Document clustering using a modified differential evolution algorithm

R. M. Aliguliyev
Institute of Information Technology of National Academy of Sciences of Azerbaijan, Baku
e-mail: a.ramiz@science.az, aramiz@iit.ab.az

Document clustering algorithms play important role in helping users to get relevant information, navigate, summarize and organize an enormous amount of textual document available on the Internet, and in digital libraries, news sources, and company-wide intranets. The objective of our work is to develop a high quality criterion functions for partitional document collection. In present paper we introduce two weighted criterion functions. We've also proposed a modified differential evolution algorithm to optimize the criterion functions. The proposed methods experimentally were evaluated on WebKb dataset. Experiments showed that the weighted criterion functions outperform the results obtained by the unweighted criterion functions.

Keywords: document clustering, weighted k-means, modified differential evolution.

Introduction

Clustering is useful for organization, summarization and navigation of semi-structured web pages and has been investigated as a fundamental operation in many areas such as data mining [1], information retrieval [2, 3], topic detection [4, 5], and as a preprocessing step for other algorithms such as document summarization [6 – 12]. For instance, in [6 – 9] the authors show that using clustering techniques for preprocessing in document summarization is a viable and effective technique. Unlike document classification, no labeled documents are provided in clustering; hence, clustering is also known as unsupervised learning. The methods used for document clustering covers several research areas, such as database, information retrieval, and artificial intelligence including machine learning and natural language processing [1 – 3, 13, 14].

High quality document clustering algorithms play important role in helping users to get relevant information, navigate, summarize and organize an enormous amount of textual document available on the Internet, and in digital libraries, news sources, and company-wide intranets. The objective of our work is to develop a high quality criterion functions for partitional document collection. In present paper we introduce two weighted criterion functions. We've also proposed a modified differential evolution algorithm to optimize the criterion functions. The proposed methods experimentally were evaluated on WebKb dataset.

1. Related Work

Clustering methods have been addressed in many contexts and disciplines such as data mining, document retrieval, image segmentation and pattern classification The prevalent

clustering algorithms have been categorized in different ways depending on different criteria. As with many clustering algorithms, there is trade-off between speed and quality of results. The existing clustering algorithms can be simply classified into the following two categories: *hierarchical* clustering and *partitional* clustering. A comprehensive survey of the various clustering algorithms can be found in [15, 16].

From the complexity of the problem to find the best clustering or the least very good clusterings for a given set of n objects, it should be clear, that a hierarchical method can only be feasible in very specific cases. But such a method could be used for an initial clustering. More important are the so-called *partitioning algorithms* where contrary to the hierarchical methods each object will be attached individually to a cluster. *Partitional* clustering algorithms assign each object to a partition. In recent years, it has been recognized that the partitional clustering technique is well suited for clustering a large document database due to their relatively low computational requirements [17].

The *k-means* method [15, 16, 18] is a commonly used partitional clustering method in information retrieval and other related research areas. Given from n objects, the method first select k objects as initial k clusters. It then 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 [19, 20]. In [21] presented a novelty-based document clustering method by proposing a new algorithm based on the *k-means* method. The novelty-based clustering method is based on a novelty-based similarity measure, which is an extension of a traditional approach in information retrieval, the cosine similarity measure in the VSM [2]. Novelty-based document clustering is a document clustering technique that puts high weights on recent documents and low weights on old ones. In [20] is proposed a generic iterative clustering scheme that, coupled with some particular reweighting scheme, may indeed bring improvements over "classical" clustering from the theoretical standpoint. This iterative clustering scheme can be specialized to bring weighted variants of *k-means*, fuzzy *k-means*, Expectation Maximization, and harmonic means clustering, among others [16, 18]. The experimental results clearly display differences in the benefits of the weighting scheme depending on the original clustering algorithms.

A major problem of using the existing methods is the selection of variables. The existing algorithms cannot select variables automatically because they treat all variables equally in the clustering process. In [22] is presented a new *k-means* type algorithm called $W-k$-*means* that can automatically weight variables based on the importance of the variables 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 in deciding the cluster memberships of objects in the next iteration. The weights can be used to identify important variables for clustering and the variables which may contribute noise to the clustering process and can be removed from the data in the future analysis.

2. Document Clustering

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 [3].

2.1. Document Representation Model

Let given the collection of documents $\mathbf{D} = (D_1, D_2, \ldots, D_n)$, where n is the number of documents in the collection. Let $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ represent all the terms occurred in the document collection $\mathbf{D}$. Here m is the number of unique terms in the document collection. In the most clustering algorithms, the dataset to be clustered is represented as a set of vector, where each vector corresponds to a single object and is called the feature vector. The feature vector should include proper features to represent the object.

We represent each document using the VSM. In this model each document D_i is located as a point in a 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. The component w_{ij} is defined using the scheme tf-idf. The tf-idf scheme combines the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. This weighting scheme assigns to term a weight in document given by

$$w_{ij} = n_{ij} \times \log\left(\frac{n}{n_j}\right), \tag{1}$$

where n_{ij} is the term frequency (i. e. denotes how many term T_j occurs in document D_i), n_j denotes the number of documents in which term T_j appears. The term $\log(n/n_j)$, which is very often referred to as the idf factor, accounts for the global weighting of term T_j. The idf factor has been introduced to improve the discriminating power of terms in the traditional information retrieval.

In other words, tf-idf assigns to term T_j a weight in document D_i that is

- highest when T_j occurs many times within a small number of documents (thus lending high discriminating power to those documents);
- lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
- lowest when the term occurs in virtually all documents.

2.2. Similarity Measure

The cosine measure has been one of the most popular document similarity measures due to its sensitivity to document vector pattern. The cosine measure computes the cosine of the angle between two feature vectors and is used frequently in text mining where vectors are very large but sparse. The cosine similarity between two documents D_i and D_l calculate as:

$$\cos(D_i, D_l) = \frac{\sum\limits_{j=1}^{m} w_{ij} w_{lj}}{\sqrt{\sum\limits_{j=1}^{m} w_{ij}^2 \cdot \sum\limits_{j=1}^{m} w_{lj}^2}}, \quad i, l = 1, \ldots, n. \tag{2}$$

2.3. Weighted Clustering

A hard clustering $\mathbf{C}$ is a partition of the dataset $\mathbf{D}$ into mutually disjoint subsets $C_1, C_2, \ldots, C_k$ called clusters. Formally, $\mathbf{C} = (C_1, C_2, \ldots, C_k)$ such that $C_p \bigcap C_q = \varnothing$,

$p \neq q$ (i. e., two different clusters should have no documents in common) and $\bigcup_{p}^{k} C_p = D$ (i. e., each document should definitely be belonged to a cluster). We also assume that for all $q = 1, \ldots, k$ $C_q \neq \varnothing$ and $C_q \in \mathbf{D}$, 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.

Our study involves two weighted clustering criterion functions that are given in below. These criterion functions have been proposed in the context of partitional clustering algorithms:

$$\mathcal{F}_1^{\alpha} = \sum_{p=1}^{k} \sum_{D_i, D_l \in C_p} \frac{\cos(D_i, D_l)}{\alpha_i \alpha_l} \to \max, \tag{3}$$

$$\mathcal{F}_2^{\alpha} = \sum_{p=1}^{k} \sum_{D_i \in C_p} \frac{\cos(D_i, O_p)}{\alpha_i} \to \max. \tag{4}$$

In these clustering methods to each document is assigned a weight defining its position in a document collection [23, 24]:

$$\alpha_i = \frac{\cos(D_i, O)}{\sum_{l=1}^{n} \cos(D_l, O)}, \quad i = 1, \ldots, n, \tag{5}$$

where $\mathbf{O}$ is the center of the document collection $\mathbf{D}$.

The jth coordinate o_j of the center O calculate as: $o_j = \frac{1}{n} \sum_{i=1}^{n} w_{ij}$ $(j = 1, \ldots, m)$. The weight (5) defines the degree of relative similarity of the document D_i to the centre of the $\mathbf{D}$, i. e., this weight defines the position of the document D_i relative to the center of the entire collection $\mathbf{D}$. It is not difficult to see that $\sum_{i=1}^{n} \alpha_i = 1$.

The $\mathcal{F}_1^{\alpha}$ criterion function (3) maximizes the sum of the pairwise similarities between the documents assigned to each cluster. The $\mathcal{F}_2^{\alpha}$ criterion function (4) is the weighted version of the k-means algorithm.

To show efficiency of weight assignment to documents the criterion functions (3), (4) we shall compare with the following criterion functions

$$\mathcal{F}_1 = \sum_{p=1}^{k} \sum_{D_i, D_l \in C_p} \cos(D_i, D_l) \to \max, \tag{6}$$

$$\mathcal{F}_2 = \sum_{p=1}^{k} \sum_{D_i \in C_p} \cos(D_i, O_p) \to \max. \tag{7}$$

The criterion functions (6), (7) can be received from the functions (3), (4) at the assumption $\alpha_i = \alpha > 0$ for any i $(i = 1, \ldots, n)$. The criterion function (6) maximizes the sum of the average pairwise similarity between the documents assigned to each cluster. The criterion function (7) is the vector-space variant of k-means method [17].

2.4. Clustering using Differential Evolution

The evolutionary algorithms differ mainly in the representation of parameters (usually binary strings are used for genetic algorithms while parameters are real-valued for evolution strategies and differential evolution) and in the evolutionary operators [14].

2.4.1. Chromosome Encoding

For representing the ath chromosome of the population at the current generation (at time t) here the following notation has been used:

$$X_a(t) = [x_{a,1}(t), x_{a,2}(t), \ldots, x_{a,m_k}(t)], \tag{8}$$

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

According to this encoding the first m positions represent the m dimensions of the first cluster centre, the next m genes represent those of the second cluster centre, and so on. For example, let $m = 2$ and $k = 4$. Then $X = [0.16; 0.37; 0.23; 0.75; 0.82; 0.26; 0.94; 0.68]$ represents the four cluster centers $(0.16; 0.37)$, $(0.23; 0.75)$, $(0.82; 0.26)$ and $(0.94; 0.68)$.

2.4.2. Population Initialization

The k cluster centers encoded in each chromosome are initialized to k randomly chosen points from the n points $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$. This process is repeated for each of the N chromosomes in the population.

2.4.3. Fitness Computation

The fitness functions according to objective functions (3), (4) we define as follows:

$$f_1^{\alpha}(X) = \frac{1}{\mathcal{F}_1^{\alpha}(X)}, \tag{9}$$

$$f_2^{\alpha}(X) = \frac{1}{\mathcal{F}_2^{\alpha}(X)}, \tag{10}$$

so that minimization of these fitness functions (9), (10) lead to maximization of criterion functions (3), (4), respectively. Fitness functions, corresponding to the criterion functions (6), (7), define in a similar way.

2.4.4. Modified Crossover

The proposed version for the chromosome of the best current solution $X_b(t)$ randomly chooses two other chromosomes $X_a(t)$ and $X_c(t)$ $(b \neq a \neq c)$ from the same generation. Then it calculates the scaling difference $(1 - \lambda_{cr})(x_{c,s}(t) - x_{a,s}(t))$ and creates a trial offspring chromosome by adding the result to the chromosome $X_b(t)$ scaled by the factor λ_{cr}. Thus for the sth gene $(s = 1, 2, \ldots, m_k)$ $y_{b,s}(t+1)$ of child chromosome $Y_b(t+1)$, we have:

$$y_{b,s}(t+1) = \begin{cases} MaxMin(\lambda_{cr} x_{b,s}(t) + (1 - \lambda_{cr})(x_{c,s}(t) - x_{a,s}(t))) & if \quad \theta_s < pr_{cr}, \\ x_{b,s}(t), \quad otherwise. \end{cases} \tag{11}$$

The scaling factor, $\lambda_{cr} \in [0.5, 1.0]$ and the crossover constant, $pr_{cr} \in [0, 1]$, are control parameters, which set by the user. θ_s is the uniformly distributed random number within the range $[0, 1]$ chosen once for each $s \in \{1, \ldots, m_k\}$.

The function $MaxMin\left(x_s\left(t\right)\right)$ in formula (11) defines as:

$$MaxMin\left(x_s\left(t\right)\right) = \begin{cases} x_s^{\min}\left(t\right) + \delta_s\left(t\right), & if \quad x_s\left(t\right) \leq x_s^{\min}\left(t\right), \\ x_s\left(t\right), & if \quad x_s^{\min}\left(t\right) < x_s\left(t\right) < x_s^{\max}\left(t\right), \\ x_s^{\max}\left(t\right) - \delta_s\left(t\right), & if \quad x_s\left(t\right) \geq x_s^{\max}\left(t\right), \end{cases} \tag{12}$$

where $x_s^{\min}(t) = \min\limits_{\alpha \in \{1,2,\ldots,N\}} \left\{x_{a,s}\left(t\right)\right\}$, $x_s^{\max} = \max\limits_{\alpha \in \{1,2,\ldots,N\}} \left\{x_{a,s}\left(t\right)\right\}$, $\delta_s\left(t\right) = \dfrac{x_s^{\max}(t) - x_s^{\min}(t)}{n}$.

Differential evolution uses the principle of "survival of the fittest" in its selection process which may be expressed as:

$$X_b\left(t+1\right) = \begin{cases} Y_b\left(t+1\right), & if \quad f_z^{\alpha}\left(Y_b\left(t+1\right)\right) < f_z^{\alpha}\left(X_b\left(t\right)\right), \\ X_b\left(t\right), & otherwise, \end{cases} \tag{13}$$

where fitness functions $f_z^{\alpha}\left(x\right)\left(z = 1, 2\right)$ are defined by formulas (9), (10).

2.4.5. Modified Mutation

The mutation operation for target chromosome $X_b\left(t\right)$ is performed according to the following formulation:

$$y_{b,s}\left(t+1\right) = \begin{cases} MaxMin\left(\lambda_{mt} x_{b,s}\left(t\right) + \left(1 - \lambda_{mt}\right)\left(x_{b,r}\left(t\right) - x_{b,q}\left(t\right)\right)\right) & if \quad \eta_s < pr_{mt}, \\ x_{b,s}\left(t\right), & else, \end{cases} \tag{14}$$

with random indexes $s, q, r \in \{1, \ldots, m_k\}$, integer, mutually different, $s \neq q \neq r$. The control parameters — the scaling factor, $\lambda_{mt} \in [0.5, 1.0]$ and the mutation constant, $pr_{mt} \in [0, 1]$, select by the user. η_s is the uniformly distributed random number 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 is retained in the population.

It is obvious that direct application of the basic differential evolution algorithm can lead to the case that the obtained solution can be infeasible. The infeasible solution is that solution which coordinates are beyond interval $\left(x_s^{\min}, x_s^{\max}\right)$. Therefore for prevention occurrence of the infeasible solution we modify the crossover (11) and mutation (14) operators by introducing the function $MaxMin\left(x\left(t\right)\right)$.

2.4.6. Termination Criterion

The algorithm terminates when a maximum number of fitness calculation is achieved.

2.5. Validation

Cluster validation refers to quantitatively evaluating the quality of a clustering solution.

2.5.1. Internal Validity Indices

In this subsection we present three indices using different definitions of inter and intra-cluster connectivity. Many 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 of ignoring clusters with low densities. In [14] proposed a validity measure that can deal with this situation. This measure is a function of the ratio of the sum of within-cluster scatter to between-cluster separation:

$$\mathrm{CS}_1(k) = \frac{\sum_{p=1}^{k}\left\{\frac{1}{|C_p|}\sum_{D_i\in C_p}\min_{D_l\in C_p}\mathrm{sim}(D_i, D_l)\right\}}{\sum_{p=1}^{k}\left\{\max_{q=1,\ldots,k,q\neq p}\{\mathrm{sim}(O_p, O_q)\}\right\}}. \tag{15}$$

This cluster validity index is inspired by the work reported in [25] and has been suitably modified. The denominator term in (15) computes the largest similarity between cluster centers. The numerator in (15) 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, is defined the second measure:

$$\mathrm{CS}_2(k) = \frac{\sum_{p=1}^{k}\min_{D_l\in C_p}\{\mathrm{sim}(D_i, O_p)\}}{\sum_{p=1}^{k}\max_{q=1,\ldots,k,q\neq p}\{\mathrm{sim}(O_p, O_q)\}}, \tag{16}$$

the third measure we define as:

$$\mathrm{CS}_3(k) = \frac{\sum_{p=1}^{k}\frac{1}{|C_p|}\sum_{D_i\in C_p}\mathrm{sim}(D_i, O_p)}{\sum_{p=1}^{k}\mathrm{sim}(O_p, O)}. \tag{17}$$

The numerator in (17) measures the average similarity of documents to the cluster centers. The denominator term in (17) computes the sum of similarity between cluster centers and the centre of the entire collection.

The validity indexes (15)–(17) simultaneously take care of the compactness and separation factors into account while dealing with complex structure datasets. The largest value of these measures (15)–(17) indicates a valid optimal partition.

2.5.2. External Validity Indices

Our second set of validation will focused on comparing of the clustering results produced by the proposed criterion functions with the "ground truth" results. The quality of a clustering solution was measured by using different metrics. These metrics measure the matching of clusters computed by each method to the "ground truth" clusters, meaning they measure how close each clustering method is to the "correct" clustering that would be produced manually by a human. However, in situations where documents are already labeled, we can compare the clusters with the "true" class labels.

Assume that the dataset $\mathbf{D}$ is composed of the classes $\mathbf{C}^+ = \left(C_1^+, \ldots, C_{k_+}^+\right)$ (true clustering) and we apply a clustering procedure for finding clusters $\mathbf{C} = (C_1, \ldots, C_k)$ in this dataset. We present various indices to compare two partitions $\mathbf{C} = (C_1, \ldots, C_k)$ and $\mathbf{C}^+ = \left(C_1^+, \ldots, C_{k_+}^+\right)$.

Important classes of criteria for comparing clustering solution are based on counting the pairs of points on which two clustering agree/disagree. The best-known clustering distances based on point pairs is the *F-measure* [26].

F-measure. If we want to compare a clusters $\mathbf{C}$ to a classes $\mathbf{C}^+$, 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.

The F value of the cluster C_p and the class $C_{p^+}^+$ is just the harmonic mean of the precision and the recall [26]:

$$\mathrm{F}\left(C_p, C_p^+\right) = \frac{2P\left(C_p, C_{p^+}^+\right) R\left(C_p, C_{p^+}^+\right)}{P\left(C_p, C_{p^+}^+\right) + R\left(C_p, C_{p^+}^+\right)}. \tag{18}$$

Precision is calculated as the portion of cluster C_p that is the documents of class $C_{p^+}^+$, thus measuring how homogenous cluster C_p is with respect to class $C_{p^+}^+$:

$$\mathrm{P}\left(C_p, C_{p^+}^+\right) = \frac{\left|C_p \bigcap C_{p^+}^+\right|}{|C_p|}. \tag{19}$$

Similarly, recall is calculated as the portion of documents from class $C_{p^+}^+$ that are present in cluster C_p, thus measuring how complete cluster C_p is with respect to class $C_{p^+}^+$:

$$\mathrm{R}\left(C_p, C_{p^+}^+\right) = \frac{\left|C_p \bigcap C_{p^+}^+\right|}{\left|C_{p^+}^+\right|}. \tag{20}$$

The *F-measure* of cluster C_p is the maximum F value attained at any class in the entire classes $\mathbf{C}^+ = \left(C_1^+, \ldots, C_{k^+}^+\right)$. That is,

$$\mathrm{F}\left(C_p\right) = \max_{C_{p^+}^+ \in C^+} F\left(C_p, C_{p^+}^+\right), \quad p = 1, 2, \ldots, k. \tag{21}$$

The *F-measure* of the entire collection is defined to be the sum of the individual cluster specific *F-measures* weighted according to the cluster size. That is,

$$\mathrm{F}\left(\mathbf{C}\right) = \sum_{p=1}^{k} \frac{|C_p|}{n} F\left(C_p\right). \tag{22}$$

In general, the higher the *F-measure* values, the best clustering solution is.

Now we propose information-based methods to compare partitions $\mathbf{C} = (C_1, \ldots, C_k)$ and $\mathbf{C}^+ = \left(C_1^+, \ldots, C_{k^+}^+\right)$. The commonly used external validity indices based on information are *partition coefficient*, and *variation of information*. These measures represent plausible ways to evaluate the homogeneity of a clustering solution.

Partition Coefficient. The partition coefficient (PC) was introduced by Bezdek [27]. The partition coefficient measures the amount of overlap between clusters. Considering a cluster C_p, the PC defines as follows:

$$\mathrm{PC}\left(C_p\right) = \sum_{p^+=1}^{k^+} \left(\frac{\left| C_p \bigcap C_{p^+}^+ \right|}{|C_p|} \right)^2 . \tag{23}$$

$\mathrm{PC}\left(C_p\right)$ is a value between $\frac{1}{k^+}$ and 1. If almost all documents of C_p belong to a same cluster in C_{p+}^+, then $\mathrm{PC}\left(C_p\right)$ is close to 1. Now, if documents of C_p are randomly divided into all clusters of $\mathbf{C}^+$ then $\mathrm{PC}\left(C_p\right)$ is close to $\frac{1}{k^+}$.

A global partition coefficient is computed:

$$\mathrm{PC}\left(\mathbf{C}, \mathbf{C}^+\right) = \frac{1}{k} \sum_{p=1}^{k} PC\left(C_p\right) = \frac{1}{k} \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} \left(\frac{\left| C_p \bigcap C_{p^+}^+ \right|}{|C_p|} \right)^2 . \tag{24}$$

$\mathrm{PC}\left(\mathbf{C}, \mathbf{C}^+\right)$ is also a value between $\frac{1}{k^+}$ and 1. Now, if $\mathrm{PC}\left(\mathbf{C}, \mathbf{C}^+\right)$ is close to $\frac{1}{k^+}$, $\mathbf{C}$ and $\mathbf{C}^+$ are almost independent. Moreover, if $\mathrm{PC}\left(\mathbf{C}, \mathbf{C}^+\right)$ is close to 1, then $\mathbf{C}$ is close to $\mathbf{C}^+$.

Variation of Information. Another information-based clustering measure is *variation of information* (*VI*) [28]:

$$\mathrm{VI}\left(\mathbf{C}, \mathbf{C}^+\right) = \frac{1}{n} \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} |C_p \cap C_{p^+}^+| \log \left(\frac{|C_p||C_{p^+}^+|}{|C_p \cap C_{p^+}^+|^2} \right). \tag{25}$$

The maximum value of the variation of information is $\log n$, which is achieved when the partitions are as far apart as possible, which in this case 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 one can simply normalize by $\log n$, as we do in the calculations presented here:

$$\mathrm{VI}\left(\mathbf{C}, \mathbf{C}^+\right) = \frac{1}{n \log n} \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} |C_p \cap C_{p^+}^+| \log \left(\frac{|C_p||C_{p^+}^+|}{|C_p \cap C_{p^+}^+|^2} \right). \tag{26}$$

In general, the smaller the variation of information values, the best clustering solution is.

3. Experiments

For our experiments we have used subset the WebKb dataset, as it is currently the most widely used benchmark in document clustering research. The subset contains 3.907 pages were manually classified into 7 categories. This dataset is available from [29].

In all the experiments discussed in this paper, stopwords were removed using the stoplist provided in [30] and the terms were stemmed using Porter's scheme [31], which is a commonly used algorithm for word stemming in English.

3.1. Simulation Strategy and Parameters

The optimization procedure used here is stochastic in nature. Hence, for each criterion function it has been run several times. The results reported in this section are averages over 50 runs for each criterion functions. Finally, we would like to point out that algorithm was developed from scratch in 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. For establishment of DE parameters the WebKb dataset has been broken in two parts: train and test. The train dataset contains 979 documents and the test dataset contains 2928 documents. After training of DE on train dataset for parameters following values are established: N*(population size)* $= 1000$, $t_{\max}$ *(number of fitness evaluation)* $= 500$, λ_{cr}*(the scaling factor for crossover)* $= 0.9$, pr_c*(the crossover constant)* $= 0.8$, λ_{mt} *(the scaling factor for mutation)* $= 0.7$, pr_m*(the mutation constant)* $= 0.6$. After setting the parameters of differential evolution experiments have been spent on test dataset.

3.2. Results and Analysis

We report experiments comparing our weighted versions of clustering algorithms with their unweighted versions. Table gives the comparative analysis of the results of weighted and unweighted criterion functions. As can be seen from Table 1 the weighted criterion functions produced the best results than unweighted criterion functions. Highlighted entries represent the best performing criterion function in terms of average validity indices. In brackets it is shown improvement in percentage. Here, we've used relative improvement

$$\frac{(weighted\ method - unweighted\ method)}{unweighted\ method} \times 100$$

for the indexes CS_1, CS_2, CS_3, *F-measure*, and *PC*. For the index *VI* we've used relative improvement

$$\frac{(unweighted\ method - weighted\ method)}{weighted\ method} \times 100.$$

Comparison the weighted and unweighted criterion functions

Criterion functions	Validity indices					
	CS_1	CS_2	CS_3	F-measure	VI	PC
$\mathcal{F}_1^\alpha$	**7.7527** (66.55 %)	**9.9513** (98.94 %)	**6.8687** (53.57 %)	**0.8294** (1.53 %)	**0.1267** (3 %)	**0.8040** (8.99 %)
$\mathcal{F}_1$	4.6548	5.0021	4.4726	0.8169	0.1305	0.7377
$\mathcal{F}_2^\alpha$	**0.0011** (57.14 %)	**0.0004** (100 %)	**0.4667** (7.88 %)	**0.6835** (17.80 %)	**0.3136** (5.26 %)	**0.6533** (20.42 %)
$\mathcal{F}_2$(k-means)	0.0007	0.0002	0.4326	0.5802	0.3301	0.5425

The experiments presented in this subsection showed interesting trends. The quality of the solutions produced by some seemingly similar criterion functions are often substantially different. For instance, both $\mathcal{F}_1^\alpha$ and $\mathcal{F}_2^\alpha$ find clusters by maximizing a particular within cluster similarity function. However, $\mathcal{F}_1^\alpha$ performs substantially better than $\mathcal{F}_2^\alpha$. This is also true for unweighted criterion functions $\mathcal{F}_1$ and $\mathcal{F}_2$. Finally, from the analysis it is possible to draw a conclusion, that internal validity indexes are more sensitive to weighing (assigning of weight to document) than external validity indexes.

Conclusion

In our study a simple weighed approach for document clustering is proposed. In this approach to each document assigned the weight defining its relative positions in a collection of documents. Experiments have shown that such approach improves clustering solution. To be convinced of it we compared the performance of the *k-means* method with the weighted variant of the *k-means* method proposed by us. The experimental results were shown that our criterion function outperforms the *k-means* method. For estimation of clustering solution have been used six validity indexes, three from them are internal validity indexes, other three — external validity indexes. In this paper to optimize the criterion functions we developed a modified DE algorithm. The proposed modification prevents the occurrence of infeasible solution in the work process of DE.

Acknowledgment

Author acknowledges the anonymous referee for fruitful comments, which did improve quality of the paper.

References

[1] Han J., Kamber M. Data mining: Concepts and techniques (2nd ed.). San Francisco, Morgan Kaufman, 2006.

[2] Baeza-Yates R., Ribeiro-Neto R. Modern Information Retrieval. N.Y.: Addison Wesley, ACM Press, 1999.

[3] Hammouda K.M., Kamel M.S. Efficient phrase-based document indexing for web document clustering // IEEE Transactions on Knowledge and Data Eng. 2004. Vol. 16, No. 10. P. 1279–1296.

[4] Allan J. (ed.). Topic Detection and Tracking: Event-Based Information Organization. USA, Kluwer Acad. Publ. Norwell, 2002.

[5] Kuo J.-J., Chen H.-H. Cross-document event clustering using knowledge mining from co-reference chains // Informat. Proc. and Management. 2007. Vol. 43, No. 2. P. 327–343.

[6] Alguliev R.M., Alyguliev R.M. Automatic text documents summarization through sentences clustering // J. Automat. and Informat. Sci. 2008. Vol.40, No. 9. P. 53–63.

[7] Alguliev R.M., Alyguliev R.M., Bagirov A.M. Global optimization in the summarization of text documents // Automat. Control and Comput. Sci. 2005. Vol. 39, No. 6. P. 42–47.

[8] Aliguliyev R.M. A new sentence similarity measure and sentence based extractive technique for automatic text summarization // Expert Systems with Appl. 2009. Vol. 36, No. 4. P. 7764–7772.

[9] Aliguliyev R.M. Automatic document summarization by sentence extraction // Comput. Technol. 2007. Vol. 12, No. 5. P. 5–15.

[10] Dunlavy D.M., O'Leary D.P., Conroy J.M., Schlesinger J.D. QCS: A system for querying, clustering and summarizing documents // Inform. Proc. and Management. 2007. Vol. 43, No. 6. P. 1588–1605.

[11] Hu P., He T., Ji D., Wang M. A study of Chinese text summarization using adaptive clustering of paragraphs // Proc. of the 4-th Intern. Conf. on Computer and Informat. Technol. (CIT'04). Wuhan, China. 2004. P. 1159–1164.

[12] Mana-Lopez M.J., De Buenaga M., Gomez-Hidalgo J.M. Multidocument summarization: An added value to clustering in interactive retrieval // ACM Transact. on Informat. Systems. 2004. Vol. 22, No. 2. P. 215–241.

[13] Al-Omary A.Y., Jamil M.S. A new approach of clustering based machine-learning algorithm // Knowledge-Based Systems. 2006. Vol. 19, No. 4. P. 248–258.

[14] Das S., Abraham A., Konar A. Automatic clustering using an improved differential evolution algorithm // IEEE Transact. on Systems, Man, and Cybernetics. Pt A. Systems and Humans. 2008. Vol. 38, No. 1. P. 218–237.

[15] Grabmeier J., Rudolph A. Techniques of cluster algorithms in data mining // Data Mining and Knowledge Discovery. 2002. Vol. 6, No. 4. P. 303–360.

[16] Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review // ACM Comput. Surveys. 1999. Vol. 31, No. 3. P. 264–323.

[17] Zhao Y., Karypis G. Empirical and theoretical comparisons of selected criterion functions for document clustering // Machine Learning. 2004. Vol. 55, No. 3. P. 311–331.

[18] Hammerly G., Elkan C. Alternatives to the k-means algorithm that find better clustering // Proc. of the 11-th ACM Intern. Conf. on Informat. and Knowledge Management (CIKM'02). Virginia, USA, 2002. P. 600–607.

[19] Li Y., Chung S.M., Holt J.D. Text document clustering based on frequent word meaning sequences // Data and Knowledge Eng. 2008. Vol. 64, No. 1. P. 381–404.

[20] Nock R., Nielsen F. On weighting clustering // IEEE Transact. on Pattern Analysis and Machine Intelligence. 2006. Vol. 28, No. 8. P. 1223–1235.

[21] Khy S., Ishikawa Y., Kitagawa H. A novelty-based clustering method for on-line documents // World Wide Web. 2008. Vol. 11, No. 1. P. 1–37.

[22] Huang J.Z., Ng M.K., Rong H., Li Z. Automated variable weighting in k-means type clustering // IEEE Transact. on Pattern Analysis and Machine Intelligence. 2005. Vol. 27, No. 5. P. 657–668.

[23] Aliguliyev R.M. Clustering of document collection — a weighting approach // Expert Systems with Appl. 2009. Vol. 36, No. 4. P. 7904–7916.

[24] Aliguliyev R.M. Performance evaluation of density-based clustering methods // Informat. Sci. 2009. Vol. 179, No. 20. P. 3583–3602.

[25] Chou C.H., Su M.C., Lai E. A new cluster validity measure and its application to image compression // Pattern Analysis and Appl. 2004. Vol. 7, No. 2. P. 205–220.

[26] Crescenzi V., Merialdo P., Missier P. Clustering web pages based on their structure // Data and Knowledge Eng. 2005. Vol. 54, No. 3. P. 279–299.

[27] Bezdek J.C., Pal N.R. Some new indexes of cluster validity // IEEE Transact. on Systems, Man and Cybernetics. Pt. B. Cybernetics. 1998. Vol. 28, No. 3. P. 301–315.

[28] Patrikainen A., Meila M. Comparing subspace clusterings // IEEE Transact. on Knowledge and Data Eng. 2006. Vol. 18, No. 7. P. 902–916.

[29] http://www.cs.cmu.edu

[30] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

[31] Porter M. An algorithm for suffix stripping // Program. 1980. Vol. 14, No. 3. P. 130–137.

Received for publication 24 February 2009