the clustering entropy [13]. The entropy of the cluster \( C_p \) is defined to be

\[ E(C_p) = -\frac{1}{\log(k')} \sum_{p'=1}^{k'} \left( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \right) \log \left( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \right), \quad p = 1, 2, \ldots, k. \]  

(42)

Note that when \( x \) is close to 0, then \( x \log x \) is close to 0. So we consider that \( 0 \log 0 = 0 \).

Since the entropy considers the distribution of semantic classes in a cluster, it is a more comprehensive measure than the purity. Note that we have normalized the entropy to take values between 0 and 1. If almost all the documents of cluster \( C_p \) belong to the same class \( C_p^+ \), then \( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \) is close to 1 for \( p = p^+ \) and is close to 0 for \( p \neq p^+ \). So the entropy \( E(C_p) \) is close to 0 (since \( 0 \log 0 = 0 \) and \( 1 \log 1 = 0 \)). On the other hand, if the documents of cluster \( C_p \) are randomly divided among all the classes of \( C^+ \), then \( \frac{|C_p \cap C_{p'}^+|}{|C_p|} \) is close to \( \frac{1}{k'} \) and the entropy of cluster \( C_p \) is close to 1.

In contrast to the purity measure, an entropy value of 0 means that the cluster is comprised entirely of one class, while an entropy value near 1 implies that the cluster contains a uniform mixture of all classes.

The global clustering entropy of the entire collection is defined to be the sum of the individual cluster entropies weighted according to the cluster size. That is,

\[ E(C) = \sum_{p=1}^{k} \frac{|C_p|}{n} E(C_p). \]  

(43)

The global entropy also takes values between 0 and 1. A perfect clustering solution will be one that produces clusters that contain documents from only a single class, in which case entropy will be zero. In general, the smaller the entropy, the better the quality of the cluster.

\textbf{Variation of information.} Another information-based clustering measure is the \textit{variation of information (VI)} [43]. The \textit{variation of information} is a recently proposed clustering criterion based on information-theoretic concepts. It measures the amount of information that we gain and lose when going from the clustering \( C \) to another clustering \( C^+ \). Patrikainen and Meila [43] define it as
This property is undesirable then one can simply normalize by log where $H(C | C^+)$ is the conditional entropy of $C$ given $C^+$.

An equivalent way of writing the distance $\text{VI}(C, C^+)$ is as follows [43]:

$$\text{VI}(C, C^+) = -\sum_{p=1}^{k} \sum_{p'=-1}^{k'} P(C_p, C_p^+) \log \left( \frac{P(C_p, C_p^+)}{P(C_p^+)} \right) - \sum_{p=1}^{k} \sum_{p'=-1}^{k'} P(C_p, C_p^+) \log \left( \frac{P(C_p, C_p^+)}{P(C_p)} \right).$$  \hspace{1cm} (45)

The variation of information is the sum of the information needed to describe $C$ given $C^+$ and the information needed to describe $C^+$ given $C$.

The joint distribution $P(C_p, C_p^+)$ in (45) is equal to

$$P(C_p, C_p^+) = \frac{|C_p \cap C_p^+|}{n}. \hspace{1cm} (46)$$

This immediately also implies that

$$P(C_p) = \frac{|C_p|}{n}. \hspace{1cm} (47)$$

and

$$P(C_p^+) = \frac{|C_p^+|}{n}. \hspace{1cm} (48)$$

Based upon these calculations (46)–(48), we define the variation of information (45) as

$$\text{VI}(C, C^+) = \frac{1}{n} \sum_{p=1}^{k} \sum_{p'=-1}^{k'} \left| C_p \cap C_p^+ \right| \log \left( \frac{|C_p \cap C_p^+|}{|C_p \cap C_p^+|^2} \right). \hspace{1cm} (49)$$

The maximum value of the variation of information is $\log n$, which is achieved when the partitions are as far apart as possible. In this case it means that one of them places all the documents together in a single cluster while the other places each document in a cluster on its own. The maximum value increases with $n$ because larger datasets contain more information, but if this property is undesirable then one can simply normalize by $\log n$, as we do in the calculations presented here:

$$\text{VI}(C, C^+) = \frac{1}{n \log n} \sum_{p=1}^{k} \sum_{p'=-1}^{k'} \left| C_p \cap C_p^+ \right| \log \left( \frac{|C_p \cap C_p^+|}{|C_p \cap C_p^+|^2} \right). \hspace{1cm} (50)$$

In general, the smaller the variation of information, the better the clustering solution.

The variation of information is presented as a distance measure for comparing clusterings of the same dataset. Therefore it does not distinguish between hypothesized and target clusterings.

**V-measure.** The V-measure is an entropy-based measure that explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied [46]. We define the homogeneity as

$$\text{hom}(C) = \begin{cases} 1, & \text{if } H(C | C^+) = 0 \\ 1 - \frac{H(C | C) H(C^+ | C)}{H(C^+ | C)} & \text{else} \end{cases}. \hspace{1cm} (51)$$

where

$$H(C | C^+) = -\sum_{p=1}^{k} \sum_{p'=-1}^{k'} \left| C_p \cap C_p^+ \right| \log \left( \frac{|C_p \cap C_p^+|}{\sum_{p'=-1}^{k'} |C_p \cap C_p^+|} \right), \hspace{1cm} (52)$$

$$H(C^+) = -\sum_{p'=-1}^{k'} \frac{\sum_{p=-1}^{k} |C_p \cap C_p^+|}{k'} \log \left( \frac{\sum_{p=-1}^{k} |C_p \cap C_p^+|}{k'} \right). \hspace{1cm} (53)$$

$H(C | C^+) = 0$ when each cluster contains only members of a single class, a perfect homogeneous clustering. In the degenerate case when $H(C^+) = 0$, when there is only a single class, we define the homogeneity to be 1.

Completeness is symmetric to homogeneity. Therefore, by symmetry with to the calculation above, we define the completeness as

$$\text{comp}(C) = \begin{cases} 1, & \text{if } H(C | C^+) = 0 \\ 1 - \frac{H(C | C^+) H(C | C)}{H(C | C^+)} & \text{else} \end{cases}. \hspace{1cm} (54)$$
where

$$H(C|C^+) = -\sum_{p=1}^{k^+} \frac{1}{k} \sum_{p=1}^{k^+} \frac{|C_p \cap C_{p^*}|}{n} \log \left( \frac{|C_p \cap C_{p^*}|}{\sum_{p=1}^{k^+} |C_p \cap C_{p^*}|} \right).$$

(55)

$$H(C) = -\sum_{p=1}^{k} \frac{1}{k} \sum_{p=1}^{k} \frac{|C_p \cap C_{p^*}|}{k^*} \log \left( \frac{|C_p \cap C_{p^*}|}{\sum_{p=1}^{k^*} |C_p \cap C_{p^*}|} \right).$$

(56)

Based upon these calculations of the homogeneity and completeness, we calculate the V-measure of a clustering solution by computing the harmonic mean of the homogeneity and completeness, just as the precision and recall are commonly combined into the F-measure:

$$V(C) = \frac{2 \hom(C) \comp(C)}{\hom(C) + \comp(C)} = \frac{2 \hom(C) \comp(C)}{\hom(C) + \comp(C)}.$$  

(57)

Notice that the computation of the homogeneity, the completeness and the V-measure are completely independent from the number of classes, the number of the clusters, the size of the dataset and the clustering algorithm used.

6. Experiments

In this section, to test the effectiveness of the proposed approach, we compare the performance of the weighted criterion functions with the unweighted criterion functions and three methods, studied in [51]. We first describe the datasets used in the experiments.

6.1. Experimental data

We used datasets with a wide variety of numbers of clusters, numbers of documents and numbers of terms:

- **webkb**. The webkb dataset contains web pages collected from computer science departments of various universities. There are 8282 documents and they are divided into 7 categories: student, faculty, staff, department, course, project and other. This dataset is available from [53].
- **re0**. This dataset is the subset of the Reuters-21578 Text Categorization Test collection containing the 13 most frequent categories among the 135 topics [54].
- **rec**. The rec dataset containing autos, motorcycles, baseball and hockey, which was selected from the version 20News-18828 and contains 3970 documents [55].
- **wap**. This dataset contains 1560 documents consisting of news articles from 20 different topics in October 1997 collected in the WebACE project [28].
- **fbis**. The fbis dataset is derived from the TREC-5 collection. This dataset is available from [56].

These datasets are standard text datasets that are often used as benchmarks for document clustering [30,39,51]. General characteristics of the datasets are summarized in Table 1.

<table>
<thead>
<tr>
<th>Data</th>
<th>Number of documents</th>
<th>Number of classes</th>
<th>Number of terms</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Before preprocessing</td>
<td>After preprocessing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>webkb</td>
<td>8282</td>
<td>7</td>
<td>20682</td>
<td>3000</td>
<td>Web knowledge base</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Webpages</td>
<td>[53] Web pages</td>
</tr>
<tr>
<td>re0</td>
<td>1504</td>
<td>13</td>
<td>2886</td>
<td>500</td>
<td>Reuters-21578</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reuters-21578 [54]</td>
<td>Newsgroup posts</td>
</tr>
<tr>
<td>rec</td>
<td>3970</td>
<td>4</td>
<td>16783</td>
<td>2000</td>
<td>20News-18828</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20News-18828 [55]</td>
<td>Newsgroup posts</td>
</tr>
<tr>
<td>wap</td>
<td>1560</td>
<td>20</td>
<td>8460</td>
<td>1000</td>
<td>WebACE [28]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Webpages</td>
<td>[28] Web pages</td>
</tr>
<tr>
<td>fbis</td>
<td>2463</td>
<td>17</td>
<td>12764</td>
<td>1500</td>
<td>TREC-5 [56]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Newspaper articles</td>
<td>[56]</td>
</tr>
</tbody>
</table>

6.2. Preprocessing

First, we removed stopwords. These are words that are non-descriptive for the topic of a document. Following the common practice, we used a stoplist provided in [52]. Second, words were stemmed using Porter’s suffix-stripping algorithm [44], so that words with different endings would be mapped to a single word. The underlying assumption is that different morphological variations of words with the same root/stem are thematically similar and should be treated as a single word. In our experiments, we also considered the effect of including terms with small weights in the document representation on
the overall clustering performance, and decided to discard words that appear with less than a given threshold weight. The rationale behind discarding terms with small weights is that in many cases they are not very descriptive of the document’s subject and make little contribution to the similarity between two documents. Besides, the terms with small weights can also introduce noise into the clustering process and make the similarity computation more expensive. Consequently, we selected the top 3000, 500, 2000, 1000, 1500 terms ranked by their weights defined by (1) for the webkb, re0, rec, wap, and fbis datasets, respectively, and used them in our experiments.

6.3. Choice of parameters for DE algorithm and the simulation strategy

The modified DE algorithm has a number of control parameters that affect its performance on different datasets. In this section we discuss the influence of parameters like the population size \( N \), the scaling factors for crossover \( k_c \) and mutation \( k_m \), the crossover constant \( pr_c \), and the mutation constant \( pr_m \).

**Population size.** To investigate the effect of the population size \( N \), the DE was executed separately with 400–1000, 50–250, 200–600, 50–250 and 100–400 chromosomes (keeping all other parameter settings same as reported in Table 2) for the webkb, re0, rec, wap and fbis datasets, respectively. Experiments showed that numbers of chromosomes more than 600 (webkb), 150 (re0), 400 (rec), 200 (wap) and 300 (fbis) produced more or less identical clustering results for DE.

**The scaling factors.** Provided all other parameters were fixed at the values given in Table 2, we let DE run over different settings of the scaling factors \( k_c \) and \( k_m \). We used \( k_c = 0.5 \) and \( k_m = 0.4 \), \( k_c = 0.6 \) and \( k_m = 0.4 \), \( k_c = 0.3 \) and \( k_m = 0.7 \), \( k_c = 0.5 \) and \( k_m = 0.7 \), \( k_c = k_m = 0.7 \), \( k_c = 0.9 \) and \( k_m = 0.7 \), \( k_c = k_m = 0.8 \). We noted that the scaling factors \( k_c = 0.9 \) and \( k_m = 0.7 \) gave the best clustering results over all the datasets considered.

**The crossover constant.** Provided all other parameters were fixed at the values shown in Table 2, the DE was run with several possible choices of the crossover constant \( pr_c \). Specifically we used random \( pr_c \). \( pr_c = 0.2 \), \( pr_c = 0.5 \), \( pr_c = 0.6 \), \( pr_c = 0.8 \), and finally \( pr_c = 0.9 \). It was observed that for all the datasets, the best convergence behavior of DE was obtained for \( pr_c = 0.8 \).

**The mutation constant.** Provided all other parameters were fixed at the values shown in Table 2, the DE was run with several possible choices of the mutation constant \( pr_m \). Specifically we used random \( pr_m \). \( pr_m = 0.1 \), \( pr_m = 0.2 \), \( pr_m = 0.4 \), \( pr_m = 0.5 \), \( pr_m = 0.7 \), and finally \( pr_m = 0.8 \). It was observed that for all the datasets, the best convergence behavior of DE was obtained for \( pr_m = 0.5 \).

The optimization procedure used here is stochastic in nature. Hence, it was run several times for each criterion function. The results reported in this section are averages over 50 runs for each criterion function. Each run was continued up to 1000 fitness evaluations. Table 2 lists all the parameter settings used for all the criterion functions. In the experiments, the number of clusters is set to be the same as the number of pre-assigned classes in the datasets for all the clustering methods.

Finally, we would like to point out that the algorithm discussed here was developed in the Delphi 7 platform on a Pentium Dual CPU, 1.6 GHz PC, with 512 KB cache, and 1 GB of main memory in Windows XP environment.

6.4. Experimental results and analysis

In this subsection, we analyze the results of the experiment from different points of view. First, we show the efficiency of assigning weights to documents. For this purpose we compare clustering solutions obtained by the weighted (6)–(11) and unweighted (12)–(17) criterion functions. Second, we conduct comprehensive performance evaluations by comparing our methods with the methods \( F_2 \), \( E_1 \) and \( W_2 \), which are the best of the seven different global criterion functions studied in [51].

1. The \( F_2 \) criterion function is

\[
F_2 = \sum_{p=1}^{k} \sum_{i=1}^{N_c} \sin(D_i, O_p) \rightarrow \text{max}.
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clusters, ( k )</td>
<td></td>
</tr>
<tr>
<td>Number of generation</td>
<td></td>
</tr>
<tr>
<td>Population size, ( N )</td>
<td></td>
</tr>
<tr>
<td>Number of iteration (fitness evaluation), ( t_{max} )</td>
<td></td>
</tr>
<tr>
<td>The scaling factor for crossover, ( k_c )</td>
<td></td>
</tr>
<tr>
<td>The crossover constant, ( pr_c )</td>
<td></td>
</tr>
<tr>
<td>The scaling factor for mutation, ( k_m )</td>
<td></td>
</tr>
<tr>
<td>The mutation constant, ( pr_m )</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Parameters setup of the DE for different datasets.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>webkb</th>
<th>re0</th>
<th>rec</th>
<th>wap</th>
<th>fbis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clusters, ( k )</td>
<td>7</td>
<td>13</td>
<td>4</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>Number of generation</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Population size, ( N )</td>
<td>600</td>
<td>150</td>
<td>400</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>Number of iteration (fitness evaluation), ( t_{max} )</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>The scaling factor for crossover, ( k_c )</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>The crossover constant, ( pr_c )</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>The scaling factor for mutation, ( k_m )</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>The mutation constant, ( pr_m )</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
The $\mathcal{F}_2$ criterion function is used by the popular vector space variant of the k-means algorithm. In this algorithm, each cluster is represented by its centroid and the goal is to find the solution that maximizes the similarity between each document and the centroid of the cluster to which it is assigned.

2. The $\mathcal{E}_1$ criterion function is
\[
\mathcal{E}_1 = \sum_{p=1}^{k} |C_p| \cdot \text{sim}(O_p, O) \rightarrow \min.
\]

The $\mathcal{E}_1$ criterion function computes the clustering by finding a solution that separates the documents of each cluster from the entire collection. Specifically, it tries to minimize the similarity between the centroid vector of each cluster and the centroid vector of the entire collection.

3. The $\mathcal{H}_2$ criterion function is
\[
\mathcal{H}_2 = \frac{\mathcal{F}_2}{\mathcal{E}_1} \rightarrow \max.
\]

The $\mathcal{H}_2$ criterion function is obtained by combining $\mathcal{F}_2$ with $\mathcal{E}_1$. Since $\mathcal{E}_1$ is minimized, $\mathcal{H}_2$ needs to be maximized as it is inversely related to $\mathcal{E}_1$.

The clustering results are shown in Tables 3–7, which give a comparative analysis of the results of the weighted and unweighted criterion functions judged by the ten validity indices on webkb, re0, rec, wap and fbis datasets, respectively. In these tables, the comparative analysis of the methods $\mathcal{F}_2$ and $\mathcal{H}_2$ is also shown. Table 8 gives a comparative analysis of the methods judged by the average values of the validity indices on all datasets which are obtained from Tables 3–7.

From Tables 3–8 we make the following main observations:

- The weighted criterion functions produced better solutions than the unweighted criterion functions for all datasets. The best entries of Tables 3–8 have been marked in boldface. The improvement is shown in brackets (missing values indicate a zero improvement). Here, we used the relative improvement $\frac{\text{weighted method} - \text{unweighted method}}{\text{unweighted method}} \times 100$ for the indices $\text{CS}_1$, $\text{CS}_2$, $\text{CS}_3$, $\text{Purity}$, $\text{F-measure}$, and $\text{PC}$. For the indices $\text{Entropy}$, $\text{Mirkin metric}$, $\text{VI}$ and $\text{V-measure}$, we used the relative improvement $\frac{\text{unweighted method} - \text{weighted method}}{\text{unweighted method}} \times 100$.

- It is easy to see that the internal validity indices were more sensitive to the document weighting than the external validity indices. Of the latter, $\text{F-measure}$ and $\text{PC}$ were the most sensitive. For this purpose, it is necessary to pay attention to improvement percentage.

- Comparison of the clustering solutions produced by our criterion functions (except $\mathcal{F}_3$) with the "ground truth" results showed that our methods yield high accuracy. For this purpose it is enough to pay attention to the values of the $\text{F-measure}$ and $\text{PC}$ indices, which reveal accuracy of approximately 80% and 77%, respectively.

- Among the weighted functions, $\mathcal{F}_2$ is the most sensitive to the weighting. It improved the results of the function $\mathcal{F}_3$ as follows: 42.17% ($\text{CS}_1$), 71.65% ($\text{CS}_2$), 22.53% ($\text{CS}_3$), 0.40% (Purity), 0.18% (Entropy), 7.47% (Mirkin), 4.16% (F-measure), 15.36% (VI), 4.64% (PC) and 0.02% (V-measure) (see Table 8).

Table 3

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validity indices</th>
<th>V-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\text{CS}_1$</td>
<td>$\text{CS}_2$</td>
</tr>
<tr>
<td>$\mathcal{F}_2$</td>
<td>4.8695</td>
<td>6.9276</td>
</tr>
<tr>
<td></td>
<td>(12.92%)</td>
<td>(6.08%)</td>
</tr>
<tr>
<td>$\mathcal{F}_3$</td>
<td>4.3123</td>
<td>6.5308</td>
</tr>
<tr>
<td></td>
<td>(9.91%)</td>
<td>(53.42%)</td>
</tr>
<tr>
<td>$\mathcal{H}_2$</td>
<td>4.1057</td>
<td>8.9568</td>
</tr>
<tr>
<td></td>
<td>(42.86%)</td>
<td>(8.22%)</td>
</tr>
<tr>
<td>$\mathcal{F}_4$</td>
<td>0.2984</td>
<td>0.3224</td>
</tr>
<tr>
<td></td>
<td>(21.18%)</td>
<td>(21.01%)</td>
</tr>
<tr>
<td>$\mathcal{F}_5$</td>
<td>7.6039</td>
<td>8.1230</td>
</tr>
<tr>
<td></td>
<td>(12.62%)</td>
<td>(12.04%)</td>
</tr>
<tr>
<td>$\mathcal{F}_6$</td>
<td>5.3466</td>
<td>7.4445</td>
</tr>
<tr>
<td></td>
<td>(33.61%)</td>
<td>(22.39%)</td>
</tr>
<tr>
<td>$\mathcal{F}_7$</td>
<td>5.9937</td>
<td>7.2449</td>
</tr>
<tr>
<td></td>
<td>(6.98%)</td>
<td>(5.69%)</td>
</tr>
<tr>
<td>$\mathcal{F}_8$</td>
<td>0.4473</td>
<td>0.4129</td>
</tr>
</tbody>
</table>
The external validity index \textit{V-measure} does not possess discriminative ability, i.e., its value on various datasets was almost identical for all methods. From this finding, it is possible to draw the conclusion that the use of the index \textit{V-measure} is not expedient for evaluating clustering results. Therefore, in the following comparisons, we did not consider the results of the index \textit{V-measure} (see Tables 9 and 10).

- The method \(\mathcal{F}_1\) gave better results than \(\mathcal{F}_2\).
- The criterion function \(\mathcal{F}_2\) (weighted \(k\)-means) outperformed the \(\mathcal{F}_2\) \((k\text{-means})\) method.

From Table 8, we obtained the ranks of the methods by each of the indices, the results of which are shown in Table 9 (as was mentioned before, in this table the results of the \textit{V-measure} index were not taken into consideration).

Hence the function \(\mathcal{F}_1^1\) had the best rank according to the five indices \(\text{CS}_1\), \textit{Purity}, \textit{Entropy}, \textit{F-measure} and \(\text{PC}\), according to the three indices \(\text{CS}_2\), \(\text{CS}_3\) and \textit{Mirkin} the function \(\mathcal{F}_1^1\) is the best, and according to the index \(\text{VI}\) the function \(\mathcal{F}_1^1\) demonstrates the best results. From this table we can also see that the function \(\mathcal{F}_1\) showed the worst results under eight indices (out of nine) and the function \(\mathcal{F}_2\) had the worst rank under just one index (\textit{Purity}).
Table 6
Values of the validity indices for clustering methods obtained with the web (WebACE) dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validity indices</th>
<th>V-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$CS_1$</td>
<td>$CS_2$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>0.8550</td>
<td>1.5252</td>
</tr>
<tr>
<td></td>
<td>$F_2$</td>
<td>0.7672</td>
</tr>
<tr>
<td></td>
<td>$F_3$</td>
<td>0.8331</td>
</tr>
<tr>
<td></td>
<td>$F_4$</td>
<td>0.7280</td>
</tr>
<tr>
<td></td>
<td>$F_5$</td>
<td>0.3277</td>
</tr>
<tr>
<td></td>
<td>$F_6$</td>
<td>0.2681</td>
</tr>
<tr>
<td></td>
<td>$F_7$</td>
<td>1.0282</td>
</tr>
<tr>
<td></td>
<td>$F_8$</td>
<td>0.9182</td>
</tr>
<tr>
<td></td>
<td>$F_9$</td>
<td>0.7438</td>
</tr>
<tr>
<td></td>
<td>$F_{10}$</td>
<td>0.7206</td>
</tr>
<tr>
<td></td>
<td>$F_{11}$</td>
<td>0.7819</td>
</tr>
<tr>
<td></td>
<td>$F_{12}$</td>
<td>0.7819</td>
</tr>
<tr>
<td></td>
<td>$F_{13}$</td>
<td>0.5480</td>
</tr>
<tr>
<td></td>
<td>$F_{14}$</td>
<td>0.5407</td>
</tr>
<tr>
<td></td>
<td>$F_{15}$</td>
<td>0.2447</td>
</tr>
<tr>
<td></td>
<td>$F_{16}$</td>
<td>0.5501</td>
</tr>
</tbody>
</table>

Table 7
Values of the validity indices for clustering methods obtained with the fibs (TREC-5) dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validity indices</th>
<th>V-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$CS_1$</td>
<td>$CS_2$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>0.9452</td>
<td>1.0657</td>
</tr>
<tr>
<td></td>
<td>$F_2$</td>
<td>0.7779</td>
</tr>
<tr>
<td></td>
<td>$F_3$</td>
<td>0.5745</td>
</tr>
<tr>
<td></td>
<td>$F_4$</td>
<td>0.5000</td>
</tr>
<tr>
<td></td>
<td>$F_5$</td>
<td>0.2869</td>
</tr>
<tr>
<td></td>
<td>$F_6$</td>
<td>0.7136</td>
</tr>
<tr>
<td></td>
<td>$F_7$</td>
<td>0.1230</td>
</tr>
<tr>
<td></td>
<td>$F_8$</td>
<td>1.1136</td>
</tr>
<tr>
<td></td>
<td>$F_9$</td>
<td>0.7136</td>
</tr>
<tr>
<td></td>
<td>$F_{10}$</td>
<td>0.6982</td>
</tr>
<tr>
<td></td>
<td>$F_{11}$</td>
<td>0.1628</td>
</tr>
</tbody>
</table>

To obtain the resulting ranks of the methods we transformed Table 9 into another one, shown in Table 10. The resultant rank in Table 10 was computed according to the following formula: Resultant rank(method) = $\sum_{s=1}^{14} \left( \frac{r_s}{14} \right)$, where $r_s$ denotes the number of times the method appears in the $s$th rank. For instance, the rank of the method $F_2$ is
From Table 10 we conclude the following:

- Among the methods, the $F^+_2$ method showed the best results.
- The $F^+_3$ method showed the worst result among the weighted criterion functions.
- The methods $F^+_1$ and $F^+_3$, $F_1$ and $F_2$, $F^+_2$ and $F^+_6$, $F_5$ and $F_6$ showed close results.
- Our methods outperformed the other three methods, $F_2$, $F_2$ w $e_1$. Among our methods, only $F^+_3$ was outranked by the other methods $F_2$ and $F_2$.
- The $e_1$ method showed the worst result among the methods $F_2$, $F_2$ and $e_1$. Note that this result conforms with the result received in [51].
- Comparing the $k$-means ($F_2$) method with its weighed variants, $F^+_2$ and $F_2$, we see that it was outperformed by both methods.

In order to show the comparison of methods more descriptively, we demonstrate it in histograms. Fig. 2, below, shows a graphical comparison of the methods.
Conclusion

In this paper, we studied twelve criterion functions for document clustering. Six of these functions were weighted, whereas the remaining six were unweighted. In our study, a weight was assigned to each document, which defined its relative position in the entire collection. The purpose of the present paper was to show that weighting improves the clustering solution. To show the efficiency of the proposed approach, the weighted methods were compared to their unweighted variants. The comparison was conducted on five datasets with widely varying numbers of clusters, documents and terms. The quality of a clustering result was evaluated using ten validity indices: three internal validity indices and seven external validity indices. The internal validity indices were used to evaluate the within-cluster scatter and between cluster separations. The external validity indices were used to compare the clustering solution produced by the proposed criterion functions with the “ground truth” results. The experiments showed that the weighted criterion functions lead to reasonably good results that outperform the results obtained from the unweighted criterion functions. Furthermore, to study the performance of our methods, we compared the methods against three clustering methods implemented in [51], $E_1$, $H_2$ and $I_2$, where $I_2$ is the vector space variant of the $k$-means method. The experimental results showed that our methods outperform the $E_1$, $H_2$ and $I_2$ ($k$-means) methods. Out of our methods, only $F_a^4$ was outperformed by the methods $F_a^2$ and $F_a^6$. In this paper, we developed a modified DE algorithm to optimize the criterion functions. An important feature of the proposed modification is that it accelerates the convergence of DE, and unlike a basic DE algorithm it guarantees that the solution it produces will be feasible.

Acknowledgement

The author would like to thank all the anonymous reviewers for their valuable comments to improve the quality of this paper.

Table 10
The resultant rank of the methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>The number of times the method is in the $s$th rank</th>
<th>Resultant rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s = 1$</td>
<td>$5$ $3$ $1$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$</td>
<td>$8.6429$</td>
</tr>
<tr>
<td>$s = 2$</td>
<td>$3$ $1$ $2$ $1$ $1$ $0$ $0$ $0$ $1$ $0$ $0$</td>
<td>$7.5714$</td>
</tr>
<tr>
<td>$s = 3$</td>
<td>$1$ $3$ $2$ $0$ $2$ $0$ $0$ $0$ $1$ $0$ $0$</td>
<td>$7.3571$</td>
</tr>
<tr>
<td>$s = 4$</td>
<td>$0$ $1$ $2$ $3$ $0$ $2$ $1$ $0$ $0$ $0$ $0$</td>
<td>$6.8571$</td>
</tr>
<tr>
<td>$s = 5$</td>
<td>$0$ $0$ $0$ $5$ $1$ $3$ $0$ $0$ $0$ $0$ $0$</td>
<td>$6.5714$</td>
</tr>
<tr>
<td>$s = 6$</td>
<td>$0$ $0$ $1$ $0$ $3$ $2$ $3$ $0$ $0$ $0$ $0$</td>
<td>$6.0000$</td>
</tr>
<tr>
<td>$s = 7$</td>
<td>$0$ $1$ $1$ $0$ $0$ $0$ $0$ $4$ $1$ $2$ $0$</td>
<td>$4.9286$</td>
</tr>
<tr>
<td>$s = 8$</td>
<td>$0$ $0$ $0$ $0$ $2$ $0$ $1$ $3$ $0$ $3$</td>
<td>$4.5714$</td>
</tr>
<tr>
<td>$s = 9$</td>
<td>$0$ $0$ $0$ $0$ $0$ $1$ $4$ $0$ $1$ $3$</td>
<td>$4.4286$</td>
</tr>
<tr>
<td>$s = 10$</td>
<td>$0$ $0$ $0$ $0$ $0$ $1$ $0$ $2$ $5$ $1$</td>
<td>$4.1429$</td>
</tr>
<tr>
<td>$s = 11$</td>
<td>$0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $8$</td>
<td>$2.5000$</td>
</tr>
<tr>
<td>$s = 12$</td>
<td>$0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$</td>
<td>$1.7857$</td>
</tr>
<tr>
<td>$s = 13$</td>
<td>$0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$</td>
<td>$1.4286$</td>
</tr>
<tr>
<td>$s = 14$</td>
<td>$0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$ $0$</td>
<td>$0.7143$</td>
</tr>
</tbody>
</table>

Fig. 2. The comparison of the methods based on the resultant rank.

7. Conclusion

In this paper, we studied twelve criterion functions for document clustering. Six of these functions were weighted, whereas the remaining six were unweighted. In our study, a weight was assigned to each document, which defined its relative position in the entire collection. The purpose of the present paper was to show that weighting improves the clustering solution. To show the efficiency of the proposed approach, the weighted methods were compared to their unweighted variants. The comparison was conducted on five datasets with widely varying numbers of clusters, documents and terms. The quality of a clustering result was evaluated using ten validity indices: three internal validity indices and seven external validity indices. The internal validity indices were used to evaluate the within-cluster scatter and between cluster separations. The external validity indices were used to compare the clustering solution produced by the proposed criterion functions with the “ground truth” results. The experiments showed that the weighted criterion functions lead to reasonably good results that outperform the results obtained from the unweighted criterion functions. Furthermore, to study the performance of our methods, we compared the methods against three clustering methods implemented in [51], $E_1$, $H_2$ and $I_2$, where $I_2$ is the vector space variant of the $k$-means method. The experimental results showed that our methods outperform the $E_1$, $H_2$ and $I_2$ ($k$-means) methods. Out of our methods, only $F_a^4$ was outperformed by the methods $F_a^2$ and $F_a^6$. In this paper, we developed a modified DE algorithm to optimize the criterion functions. An important feature of the proposed modification is that it accelerates the convergence of DE, and unlike a basic DE algorithm it guarantees that the solution it produces will be feasible.

Acknowledgement

The author would like to thank all the anonymous reviewers for their valuable comments to improve the quality of this paper.


Y. Chen, L. Tu, Density-based clustering for real-time stream data, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’07), San Jose, USA, 2007, pp. 133–142.


