The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 2, 2019
Pages 14-29
Regular Paper

C-W-FCM: Constrained Weighted Fuzzy Clustering Algorithm with a Semi-Supervised Approach for Text Classification

Soheila Ramezani Pour[1] Marjan Naderan[1] Saeid-Allah Mortazavi[2]

[1]Department of Computer Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
[2]Department of Electrical Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran

Abstract

The emergence of digital information era and rapid development of the Internet makes information to change gradually from paper form to the electronic one. This makes the users capable to search the news and books in an electronic way. Thus, the existence of systems for information retrieval appears to be essential. This paper suggests a system for text classification by means of semi-supervised fuzzy clustering with a weighted feature vector. In the proposed method, after a preprocessing phase, a Genetic Algorithm together with the TF-IDF method is used for dimensionality reduction. Accordingly, features with highest discriminating power are chosen and finally, the documents are classified with the clustering algorithm, C-W-FCM. In fact, the proposed clustering algorithm applies the Euclidean distance with different weights for different dimensions. For evaluation of the proposed approach, a number of prominent criteria for clustering, namely Fukuyama and Sugeno (FS), are used conducted on the Reuters dataset. It is assumed that a small number of documents have labels which are called the seeded set. Simulation results show that the proposed approach is 27 to 33% superior to conventional clustering algorithms based on the evaluation criteria in determining clusters. In addition, the proposed clustering algorithm increases the system effectiveness especially when documents are highly similar to each other.

Keywords: Text classification, fuzzy clustering, semi-supervised, genetic algorithm

1. Introduction

In the age of digital information, the rapid increase in the amount of published information, data, and statistics has given birth to a data explosion. On the other hand, raw or unprocessed data has little value; in fact, the practical piece lay in the information it contains. A huge bunch of data stored in the text database brings forth the necessity of data retrieval. Text mining as the main part of data mining knowledge deals with this issue. This knowledge organizes a huge collection of documents in order to discover and extract hidden relationship and actionable information. Text mining techniques consist of information recovery of documents, the extraction of relationships, and the classification of documents, which is the focus of these techniques.

Classification systems consist of Supervised, Unsupervised, and Semi-Supervised methods. In supervised classification systems, such as Support Vector Machine (SVM) and Neural Network (NN), documents are classified based on pre-defined categories. However, unsupervised systems, similar to clustering, draw inferences from datasets to cluster the data into different groups. Supervised methods are more costly than unsupervised ones since all samples are labeled. However, containing information about samples, the supervised methods are more functional than their unsupervised peers are. In order to take advantage of both methods, semi-supervised approaches, have been introduced which have a small amount of labeled data with a large amount of unlabeled data. However, the labeled data can be of great practical value.

The aim of this paper is the classification of the documents based on their properties. Semi-supervised clustering is applied since it can perform better with less side-information and without any labeling cost. In fact, clustering is one of the most useful techniques for data mining and with an unsupervised approach, it collects a set of similar data in the same group (cluster) [1].

Most often, the published documents in the workspace are not separable by one linear intelligent system. Therefore, to increase efficiency, the systems are designed based on more than one specific algorithm, which may also raise the complexity of the system and the problem of overfitting. In this study, fuzzy clustering is used to better discriminate the documents while it does not affects the complexity of the system considerably.

The Fuzzy C-Means Clustering (FCM), as one of the approaches of using the fuzzy concept for clustering, assigns weights according to the degree of membership of the samples to each cluster. It defines memberships on the interval [0, 1] and applies the Euclidean distance. Using the Euclidean distance in FCM, the values of all features in the clustering process, are considered the same. However, in the real world, the dimensions have not the same value, which imposes negative effect on the performance of FCM. To solve this problem, this paper considers specific weight for each dimension, which are calculated by the FCM algorithm.

Furthermore, since each text is composed of hundreds or thousands of words, dimensionality reduction stays as an important challenge. In order to deal with this problem and the problem of different dimensions of features for text clustering, this paper has used the Genetic Algorithm (GA) and the weighting TF-IDF approach. These two approaches were not used together in earlier works. Another challenge is the right choice of cluster centers in the process of clustering, since the clustering algorithm efficiency greatly depends on the initial centers. The present study has used the semi-supervised approach as its approach solve this issue.

Overall, the proposed method contains the following steps:

1. **Pre-processing:** including breaking spaces and elimination of redundant words, removing documents with more than one category, calculation of the root words and calculation of the TF.
2. **Feature selection:** which contains a Genetic Algorithm with the TF-IDF method
3. **Clustering:** presenting a novel algorithm as C-W-FCM, including:
   - Creating a seeded set and selecting the number of clusters
   - Creating a partition on the seeded set and selecting the initial centers of each partition.
   - Calculating the weights which are different for each dimension (showing the memberships)

and in particular, the contributions and novelties of this paper include:

- Using Genetic Algorithm and TF-IDF together for feature selection (to address the challenge of dimensionality reduction)
- Improvement of the previous W-FCM algorithm by:
  - Choosing the best initial centers
  - And applying different weights to each dimension

The rest of this paper is organized as follows: Section 2 focuses on the literature review and researches on fuzzy clustering. Section 3 presents the research framework and the details of the proposed method. The findings of the proposed method and its comparison with a typical fuzzy clustering algorithm are presented in Section 4. Finally, Section 5 concludes the paper and offers some strategies for further research.

2. Related Work

Previous methods discussed in this paper are divided into two categories: feature selection and data mining methods. Both of them are discussed in more detail below. At the end of this section the fuzzy clustering method which forms the basis of the proposed method is also described.

2.1. Related work with feature selection

Until 2003, Researchers used the only form of weighting of TF-IDF to select features. This would ignore the importance of words in relation to clusters. In [2], filter-based feature selection methods such as Chi-Square, Information Gain, and Odds Ratio have been combined with the IDF. The results compared with the TF-IDF form indicate the superiority of this method on the Reuters dataset in both micro and macro - F1 measures.

Shang et al. [3] used Gini index to select the features. The results of this approach are compared with that of powerful feature selection methods like Information Gain and Chi-Square; in many cases, the proposed method is much more efficient than those two methods. The best efficiency belongs to the SVM classifier with 0.88 and 0.69 for micro - F1 and macro - F1, respectively.

The authors in [4], unlike the previous approaches, use evolutionary algorithm for dimensionality reduction. In this study, Ant Colony Optimization (ACO) is used for feature selection. A full graph of all words (features) in the dataset is drawn in which each feature is represented as an ant that shows a solution. The evaluation criterion is Mean Squared Error (MSE) and if an ant cannot reduce this criterion in ten sequential steps, that ant is removed and replaced with another one. Finally, there are a number of feature subsets for each ant that are sorted based on classifier performance and feature selection size (a subset is more optimized if it has more efficiency and little feature selection). ACO algorithm procedure continues until reaching a maximum number of iterations.

In [5], the authors offer a combined system of genetic algorithm and clustering for feature selection. In this approach, first, the features are grouped within *k* clusters and next, using a GA, only one feature of each cluster is selected. In genetic algorithm, the length of each chromosome is equal to the number of features. The value 1 on each chromosome means selecting that feature and value of zero means deselecting it. If several words of a cluster on one chromosome are selected, one of the words will be selected randomly. Similarly, if no word of a cluster is selected, one word randomly will be selected. In the GA algorithm, the neural network is used to calculate fitness function of chromosomes. The authors believe that this strategy selects the words with the greatest similarity and increases the accuracy of the classification system as well. According to the

findings of the two datasets named " hapatitise" and "horse", the results show the accuracy of about 0.95 and 0.87, respectively.

In [6], the standard genetic algorithm has been improved in order to have a more optimal set of features. In this study, as in [5], first the features are clustered and next, using a genetic algorithm, the best combination of features of different clusters is selected. In the improved genetic algorithm, each chromosome is composed of two parts; in the first part, each feature belonging to the specific cluster is determined. The second part represents the clusters whose features are selected in the first part. This approach enables the genetic algorithm to optimize the number of the clusters. The comparison of this method with the conventional GA method represents superiority of this strategy.

In [7], a filtering method for feature clustering has been introduced. First, the feature vector matrix (rows represent features and columns represent documents) and TF - IDF of each word are constructed. Features are placed in $\sqrt{d}$clusters (d indicates the number of features) and the best clusters are selected by the genetic algorithm. Since data with common features are grouped in the same cluster, the cluster center represents each cluster. Accordingly, in the genetic algorithm, the length of each chromosome is equal to the number of clusters and the value of each gene determines whether a cluster is selected or not.

In [7], two different classifiers (SVM and KNN) with 10-Fold-Validation are used in order to evaluate the fitness function of each chromosome. The selected chromosome in the last step indicates which features of the cluster should be selected. The results of KNN algorithm usage on the Reuters dataset shows an accuracy of 70%. However, SVM algorithm with an accuracy of 0.65, is less efficient than KNN algorithm.

In recent years, the evolutionary algorithm, especially genetic algorithm are frequently used in combination with clustering in order to introduce a feature selection method. This popularity comes as a result of successful clustering algorithms in text mining and the capability of the genetic algorithm to avoid local optimum. The mentioned methods use data mining algorithm to evaluate feature subsets. However, the wrapper feature selection algorithms (due to the application of data mining algorithm to evaluate datasets), in text mining are very time-consuming. Therefore, the application of evolutionary algorithms for feature selection in this form, except in cases that the dataset is small, is not acceptable due to time limitation. In contrast, these methods increase the efficiency of text clustering algorithm.

Therefore, this paper presents a feature selection method with the genetic algorithm. The specific feature of this study is the application of word variance as a filtering feature selection method to evaluate each feature subset.

2.2. Related work on data mining algorithms

After selecting the optimal features, a suitable data-mining algorithm is chosen to identify documents with the same content. The background of this research and the related works are discussed in the following section.

Songbo Tan in [8], offers an effective strategy to classify texts in *k* categories, Nearest Neighbor. According to [8], KNN algorithm has problem in dealing with the unbalanced dataset as the training data are not dispersed equally in the categories. Therefore, a method named Drug and Pushing has been suggested in order to increase the efficiency of KNN. This method applies training errors such that KNN would be corrected by the Drug and Push. The challenge of this approach is to determine the number of steps in each iteration since the performance of the algorithm is greatly influenced by this issue. However, the complexity of this method is less than the KNN's.

In [9], the concept of fuzzy is used in KNN algorithm to determine the class of each document. In the proposed fuzzy KNN algorithm, instead of assigning each document to a particular class, a degree of membership is determined for each class. The document is assigned to the class which has the highest degree of membership. The degree of membership of each sample is influenced by "inverse distance of the K-nearest-neighbor" and "membership degree of its class". This method uses TF – IDF for feature selection. The findings show that fuzzy KNN algorithm with any number of selected features has better results than the KNN algorithm. Furthermore, SVM algorithm also benefits from this advantage.

The authors in [10] take advantage of information in the test dataset to classify documents. The proposed model consists of two train dataset and test. The documents of the train dataset are labeled whereas the documents of the test dataset are unlabeled. In order to determine the class of document x, first from the train dataset, k documents are selected from the closest documents to the document x. Next, the sum of "average cosine similarity of k documents of each category's sample" with "the degree of document similarity x from the same category's sample " is considered. Finally the document x belongs to the category with the highest value which was obtained in the previous step.

Erdem Alparslan et al. in [11] offered a classification system for categorizing the confidential documents of Turkey in three levels of security. The proposed system combines the SVM classifier and Adaptive Neuro-Fuzzy Inference System (ANFIS). In this approach, the more efficient features are selected by the weighting method TF - IDF. After feature selection, the multi-class SVM algorithm scores all documents. Then, the output of SVM algorithm is considered as the input of ANFIS. Finally, ANFIS determines the value of all documents and it uses the threshold method in order to find security label of each class.

A major challenge in KNN algorithm is to determine the appropriate value of the parameter K that has a great impact on the ranking system performance. In fact, selecting a large value for this parameter can improve the efficiency of the algorithm but the calculations are very costly. Due to the mentioned problems, in [12], KNN algorithm is combined with support vector machine. In [12], first a classifier SVM is determined for each category and in the next step, SVM models designed for different categories, play the role of train data for the KNN algorithm. To determine the category of each document, its Euclidean distance from the support vector of each category is calculated. Finally, each document belongs to the category which has the minimum Euclidean distance from the support vector of that category.

Wen Zhang et al. in [13] suggested a text classification method through a semi-supervised clustering called TESC. TESC method includes a two-step procedure: The first step is

clustering and the second step, prediction is applied to determine the label of the unlabeled texts. The details of TESC process consist of three steps: initialization, clustering and output. At initialization, each text is considered as a candidate cluster and at the beginning, its label is the same as the text label. If the text has no label, the label of the candidate cluster will be "unlabeled". In clustering, two candidate clusters with a minimum Euclidean distance between the pair of candidate clusters, are integrated in a new candidate cluster; or they are known as two unique clusters. The process of clustering is repeated until two candidate clusters remain. In this way, either the clusters are combined or they are known as two separate clusters. Finally, there are some categories in the output which are used to predict the unlabeled texts. Here, in order to avoid dead clusters, the clusters with less than three members are considered as noise and they are ignored. The advantage of this approach is its low complexity compared with SVM, NB, and BPNN (Back Propagation Neural Networks) through EM strategy (Expectation-Maximization). The disadvantage of this approach is being un-scalable and if the number of texts increases, its performance is affected.

The authors in [14] used the dataset words and side-information of the texts for better text classification. In addition, k-means algorithm and hierarchical clustering are combined in order to increase the efficiency of clustering. In [14], regardless to side-information, texts are clustered and next, the documents are re-clustered through the useful side-information. In the first stage of clustering, all documents are clustered in one group. Then, this cluster is divided in two clusters through k–means algorithm. This process continues until the number of clusters is appropriate. At each stage of cluster division, those clusters will be divided of which the total cosine similarity of their samples is less than other clusters. At the second stage, the process of document clustering along with their side-information is repeated. In [14], the Gini index theory is applied in order to reduce the dimensions and for the feature selection phase. Moreover, the useful side-information of the documents is determined through this theory.

The proposed method of this paper is similar in clustering to the methods [13] and [14] with the novelty that for the Euclidean distance measure, different dimensions have various values. In fact, the similarity is just in using clustering and not how this clustering is done, since [13] uses a bottom-up approach while [14] uses a top-down strategy. In addition, the feature selection which applies a genetic algorithm together with TF-IDF and the application of a semi-supervised clustering method turns it different from the previous studies.

In [28], a fuzzy weighted clustering semi-supervised method has been presented, which is similar in some of its aspects to the proposed method in this paper. Therefore we mention the differences with details as follows:

1. In the proposed method, to detect the centers, partitions are applied on the seeded set, while in [28] every sample which is in the seeded set in itself a center.
2. The initial weights and the weighting matrix are completely different from [28].
3. In [28], the weighting matrix is constructed from the initial membership degree matrix which itself is random. These random values affect the weight matrix, in addition. In the proposed method, the weight matrix is filled with variances of each dimension and its updating strategy is different.
4. The sequence of phases of the proposed method is different from that of [28].
5. The negative effect of the randomness of the initial membership degree matrix in [28] shows itself in large datasets which have overlaps and sample noises sensibly. In fact, in [28], the authors have used a small dataset (Iris) which is a dataset with small noise.
6. The proposed method uses the genetic algorithm which itself acts as a weighting and initial feature selection method.

Overall, the proposed method in this paper has high compatibility with different datasets and the importance of its weight matrix is more sensible in the clustering phase.

Some other recent works have been also presented in [29]-[31]. In [29], the graph concept is used for implementation of the clustering algorithm. The graph is constructed based on the similarity of documents, such that the graph vertices are the documents and the weights of edges are the similarity between these two vertices (documents). In the next phase, the center of clusters are those vertices with higher weights of edges which are connected to them. To assign other documents to each cluster, a threshold value is considered and every edge is connected to a center if its weight is above that threshold. This process is repeated until all documents are assigned in clusters.

In [30], the authors claimed that the initial selection of random centers highly affects the accuracy and speed of clustering. Hence, they presented a clustering algorithm, which calculates the initial centers based on Kullback-Leibler divergence (KL) or relative entropy. In addition, in the clustering process the KL distance is used instead of Euclidean distance. The authors have proved that their method not only decreases the total time of clustering but also the accuracy of clustering.

In [31], the authors help students and graduates to find suitable papers related to their subjects faster and with more accuracy. Their dataset contains abstracts and keywords of the papers. After some pre-processing and calculation of TF-IDF, the documents are classified using k-means, hierarchical clustering and spectral clustering. The study shows that the best results are achieved when hierarchical and spectral clustering are used.

These three recent studies have presented new approached for clustering which are also efficient but since their datasets are not general, a fair judgment is hard for comparison.

Finally, Li and Li used a genetic algorithm for text categorization in [32]. In their approach, the genetic algorithm is used to compute the feature weights and the information gain method is used to reduce the dimensions. Lastly, classification is performed using weighted Cosine distance method. Several differences exist between this work and the proposed method of this paper, especially in the GA, which are as follows:

- The fitness functions of chromosomes are different, in [32] clustering is conducted using accuracy measure while in the proposed method a different formula is used. In fact, the method in [32] is wrapper-based while ours is filter-based.
- In [32], a number of random chromosome are generated randomly, while in our method the more the number of words in a document the more chromosomes are generated from it.
- The definitions of chromosomes are different from each other in these two studies.
- Chromosome lengths are not equal in our proposed method while Li and Li claimed that they used the standard genetic

algorithm. It seems that they considered equal-length chromosomes.

- Finally, the clustering and feature selection methods are different in [32] from ours, which is described in Section 3.

2.3. The Fuzzy C-Means algorithm and its constrained version

FCM algorithm, was introduced by Dunn in 1974 [15]. In 1981, Bezdek offered the parameter *m*, which strengthened the effect of fuzzy membership and improved the FCM algorithm. The purpose of FCM algorithm is to minimize the objective *function* as equation 1 [15]:

$$J(U,V,X)=\sum_{k=1}^{c}\sum_{i=1}^{n}(u_{k,i})^{m}\left\|x_i-v_k\right\|^2 \quad (1)$$

where $X = \{x_1, x_2, ..., x_n\}$ represents the dataset and $V = \{v_1, v_2, ..., v_c\}$ is the set of the cluster centers. The set U represents the membership degree of the samples to cluster centers. In other words, $u_{k,i}$ refers to the membership degree of sample i to cluster k and it must satisfy both:

$$u_{k,i}=[0,1] \quad (2)$$

$$k=1,...,c \qquad i=1,...,n$$

$$\sum_{k=1}^{c}u_{k,i}=1 \quad (3)$$

Parameter m in (1), is used to control the degree of the fuzziness membership of each sample. There is no strict rule for choosing the optimal m, but it is usually assumed $m = 2$ [16].

I. Fix the number of clusters, c, where $2\le c\le n$, and initialize the fuzzy partition matrix U with a random value such that it satisfies conditions (2) and (3).

II. Calculate the fuzzy centers v_k using

$$v_k=\frac{\sum_{i=1}^{n}(u_{k,i})^{m}x_i}{\sum_{i=1}^{n}(u_{k,i})^{m}},\forall k=1,...,c \quad (4)$$

III. Update the fuzzy partition matrix U with

$$u_{k,i}=\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{k,i}}{d_{\bar{k},i}}\right)^{\frac{1}{m-1}}} \quad (5)$$

where

$$d_{k,i}=\left\|x_i-v_k\right\|, i=1,...,n \text{ and } k,\bar{k}=1,...,c$$

IV. Repeat step (II) to (III) until one of the termination criterion is satisfied.

Figure 1. FCM algorithm [16]

$\left\|x_i-v_k\right\|$ refers to the Euclidean distance sample i from the cluster center k. Figure 1 illustrates the FCM algorithm.

FCM algorithm continues until a termination condition occurs. One mostly used termination conditions is that the difference between the values of objective function in (1), in two successive iterations is less than a predefined threshold. Another termination condition is reaching the maximum number of iterations.

Since the FCM algorithm is dependent to the centers of the clusters, it may stop in the local minimum influenced by the initial centers. Therefore, the algorithm was developed further and the Constrained FCM was offered in [17]. This algorithm is based on a set of labeled samples, namely the *seeded set*. The procedure is as follows: once the partitioning of the set is done, the centers are calculated, which are considered as initial centers for each cluster. Especially, in the seeded set, there should be at least one sample from each cluster. This approach is used in the proposed method of this paper.

3. The Proposed Method

The proposed method consists of three main phases as pre-processing, feature selection and clustering, as shown in Figure 2. After the documents are prepared through the pre-processing phase, the features are selected for text clustering. This is done through the genetic algorithm with a fitness function, which selects the chromosome with the highest variance. In fact, at the end of this stage, a feature vector is obtained for which the variance is maximum and it is called the feature vector. Next, for each document, TF-IDF of every word of the feature vector is calculated to increase partitioning features. Finally, the feature vectors are input to the clustering phase, which uses the proposed C-W-FCM algorithm.

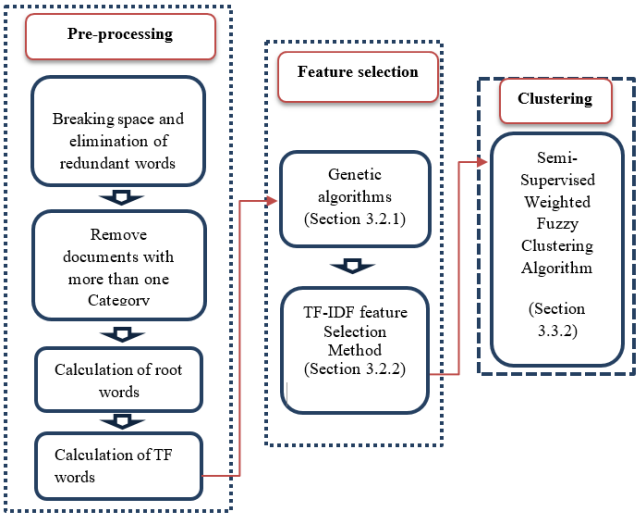

Figure 2. Steps of the proposed method

3.1. Pre-processing

Data mining and clustering algorithms cannot be applied to documents, which have raw materials as the texts contain symbols, letters and redundant words. In addition, clustering systems require a vector for each document to cluster it. The redundant words in each feature vector of the documents are not useful and they have negative effects on clustering system

performance. Therefore, prior to the clustering phase, text pre-processing should be done. In this study, a model is offered in order to cluster texts through CCSI preprocessing [18] to prepare the samples, which has the following steps:

- Breaking Space and Elimination of Redundant Words

The most common and easiest method to show a dataset word by word is through *breaking* space. In this method, the dimensional problem space is equivalent to the number of words. Splitting words in this way brings the punctuation marks such as points, comma, semicolon, question mark, etc. in the feature vectors. That is why at this stage of text pre-processing, the punctuation marks are removed.

The second step of pre-processing is the elimination of redundant words. These repeated words which exist in all texts, do not contain useful information. In addition, they do not affect data category recognition and should be removed from dataset. By redundant words, we mean words like from, in, etc., which have no effect in the classification phase.

There are other words which are not redundant ones but have low impact on the classification. For instance, consider the word "sport", which is not a redundant word. This is a repeated word in sport-related documents and therefore it cannot be a suitable word for document classification (the frequency of the word is not enough (TF)). This word in not removed, but it is not used as a feature for classification of sport-related documents. In fact, the inverse frequency must be considered in addition, which is the number of documents, which include this word (IDF). Hence, TF-IDF is applied in section 3.2.2. Also note that in documents not related to sport, the word "sport" is important but not for documents with the subject of sport.

- Removing Documents with more than one Category

Usually, there are documents that belong to several categories, or it is rarely seen that some documents do not belong to any category. These samples are called outliers and when they are removed, the dataset quality is improved. This model is called modApte Split [19]. In fact, when a sample belongs to more than one class, it causes overlap between classes, which results in bad separation of classes in the training phase. Therefore, these samples are removed in the pre-processing phase of nearly all data mining works. On the other hand, for the test phase it is possible to give these samples to the proposed system, and since our method uses a fuzzy approach it can assign multiple labels to these samples with different ratios.

- Calculating the Root of the Words

Stemming is the process of reducing inflected words to their word stem or root form. Stemming is one of the most important issues in natural language process and it is widely used in information retrieval systems, machine translation, text classification, text summarizing, indexing, and text mining.

Stemming is used in order to categorize the words with the same root in one group. For example, words such as Exit, extract and explosion have common roots. There are many algorithms for English word stemming and the most common one is the Porter algorithm [20]. Results have shown the optimal performance of this root finder as in [21] and therefore, this method is used in this paper.

- Calculating the TF of Words

After pre-processing the dataset and word identification process, a matrix is created. In this matrix, there is a row for each document and a column for the number of words. Next, the number of words frequency should be calculated for each document. Finally, there is a vector for each document and it is initialized with the TF of words.

3.2. Feature selection

Feature selection is the most important step in all data mining algorithms. In the previous section, it is mentioned that the number of initial features are equal to the sum of words of dataset. It is evident that the number of initial features is huge since a text is typically composed of 10,000 to 100,000 words. Most of these features are unnecessary for document classification and they significantly reduce efficiency.

According to the literature review mentioned in the previous section, the application of evolutionary algorithms is useful in order to avoid the local minimum. Therefore, this study uses genetic algorithm, which benefits from a filter-based feature selection method to benefit for the evaluation function.

It is important to note that the application of a weighted TF means that a feature selection phase has been performed before execution of the genetic algorithm.

3.2.1 Feature Selection Based on the Genetic Algorithm

In [22], a feature selection method based on genetic algorithms is presented which is more successful than the IDF (Inverse Document Frequency) and TV (Term Variance) feature selection methods. This paper also applies this genetic algorithm for feature selection. Accordingly, those feature vectors, which maximize the variance, are selected since these features are more important. This does not mean that the feature vector only contains words with high variance and the words that have very little variance may also be selected. In fact, a set of features, all together, make the variance maximum, not individually. Therefore, in this set there may also exist words with lower variances. The reason is stated later in figure 5 and the description above it for a special word "draft", a "vectored feature" which maximizes the variance. Note that a "vectored feature" is a sample of features, which has more than one word, but on the other hand the "feature vector" is the output of the feature selection phase. In data mining, these two concepts are different from each other.

Another clarification is as follows: when using evolutionary algorithms, to evaluate each chromosome we must use one of the machine learning (SVM, KNN, …) or filtering methods. Since there is a large number of features, the machine learning algorithms are not applicable as they are time consuming. Therefore, we turn into filtering methods. One of the filtering methods is the variance, and hence the fitness function of the genetic algorithm is the variance. Since each chromosome is a set of words (or features), therefore we select the chromosome with the highest variance, which means the selected chromosome has the most variations. Henceforth, these chromosomes are more important. For example, suppose chromosome 10 has features 2, 10 and 15, and it is selected by the fitness function. Therefore, if features 2, 10 and 15 are selected from the dataset, the documents are better separated.

The details of the genetic algorithm are described in the following.

- *Search Space*

Each individual word in the dataset is considered a dimension in the search space. For example, suppose there is a set with one hundred documents, each document is composed of ten terms; therefore, the search space has one thousand dimensions in total.

- *Chromosomes representation*

Each chromosome is considered as a vectored feature. Since the search space is so huge, the binary code is not used for chromosomes as it increases the search space and disperses the data, as well. Therefore, in each chromosome one term is determined to display one chromosome-one gene.

- *Initial population*

The evolution process of genetic algorithm starts with an initial population of chromosomes, and each chromosome represents an initial solution to the problem. The number of chromosomes of each document is determined by (6). According to (6), the more the number of words, the more chromosomes will be in the initial population:

$$noc_j = \left\lceil \left({}^{l_j}\!/\!_{L} \right) \times P \right\rceil \tag{6}$$

where, l_j refers to the length of the document *j*-th (number of total words in document *j*), *L* refers to the total number of words in dataset and *P* refers to the initial population size. A random length is considered for each chromosome and the elements of each chromosome is measured based on the words of the document. According to (6) the number of chromosomes made from a document is determined and next a random number between [1, …, number of words of that document] is selected, which is the length of that chromosome. Next, this chromosome is filled randomly with the words of that document. If the number of words is small, it is possible that some of the chromosomes be the same. In this case, the same chromosomes are deleted since the fitness function is based on the variance of words.
For example, if document A contains the words "Oil", "Barrel" and "Opec", the number of chromosomes that can be created on this document is calculated as Table (1).

- *Fitness function of chromosomes*

The fitness function of chromosome *i* and the feature vector variance of *i* are defined in (7) and (8), respectively.

$$fitness(ch_i) = mtv(ch_i, th) \times \ln(length(ch_i) + 1) \tag{7}$$

$$mtv(\vec{t}_i, th) = \sum_{j=1}^{N} \left[vf_{ij,th} - \overline{vf_{i,th}} \right]^2 \tag{8}$$

Where ch_i refers to the chromosome *i*, *th* refers to the threshold, $\vec{t}_i$ refers to the feature vector i, $vf_{ij,th}$ is the words frequency of the feature vector *i* in the document *j*, when the threshold *th* occurs, $\overline{vf_{i,th}}$ is the mean frequency in the sum of document words with threshold *th*.

Table 1. Example of constructing a chromosome

Words in Document A	**Oil**	**Barrel**	**Opec**
Number of Words Frequency	1	2	1
l = Number of Words in document A	$1+2+1=4$		
Number of distinct chromosomes in document A with distinct words	7 (by considering all combinations of distinct words: 3 chromosomes with length 1, 3 chromosomes with len 2, and 1 chromosome with len 3, 3+3+1 = 7 or by considering (6) as: (100*100/1433)=6.97=7 where the first 100 is the average number of words in each document and the second 100 is the initial population).		
Total number of chromosomes produced by document A	$P=300$ $L=100$ $\left\lvert {}^{4}\!/\!_{100} \times 300 \right\rvert = 12$		
Some Samples of Chromosomes	Random number selection between 1 to 3 (the number of distinct words in document A) Chromosome 1: " Opec" Chromosome 2: " Opec" ، "Barrel" — — Chromosome 12: " Barrel" ، "Barrel"،"Opec"		

Here, the size of threshold determines whether the frequency of words are taken into account or not. For example, if the threshold value is four and if at least four words belong to the chromosome *i* in document *j*, $vf_{ij,th}$ is measured based on the formula (9); otherwise it will be zero.

$$vf_{ij,th} = \sum_{k=1}^{m} f_{kj} \tag{9}$$

Here, f_{kj} refers to the frequency of the term k on chromosome *i* in document *j* and *m* is the number of words on chromosome *i*. When the chromosome length is one, one unit is added to the word length in order to avoid zero values in (7). This is essential to prevent the removal of the words, which have length *1* and a high variance.

- *Selection*

In this research, roulette wheel is used in genetic algorithm for selecting chromosomes of the current generation and adding them to the next generation.

- *Crossover*

Crossover is the most important operation of genetic algorithm that combines two chromosomes of the current generation to produce a better new chromosome. This study has used the single-point Crossover. First, the chromosomes are selected based on the roulette wheel and next two points are randomly chosen in two chromosomes to join, as shown in Fig. 3.

- *Mutation*

The role of Mutation, unlike the Crossover process, is the exploration of search space. In other words, this process leads to the exploration of unknown regions in the search space. This study has used the traditional mutation operation [23]. As shown in Fig. 4, in the mutation operation, one word is randomly selected from feature vector and replaced with another word from dataset.

As it is described, considering the TF for each document, features that are more efficient are obtained using the GA algorithm. Figure (2) shows that after feature selection through genetic algorithm, the vector display (vector monitor) of dataset is changed into TF-IDF weighting model. In other words, besides TF, inverse document frequency (IDF) is important.

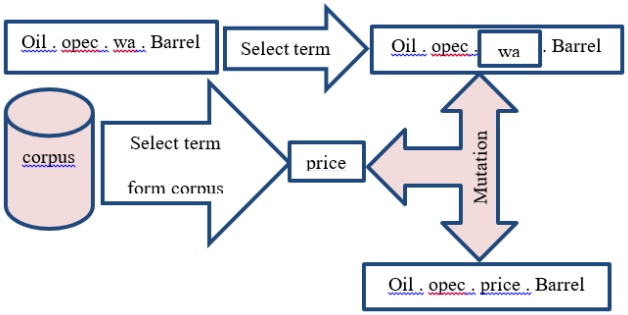

Figure 3. Crossover Operation [22]

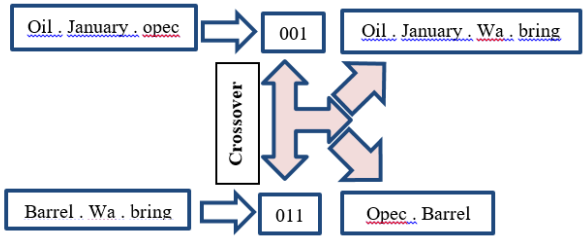

Figure 4. Mutation Operation [22]

3.2.2 Feature Selection based on TF-IDF

The SMART information retrieval system offers a marking model to a variety of TF-IDF weightings in the vector space model [19]. One of the most functional methods of weighting SMART is LTC that is used in this study [19]. LTC function, assuming the total number of words M, measures the weight of term i in the document j from (10).

$$w_{ij} = \frac{tfidf\left(t_i, d_j\right)}{\sqrt{\sum_{s=1}^{|M|}\left(tfidf\left(t_s, d_j\right)\right)^2}} \tag{10}$$

Where the numerator is obtained through (11) (and the denominator is cosine normalizer).

$$tfidf\left(t_i, d_j\right) = \left(1+\log\left(tf\left(t_i, d_j\right)\right)\right)\times \log\frac{N}{df_i} \tag{11}$$

such that $tf\left(t_i, d_j\right)$ represents the frequency of word i in the document j, N refers to the total number of documents in dataset and df_i is frequency-inverse document frequency (the number of documents in which the word i is repeated).

According to (10), if a word is repeated in all the documents $\log\frac{N}{df_i} = 0$, therefore W_{ij} will be zero. TF-IDF weighting is applied after feature selection using genetic algorithm. Therefore, both the TF and IDF are effective in text classification. It is clear that TF-IDF application before clustering is the efficient part of the proposed strategy.

3.3 Fuzzy Clustering Algorithm

According to section 2.3, fuzzy clustering algorithm collects samples based on the membership degree of each category. In this algorithm, all features' values and dimensions are identical. This is why the algorithm is influenced by the irrelevant characteristics. Therefore, this study suggests that given each feature value, a special weight should be determined for better document clustering. To achieve this goal, the Weighted FCM algorithm in [24] is used. In addition, the FCM algorithm is influenced by initial centers. Therefore, in the following sub-sections we describe the W-FCM and the proposed C-W-FCM algorithms as a semi-supervised method, which improves the algorithm.

3.3.1 Weighted FCM Algorithm

The set $D = \{X_j\}$ is defined by $j = 1,\cdots,N$ where $X_j = \left(x_{j1}, x_{j2},\cdots, x_{jd}\right)$, N is the number of documents in dataset and d refers to the features obtained through the feature selection algorithm. The objective of W-FCM algorithm is to minimize (12).

$$J(U,V,D) = \sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^{m}\left[d_{ij}^{(w)}\right]^2 \tag{12}$$

Where $U = \left(u_{ij}\right)_{C\times N}$ is the matrix of membership degree and u_{ij} refers to the membership degree of sample j of the cluster i. $V = \left(v_1, v_2,\cdots, v_c\right)^T = \left(v_{iq}\right)_{C\times d}$ represents the cluster centers and parameter m is used to control the degree of fuzziness of each sample. There is no standard for the optimal selection of *m*, but usually *m* = 2 [16]. The membership degree of samples must be determined in accordance with (13) and (14).

$$u_{ij} = [0,1] \tag{13}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \tag{14}$$

Where $d_{ij}^{(w)}$ refers to the weighted Euclidean distance which is obtained by (15).

$$d_{ij}^{(w)} = \left\| diag(w)\times\left(X_j - V_i\right)\right\| \tag{15}$$

$$diag(w)=\begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & w_d \end{pmatrix} \quad (16)$$

$$\sum_{q=1}^{d} w_q = 1 \quad (17)$$

Where $diag(w)$ is a diagonal matrix of $d \times d$ and $W=(w_1, w_2, \cdots, w_d)$; where w_q refers to the weight of dimension *q*. The centers, membership degree and weight matrix are updated, respectively according to:

$$V_i = \sum_{j=1}^{N} {u_{ij}}^m X_j \Big/ \sum_{j=1}^{N} {u_{ij}}^m \quad (18)$$

$$u_{ij} = 1 \Big/ \sum_{k=1}^{c} \left(\left[{d_{ij}}^{(w)} \right] / \left[{d_{kj}}^{(w)} \right] \right)^{1/m-1} \quad (19)$$

$$w_q^t = 1 \Big/ \sum_{l=1}^{d} \left(\frac{\sum_{i=1}^{c}\sum_{j=1}^{N} \left[{u_{ij}}^t \right]^m \left(x_{jq} - v_{iq} \right)^2}{\sum_{i=1}^{c}\sum_{j=1}^{N} \left[{u_{ij}}^t \right]^m \left(x_{jl} - v_{il} \right)^2} \right) \quad (20)$$

The initial value of the weight matrix is determined based on the word variances. In other words, the more the word variance, the more the value of that dimension. It should be noted that the weights must be determined in accordance with (17). Equation (20), adopted from [24], tries to find the ratio between intra- and inter-cluster distances. In fact, without regarding the 1, the numerator shows the intra-cluster distance and the denominator shows the inter-cluster distance. The less this ratio, the better is the classifier. Therefore, with considering the 1, the higher value in (20) the better is the weight, which relates to higher variances. This equation also resembles the V_{FS} criterion, discussed in section 4.
Finally, the steps of the W-FCM algorithm are summarized as follows:

1. Determining the number of clusters, initializing the weight matrix and initial centers and then, measuring the membership degree matrix of the samples against the initial centers through (19).
2. Updating centers through (18).
3. Updating the membership degree and weight matrix through (19) and (20), respectively.
4. Repetition of steps 2 and 3, until the termination conditions occur (when the difference between the values of (12) in two consecutive steps is lower than the predefined threshold, considered here 10^{-7}).

In W-FCM algorithm, one other step is added to FCM algorithm to determine the features' weight in proportion to the importance of each feature. The similarity between W-FCM algorithm and FCM algorithm is that the number of the cluster centers are initialized randomly. Hence, the FCM algorithm problem still remains. However, the advantage of this algorithm is that it does not take time to determine the features' weight because the optimal weights are updated in each step as the clustering proceeds.

3.3.2 The proposed method: Constrained W-FCM Algorithm

The semi-supervised FCM algorithm is regarded as a development of fuzzy clustering in which a few samples of dataset are labeled and each cluster has a labeled representative. The advantages of semi-supervised fuzzy clustering model include: low- cost labeling, specified number of clusters and fast convergence. This algorithm still uses the Euclidean distance to measure the similarity of the samples. Consequently, the problem of equal dimension weights still exists. To solve the problem, the W-FCM algorithm (introduced in the previous section) are joined together into C-W-FCM or Constrained Weighted FCM algorithm.
The process of C-W-FCM algorithm is similar to that of W-FCM and the only difference is that C-W-FCM algorithm applies some labeled samples named the Seeded set. In fact, the seeded set is the set of labeled documents. In fact, C-W-FCM algorithm is expressed based on a seeded set and the basic operation of the algorithm is done by this set.
In the first stage of clustering in W-FCM, instead of initializing the number of the clusters and determining the cluster centers randomly, the initial centers and the number of clusters are determined by labeled samples. This is accomplished by partitioning which is defined as: a partition on the seeded set is dividing it into non-empty subsets such that their union is the seeded set and their intersections are null. Assume the seeded set contains three members (three labeled documents): $\{x_1, x_2, x_3\}$, such that each member is a three dimensional vector (means that each document has three features). Two sample partitions on this set are as:

$P_1 = \{\{x_1\}, \{x_2\}, \{x_3\}\}$
$P_2 = \{\{x_1\}, \{x_2, x_3\}\}$

Next, to obtain the center of each partition, we do as the following: for the first partition, we add all the three subsets $x_1+x_2+x_3$ and divide by 3. The result is 1x3 vector, which shows one choice of the centers.
For the second partition, first we add the second subset, $x_2+x_3=x_{23}$, and divide it by 2, which is also a three dimensional vector. Next, we add x_1+x_{23} and again divide by 2. The result is also a three dimensional vector which shows another choice for centers. By this method of partitioning, we can distribute the centers in the space.
Next, the clustering procedures are followed as that of W-FCM. The procedure of the C-W-FCM algorithm is as follows:

1. Creating a seeded set, as there is at least one representative from each cluster.
2. Selecting the number of clusters (number of partitions) based on the number of distinct categories in the seeded set.
3. Creating a partition on the seeded set and selecting the initial centers of each partition.
4. Initializing the weight matrix and measuring the membership degree matrix of the samples according to (19).
5. Updating centers according to (18).
6. Updating membership degree and weight matrix based on (19) and (20), respectively.

7. Repeating steps 5 and 6 till the establishment of the termination conditions (when the difference between the values in (12) in two consecutive repetitions is lower than the predefined threshold 10^{-7}).

Compared to the algorithm W-FCM, the advantages of C-W-FCM algorithm are: faster convergence, less affection by initial centers and smart operation. The evidence for faster convergence is the addition of the seeded set, which guides the algorithm with a good initial set.

4. Simulation Results and Performance Evaluation

The proposed approach classifies texts through the clustering algorithms and the small set of labeled samples. After clustering, the labels of samples are not clear and only the cluster and the document's category are identifiable. Therefore, several clustering evaluation criteria such as V_{PC}, V_{PE}, V_{MPC} and V_{FS} are applied to assess the proposed approach.

It is worth mentioning that other unsupervised and semi-supervised methods, e.g., [13], have applied the supervised criteria such as Recall and Precision in order to identify each document's label after the clustering process. Therefore, comparison could not be conducted as in this paper, the performance measures which are specified for unsupervised methods are used. In fact, in unsupervised methods, the samples are finally labeled which violates the principle behind using unsupervised methods of not needing labeled data. On the other hand, we could have applied our specific measures for evaluating them, but the problem is the unavailability of their parameters and ambiguities in implementation. This results in inappropriate and unfair comparison, which breaks up the attitude of comparing the results.

In order to evaluate the suggested method, we compare C-W-FCM results with two clustering algorithms (FCM and W-FCM) since C-W-FCM is an improved form of these two clustering algorithms.

The Reuters dataset **[25]** as the most used evaluation data set has been applied to evaluate the performance of the proposed method. In this paper, similar to previous studies such as [26], the documents are classified into five categories: crude (374 documents), interest (272 documents), trade (327 documents), money-fx (309 documents) and money-supply (151 documents). These documents contain 4026 distinct words.

In order to measure the superiority of the proposed approach over the two other clustering approaches, four criteria V_{PC}, V_{PE}, V_{MPC} and V_{FS} are applied as defined in Table (2). Three criteria V_{PC}, V_{PE} and V_{MPC} depend on the membership degree and they change based on the membership degree given by the clusters and therefore, it may cause failure and inefficiency. However, the fourth index, V_{FS} takes into account both membership degree and dataset structure. In this index, the first and second parts are density (the objective function in W-FCM) and separation criteria, respectively. The focus of this index is the clusters in which the intra-cluster distance is low and the inter-cluster distance is high. In fact, the main goals are [27]:

- to minimize the index so that the data density within clusters will be high
- and to separate clusters as much as possible.

Dataset in the form of a matrix with dimensions of 4026 * 1433 is considered as the input of genetic algorithm [22], which has the initial values of the parameters in Table (3).

Table2. Evaluation Criteria

Evaluation functions	Defined Functions	Description
V_{PC}	$\frac{1}{n}\sum_{k=1}^{c}\sum_{i=1}^{n}(u_{k,i})^2$	$\frac{1}{c} \le V_{PC} \le 1$ Max(VPC)
V_{PE}	$-\frac{1}{n}\sum_{k=1}^{c}\sum_{i=1}^{n}(u_{k,i}\log_2 u_{k,i})$	$0 \le V_{PE} \le \log_2 c$ Min(V_{PE})
V_{MPC}	$1-\frac{c}{c-1}(1-V_{PC})$	$0 \le V_{MPC} \le 1$ Max(V_{MPC})
V_{FS}	$\sum_{k=1}^{c}\sum_{i=1}^{n}u_{k,i}^{m}\|x_i - v_k\|^2 - \sum_{k=1}^{c}\sum_{i=1}^{n}u_{k,i}^{m}\|v_k - \bar{v}\|^2$	——— Min(VFS) $\bar{v} = \frac{1}{c}\sum_{k=1}^{c}v_k$

Table. GA Parameters [22]

Parameter	Value
Threshold (th)	0.70
Minimum length vector features	1
Maximum length vector features	5
initial population	500
Crossover rate	0.80
Mutation rate	0.10
Iteration	300

Similar to [22], the performance of the clustering system for all features is obtained for 0.01, 0.03, 0.05 and 0.10 percent of the features.

As described in subsection 3.2.1, the proposed feature selection approach through genetic algorithm may aslo result in selection of the words with a low variance (since the chromosomes are vectored features). For example, the 5^{th} feature with a very high variance is located next to the terms with low variance such as features 25, 30, etc. which have about zero variances (Figure 5). Combining terms with high and low variances makes a better distinction among categories since the meaning of a word may change in different documents. For example, the word Draft means document, war, bill, etc. Although the word Draft may have high variance, the multiplicity of meaning might be misleading in the process of clustering. Therefore, the purpose of this step is to find the words with low variance like document and war and when they are located next to Draft in the feature vector, the documents that have *Draft* which means *document*, will be distinguished easily from the documents in which *Draft* means *war*.

4.1 FCM Result

In this part, FCM algorithm is applied for different percentages of features. The results of this algorithm, for the number of clusters: two, three, four and five and for percentages of all features are presented in figures 6 to 9.

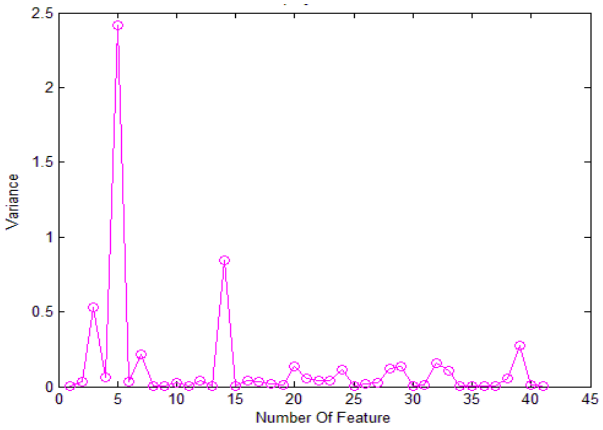

Figure 5. The variance of feature vector selection with genetic algorithm and 0.01 of features

As can be seen, FCM algorithm in all four criteria for 0.01 of all features gets a better result. Since the documents of dataset are classified in five groups, it is expected that for all four criteria the best result will be for the number of clusters being five. Unlike expectation, PC and PE standards have better results when the number of clusters is fewer. However, MPC and FS criteria that are more accurate, have results that are more reasonable. Thus, FS is regarded as a reliable criterion for the evaluation of clusters. It can be concluded from figures 6-9 that:

- The best outcome for FCM is obtained for 0.01 of all features.
- MPC and FS criteria, and especially FS, are more accurate and reliable than PC and PE.

4.2 W-FCM Result

In this section, the W-FCM algorithm is applied for different percentages of features. The results of this algorithm, for the number of clusters: two, three, four and five and for percentages of all features are presented in figures 10 to 13.

As seen from these figures, W-FCM algorithm also gets the best efficiency for 0.01 of features in all four criteria and for 4 or 5 clusters. If there are 2 or 3 clusters, except FS, the other three criteria get better outcomes only for 0.03 of features. The reasons are:

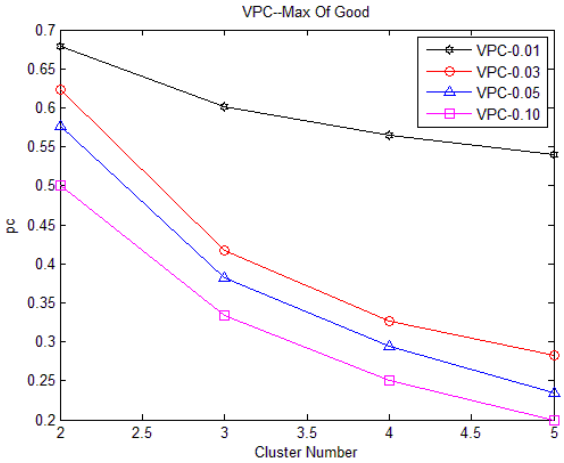

Figure 6. evaluation of V_{PC} criterion for FCM approach

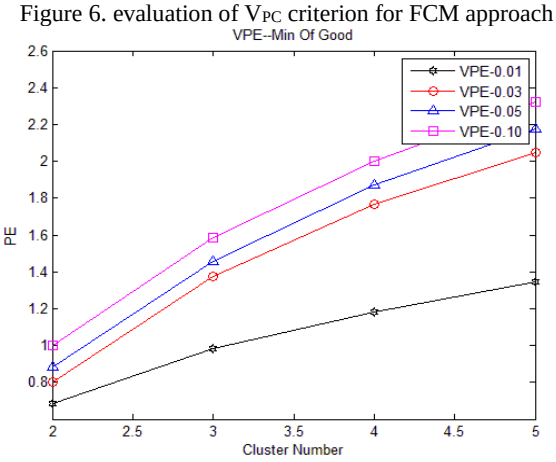

Figure 7. evaluation of V_{PE} criterion for FCM approach

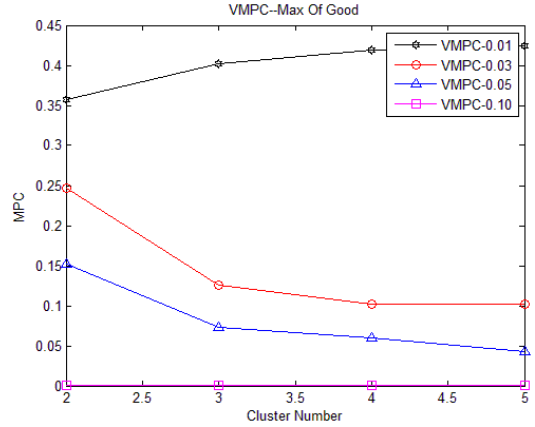

Figure 8. evaluation of V_{MPC} criterion for FCM approach

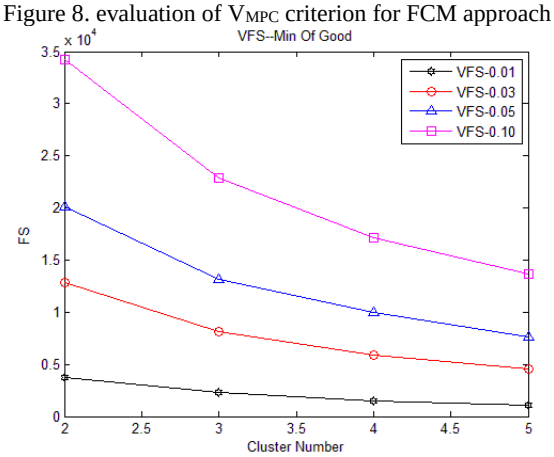

Figure 9. evaluation of V_{FS} criterion for FCM approach

The documents are classified in five groups and when the number of clusters are fewer, the system will be more complex in order to improve performance. A factor that improves the system performance is increasing the number of features. Hence, the more the number of clusters, the more simple the system and if the number of features are less, the result will be better. It is important to note that increasing the number of features is limited and beyond the determined threshold, overfitting happens in the system; in the opposite situation when the number of features is few, underfitting happens.
About FS criterion with two clusters, it is expected that the result will be the same as the three previous criteria. However, if there are two clusters, decreasing the number of features leads to underfitting (Fig. 13). When the number of features is increased, the common features of two clusters also increases. As a result, the distance between two clusters decreases and this criterion increases.

The results of different percentages of features in four or five clusters are compared by the FS criterion and the following conclusions are inferred:

- The best result for W-FCM is obtained for 0.01 of the features.
- This cluster could not split the documents of two similar groups (money - fx and money - supply), because the best outcome is achieved for the one which has four clusters but the documents logically belong to the fifth group.

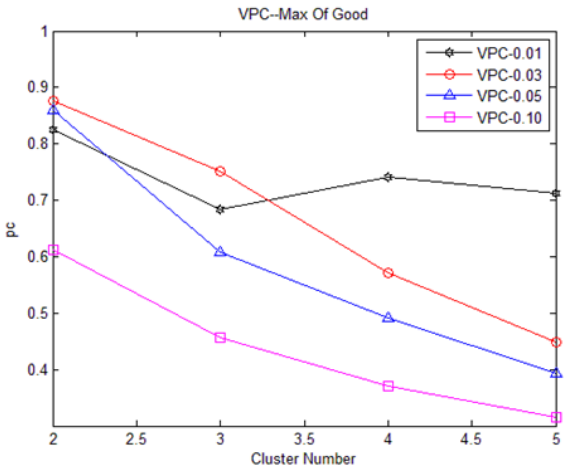

Figure 10. evaluation of V_{PC} criterion for W-FCM approach

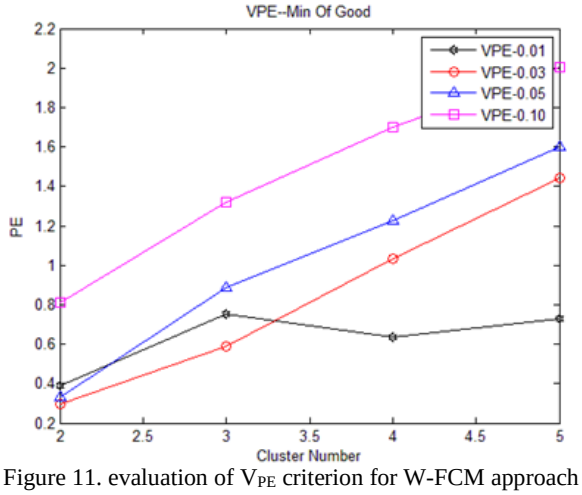

Figure 11. evaluation of V_{PE} criterion for W-FCM approach

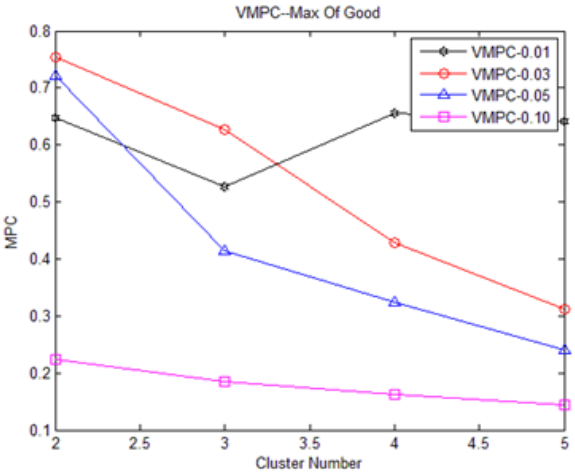

Figure 12. evaluation of V_{MPC} criterion for W-FCM approach

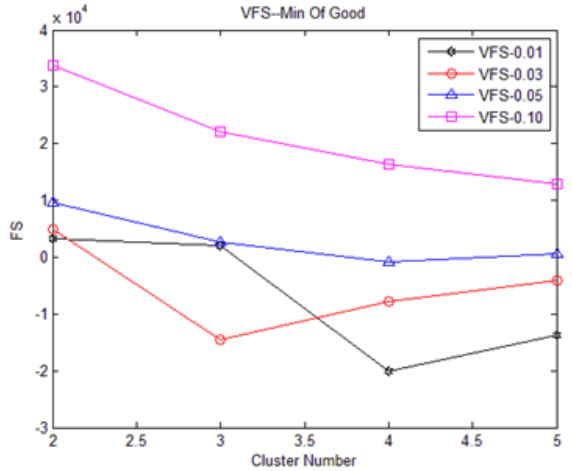

Figure 13. evaluation of V_{FS} criterion for W-FCM approach

4.3 Comparison between FCM and W-FCM Algorithms

Through feature valuation, it is expected that W-FCM has more favorable results than FCM based on clustering and feature numbers. In addition, weighting features is very important in the simulation results. The results of comparison between these two algorithms are presented in Tables 4 to 7.
The results indicate that W-FCM, based on the number of features, has higher performance than FCM. In FCM, during the text classification, every distinct word in dataset is regarded as one feature and therefore, the dimensions of the samples grow up. In this procedure, the importance of feature selection is more evident. Since W-FCM uses the weighted feature vector, it works as a feature selection method. In fact, weighting features is a developed form of the feature selection phase. When a feature is selected in the feature selection process, it has a weight of 1 and if it is zero, it will not be selected. Therefore, it is clear that W-FCM performance is better since it has one more step during the feature selection process.
In addition, dimension weighting makes W-FCM algorithm to reach the results more quickly than FCM. The findings show that W-FCM gets the results through 25 to 35 repetitions. However, FCM gets the results after 51 clustering operations.

4.4 C-W-FCM Result

The results of C-W-FCM algorithm are measured as 0.30 of the samples are labeled, admitting [13]. These results are indicated in Figures 14 through 17 according to four criteria of clustering evaluation with different percentages of features. Accordingly, C-W-FCM algorithm gets the best results for 0.01 of features.

The remarkable point is that, C-W-FCM algorithm for 0.01 of the features, unlike two previous algorithms, with five clusters gets better results. This reflects the success of the proposed method for separation of two similar documents.

Table 4. The results of FCM and W-FCM algorithms with 0.01 of features

	W-FCM				**FCM**			
Validity functions	Number Of Clusters				Number Of Clusters			
	2	3	4	5	2	3	4	5
V_{PC}	0.8240	0.6844	0.7414	0.7134	0.6783	0.6009	0.5646	0.5395
V_{PE}	0.3902	0.7535	0.6332	0.7317	0.6865	0.9852	1.1815	1.3461
V_{MPC}	0.6481	0.5265	0.6553	0.6417	0.3567	0.4014	0.4195	0.4244
V_{FS}	+ 3.2686e+03	+ 1.9539e+03	- 2.0044e+04	- 1.3672e+04	+ 3.7819e+03	+ 2.3061e+03	+ 1.5075e+03	+ 1.0806e+03

Table 5. The results of FCM and W-FCM algorithms with 0.03 of features

	W-FCM				**FCM**			
Validity functions	Number Of Clusters				Number Of Clusters			
	2	3	4	5	2	3	4	5
V_{PC}	0.8767	0.7512	0.5707	0.4491	0.6233	0.4169	0.3269	0.2820
V_{PE}	0.2982	0.5863	1.0347	1.4404	0.8010	1.3723	1.7652	2.0496
V_{MPC}	0.7534	0.6267	0.4275	0.3114	0.2467	0.1253	0.1026	0.1025
V_{FS}	+ 4.8631e+03	- 1.4617e+04	- 7.7404e+03	- 4.1216e+03	+ 1.2867e+04	+ 8.1578e+03	+ 5.9138e+03	+ 4.5697e+03

41 features (0.01 of features) are appropriate for training the model and the rest is not confirmed. When the seeded set is identified, the number of appropriate clusters is five and the number of suitable features to cluster documents in Reuters dataset is 41 (0.01 of the features).

Table 8 presents the results of the proposed algorithm when the number of clusters is five for different percentages of the feature vectors.

Table 6. The results of FCM and W-FCM algorithms with 0.05 of features

	W-FCM				**FCM**			
Validity functions	Number Of Clusters				Number Of Clusters			
	2	3	4	5	2	3	4	5
V_{PC}	0.8608	0.6090	0.4923	0.3925	0.5761	0.3820	0.2948	0.2347
V_{PE}	0.3318	0.8845	1.2249	1.6015	0.8816	1.4570	1.8723	2.1776
V_{MPC}	0.7216	0.4135	0.3231	0.2406	0.1522	0.0730	0.0597	0.0434
V_{FS}	+ 9.5644e+03	+ 2.5757e+03	- 762.3606	+ 602.4494	+ 2.0078e+04	+ 1.3123e+04	+ 9.9886e+03	+ 7.5916e+03

Table 7. The results of FCM and W-FCM algorithms with 0.10 of features

	W-FCM				**FCM**			
Validity functions	Number Of Clusters				Number Of Clusters			
	2	3	4	5	2	3	4	5
V_{PC}	0.6122	0.4567	0.3715	0.3158	0.5000	0.3333	0.2500	0.2000
V_{PE}	0.8092	1.3213	1.6992	2.0017	1.0000	1.5850	2.0000	2.3219
V_{MPC}	0.2243	0.1850	0.1619	0.1448	2.2959e-11	1.2483e-11	1.1168e-11	1.3256e-11
V_{FS}	+ 3.3729e+04	+ 2.2119e+04	+ 1.6350e+04	+ 1.2923e+04	+ 3.4281e+04	+ 2.2854e+04	+ 1.7141e+04	+ 1.3712e+04

In C-W-FCM method, V_{FS} criterion gets very small which indicates that the proposed algorithm uses the labeled samples to separate different classes efficiently. The sharp decay in the mentioned criterion is due to the increased separation range of

documents in money-fx and money-supply categories, which overlap due to their high level of similarity.

According to the previous findings and admitting [22], clustering algorithm with 0.01 of terms in the Reuters dataset has achieved the best results. In this section, the results of the proposed C-W-FCM algorithm, W-FCM, and FCM for 0.01 of the initial features are compared together. The results of these algorithms, when the number of clusters is five, are presented in Table 9. As can be seen, the proposed algorithm has the best performance in all four criteria since it takes advantage of W-FCM algorithm and the semi-supervised model. This approach splits the documents of two similar categories (money - fx and money - supply) while W-FCM algorithm failed to split the documents of these categories. The findings in tables (4) through (7) indicate when the number of clusters is four rather than five, W-FCM has higher performance.

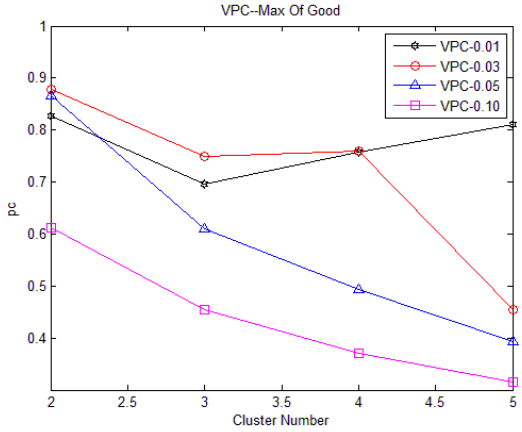

Figure 14. evaluation of V_{PC} criterion for C-W-FCM approach

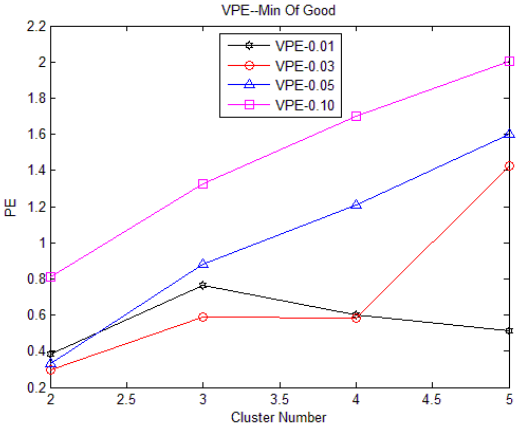

Figure 15. evaluation of V_{PE} criterion for C-W-FCM approach

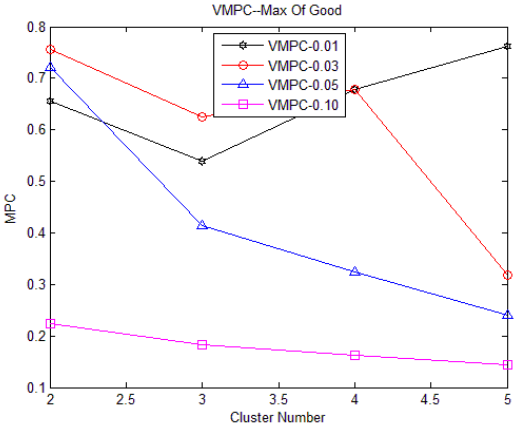

Figure 16. evaluation of V_{MPC} criterion for C-W-FCM approach

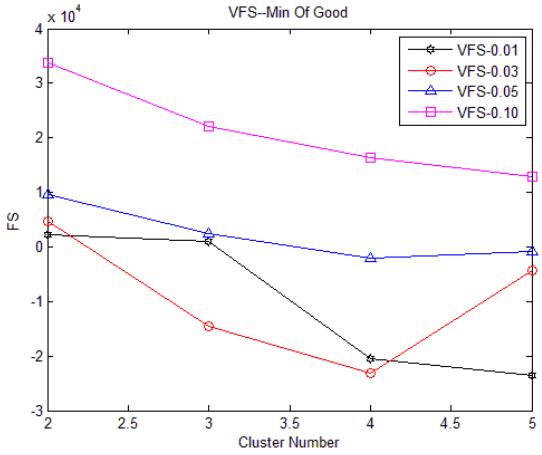

Figure 17. evaluation of V_{FS} criterion for C-W-FCM approach

Table 8. The results of C-W-FCM for different percentages of features in the 0.30 ratio of the data set

C-W-FCM Algorithm	0.01 percent of Features	0.03 percent of Features	0.05 percent of Features	0.10 percent of Features
V_{PC}	0.8099	0.4480	0.4226	0.3150
V_{PE}	0.5155	1.4432	1.5015	2.0021
V_{MPC}	0.7623	0.3100	0.2783	0.1437
V_{FS}	$-2.3531e^{+4}$	$-4.0931e^{+3}$	$-2.3560e^{+3}$	$+1.2933e^{+4}$

Generally, this was because W-FCM assumes the documents belong to four, and not five, categories. Henceforth, W-FCM fails to split the documents of two similar categories.

Another description is as follows: we expected both W-FCM and C-W-FCM to have the same best performance on all the measures when the number of clusters is five. However, for W-FCM the best result occurs when the number of clusters is four with 1% of the features (figures 10-13). Which is when the documents from two clusters with similar subjects are classified in one cluster. Nevertheless, for C-W-FCM the best result occurs when the number of clusters is five with 1% of the features (figures 14-17).

Moreover, according to Table 9, the most important clustering measure, VFS has a high reduction. Therefore, we can conclude that C-W-FCM is able to discriminate the two subjects: money-fx and money-supply (which have similar topics). This also results in higher inter-cluster distance compared to W-FCM.

5. Conclusion and Future Work

In this paper, we address of the main challenges of the document classification as the high dimensionality of the dataset. In a very simple case, the initial set of features is defined based on the total number of words existing in the text set.

Table 9. Comparison of C-W-FCM and the other two clustering algorithms

	V_{PC}	V_{PE}	V_{MPC}	V_{FS}
FCM	0.5395	1.3461	0.4244	+1.0806e+03
W-FCM	0.7134	0.7317	0.6417	-1.3672e+04
C-W-FCM 0.30 Seeded	**0.8099**	**0.5155**	**0.7623**	**$-2.3531e^{+4}$**

Since each news text is composed of many words, the dimensions increases sharply. Therefore, classification systems must identify unnecessary features and remove them from dataset in order to predict issues accurately.

This study has reviewed the existing feature selection methods and suggested an optimal feature selection method, which is developed from the evolutionary approaches and filtering feature selection. It suggests an optimal strategy for feature selection through the genetic algorithm and evaluation function based on the words variance. The genetic algorithm is applied to avoid the local minimum and achieves a desired result through the mutation step. The application of variance for weighting the features subset solves the problem of wrapped feature selection algorithms and increases speed and efficiency.

In addition, in this study, the clustering algorithm is applied to select an appropriate cluster. Since FCM algorithm has defects such as depending on initial centers, employing Euclidean distance, and identical treatment with features, we proposed a developed FCM in this study, namely C-W-FCM. The proposed method benefits from weighting the features, determining the number of clusters automatically, and identifying centers intelligently. These strategies increase the efficiency and speed of convergence in the proposed method compared to the conventional clustering for texts classification.

The findings indicate noticeable efficiency of the proposed C-W-FCM algorithm, from about 27 to 33% compared to FCM algorithm and about 9 to 12% compared to W-FCM algorithm. It benefits from a weighting method as a feature selection approach. Weighting features doesn't need the re-execution time for measurement due to the fact that the weight matrix is updated in each step. On the other hand, it must be taken into account that the automatic updating requires labeling some of the samples.

Although the proposed method has acceptable performance for text classification, some recommendations are suggested for further research on better performance and better results. The suggestions are:

- Another development of clustering algorithm has been proposed in which the Minkowski criterion is employed. The experiments indicate the superiority of this method over Euclidean distance. Therefore, it is recommended that this measure would be used for C-W-FCM clustering algorithm.
- A series of samples must be selected randomly to create a seeded set; future research is needed to apply evolutionary algorithms to determine appropriate candidates for the labeling process.

Acknowledgements

The authors would like to thank the High Performance Computing Center of Shahid Chamran University of Ahwaz (SCU-HPCC) for providing computational resources.

References

[1] J. Ghosh and A. Liu, The Top Ten Algorithms in Data Mining, V. Kumar and X.Wu, Eds. Boca Raton, New York: CRC Press, ch. 2, p. 22, 2009.

[2] F. Debole and F. Sebastiani, "Supervised Term Weighting for Automated Text Categorization," Studies in Fuzziness and Soft Computing, vol. 138, pp. 784-788, 2003.

[3] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," Expert Systems with Applications, vol. 33, no. 1, pp. 1-5, 2007.

[4] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," Expert Systems with Applications, vol. 36, no. 3, pp. 6843-6853, 2009.

[5] M. Rostami and P. Moradi, "A Clustering Based Genetic Algorithm for Feature," Proc, 6th Conference on Information and Knowledge Technology, Tehran, pp. 112-116, 2014.

[6] T. P. Hong, C. H. Chen, and F. S. Lin, "Using group genetic algorithm to improve performance of attribute clustering," Applied Soft Computing, vol. 29, pp. 371-378, 2015.

[7] C. H. P. Ferreira, D. M. R. de Medeiros, and F. Santana, "FCFilter: Feature selection based on clustering and genetic algorithms," Proc. IEEE Congress on Evolutionary Computation (CEC), pp. 2106-2113, 2016.

[8] S. Tan, "An effective refinement strategy for KNN text classifier," Expert Systems with Applications, vol. 30, no. 3, pp. 290-298, 2006.

[9] B. Trstenjaka, S. Mikacb, and D. Donkoc, "KNN with TF-IDF Based Framework for Text Categorization," 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, vol. 69, pp. 1356-1364, 2014.

[10] G. Ram´ırez-de-la-Rosa, M. Montes-y-G´omez, and L. Villase˜nor-Pineda, "Enhancing Text Classification by Information Embedded in the Test Set," Computational Linguistics and Intelligent Text Processing, vol. 6008, pp. 627-637, 2010.

[11] E. Alparslan, A. Karahoca, and H. Bahsi, "Classification of confidential documents by using adaptive neurofuzzy inference systems," Procedia Computer Science, vol. 3, pp. 1412-1417, 2012.

[12] C. H. Wana, L. H. Leeb, R. Rajkumarb, and D. Isa, "A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine," Expert Systems with Applications, vol. 39, no. 15, pp. 11880-11888, 2012.

[13] W. Zhang, X. Tang, and T.i Yoshida, "TESC: An approach to Text classification using Semi- supervised Clustering," Knowledge-Based Systems, vol. 75, pp. 152-160, 2015.

[14] R. E. Thomas and S. S. Khan, "Co-Clustering with Side Information for Text Mining," Proc. Data Mining and Advanced Computing (SAPIENCE), pp. 1-4, 2016.

[15] J. C. Dunn, "Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problems," Journal of Cybernetics, vol. 4, no. 2, pp. 1-15,

1974.
[16] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New york, London: Utah State University, 1981.
[17] K. Li, Z. Cao, L. Cao, and R. Zhao, "A novel semi-supervised fuzzy c-means clustering method," Proc. Control and Decision Conference, Chinese, 2009.
[18] R. Baghel and D. R. Dhir, "A Frequent Concepts Based Document Clustering Algorithm," International Journal of Computer, vol. 8, no. 3, pp. 6-12, 2010.
[19] C. D. Manning, P. Raghavan, and H.Schütze, An Introduction To Information Retrieval. Cambridge, England: Cambridge University Press, 2009.
[20] (2016,November) Snowball. [Online]. http://snowball.tartarus.org/algorithms/porter/stemmer.html
[21] M. Craven and S. Slattery, "Relational Learning with Statistical Predicate Invention: Better Models for Hypertext," Machine Learning, vol. 43, no. 1, pp. 97-119, 2001.
[22] P. Shamsinejadbabki and M. Saraee, "A New Unsupervised Feature Selection Method for Text Clustering Based on Genetic Algorithms," Journal of Intelligent Information Systems, vol. 38, no. 3, pp. 669-684, 2012.
[23] D. A. Coley, An introduction to genetic algorithms for scientists and engineers. Hong Kong: World Scientific, 1999.
[24] A. Jahanbakhsh Pourjabari and M. Seyedzadegan, "An improved method of fuzzy c-means clustering by using feature selection and weighting," IJCSNS International Journal of Computer Science and Network Security, vol. 16, no. 10, pp. 64-69, 2016.
[25] D. Lewis. (2016, April) daviddlewis. [Online]. http://www.daviddlewis.com/resources/testcollections/reuters21578
[26] N. Azam and J. T. Yao, "Comparison of term frequency and document frequency based feature selection metrics in text categorization," Expert Systems with Applications, vol. 39, no. 5, pp. 4760-4768, 2012.
[27] K. L. Wua and M. S. Yang, "A cluster validity index for fuzzy clustering," Pattern Recognition Letters, vol. 26, no. 9, pp. 1275-1291, 2005.
[28] M. Nazari and J. Shanbehzadeh, "Improve Semi-Supervised Fuzzy C-Means Clustering Based on Feature Weighting", Global Journal of Science, Engineering and Technology, no. 14, pp. 82-89, 2013.
[29] P. Chanda and A. K. Das, "A Novel Graph Based Clustering Approach to Document Topic Modeling," Proc. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018.
[30] Z. Huan, Z. Pengzhou, and G. Zeyang, "K-means Text Dynamic Clustering Algorithm Based on KL Divergence," in 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 2018.
[31] Y. Zhang and Y. Wan, "How to Find Valuable References? Application of text mining in abstract clustering," Proc. 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, 2017.
[32] Junwei Li and Xiangqian Li, "Feature Weighting Method Based on Real-coded Genetic Algorithm in Text Categorization", Proc. 8th International Symposium on Computational Intelligence and Design (ISCID), 2015.

Soheila Ramezani Pour received her M.Sc. degree in Computer Engineering, major in Artificial Intelligence from Shahid Chamran University of Ahvaz (SCU), Ahvaz, Iran in 2017. Her research interests include artificial intelligence, data mining, reinforcement learning, fuzzy logic and evolutionary methods.
Email: s.ramezani1989@gmail.com

Marjan Naderan received her B.Sc. degree in Computer Engineering in 2004 and the M.Sc. degree in Information Technology in 2006 both from Sharif University of Technology (SUT), Tehran, Iran. She received the Ph.D. degree in Computer Engineering, major in computer networks in Feb. 2012, from Amirkabir University of Technology (AUT), Tehran, Iran. Dr. Naderan joined the Computer Engineering department of Shahid Chamran University (SCU) in Ahvaz, Iran in Sep. 2012. She was the head of the Computer Engineering department from 2013 to 2015. She is currently the director of the HPC Center in Shahid Chamran University of Ahvaz (SCU-HPCC). Her research interests include computer networks, wireless and mobile networks, IoT and cloud computing, social networks, object tracking, network optimization, simulation of network protocols and bio-inspired and intelligent methods in networks.
Email: m.naderan@scu.ac.ir

Saeid-Allah Mortazavi received his B.Sc. and M.Sc. degrees in electrical engineering from the Ferdowsi University, Mashad, Iran in1989 and 1992. He received Ph.D. in electrical engineering from IIT-Delhi in Jan, 1999. He is currently a Professor with the Department of Electrical Engineering, Shahid Chamran University of Ahvaz, Iran, where he has been since 1999. His research interests are soft computing and intelligent control systems.
Email: mortazavi_s@scu.ac.ir

Paper Handling Data:

Submitted: 06.27.2018
Received in revised form: 11.24.2018
Accepted: 03.02.2019
Corresponding author: Dr. Marjan Naderan
Affiliation of the corresponding author: Assistant Professor, Department of Computer Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz