J Supercomput
DOI 10.1007/s11227-017-2046-2

Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering

Laith Mohammad Abualigah[1] · **Ahamad Tajudin Khader**[1]

Abstract The text clustering technique is an appropriate method used to partition a huge amount of text documents into groups. The documents size affects the text clustering by decreasing its performance. Subsequently, text documents contain sparse and uninformative features, which reduce the performance of the underlying text clustering algorithm and increase the computational time. Feature selection is a fundamental unsupervised learning technique used to select a new subset of informative text features to improve the performance of the text clustering and reduce the computational time. This paper proposes a hybrid of particle swarm optimization algorithm with genetic operators for the feature selection problem. The k-means clustering is used to evaluate the effectiveness of the obtained features subsets. The experiments were conducted using eight common text datasets with variant characteristics. The results show that the proposed algorithm hybrid algorithm (H-FSPSOTC) improved the performance of the clustering algorithm by generating a new subset of more informative features. The proposed algorithm is compared with the other comparative algorithms published in the literature. Finally, the feature selection technique encourages the clustering algorithm to obtain accurate clusters.

Keywords Unsupervised text feature selection · Particle swarm optimization · Genetic operators · K-mean text clustering · Hybridization

✉ Laith Mohammad Abualigah
laythdyabat@ymail.com; lmqa15_com072@student.usm.my

[1] School of Computer Sciences, Universiti Sains Malaysia (USM), Gelugor, Pulau Pinang, Malaysia

Published online: 11 April 2017

1 Introduction

In recent years, the increasing amount of digital text information on Internet web pages and modern applications has affected the text analysis process. Text clustering is a suitable technique used to partition a huge set of text documents into a predetermined number of clusters [1,2]. It has been used in many domains in the area of text mining such as text retrieval, text categorization, human movement, fall detection image segmentation [3,4]. The vector space model (VSM) is a common popular model used in the area of the text mining to represent the text features of each document as a vector of terms weights. In this model, each term weight is represented as one-dimensional space [5]. Thus, the performance of the clustering technique is affected by the size of dimension space and uninformative features [6].

Always, text documents contain informative and uninformative features, where uninformative features are noisy, irrelevant, and redundant [7,8]. Unsupervised text feature selection is the main method used to find a new subset of informative features for each document. Classic feature selection techniques fundamentally include document frequency (DF), mutual information (MI), information gain (IG), and Chi-square test (CHI) [9]. These techniques lack mathematical models and are thus difficult to improve [10]. Generally, feature selection technique has two objectives: (1) to maximize the performance of the text clustering algorithm (2) to minimize the number of uninformative features [11].

Meta-heuristic optimization algorithms are applicable for many optimization problems in the past two decades or more. Meta-heuristic algorithms have been successfully used in the area of text mining to solve text feature selection problems [12]. Methods proposed for text feature selection include population-based optimization algorithms, such as harmony search, genetic algorithm, and particle swarm optimization, which have gained increasing research attention. These algorithms aim to obtain improved solutions by applying knowledge from previous iterations. Several meta-heuristic algorithms are utilized to solve feature selection problems; these algorithms include genetic algorithm (GA), harmony search (HS), particle swarm optimization (PSO), cuckoo search (CS), ant colony optimization (ACO), cat swarm optimization (CSO), and artificial bee colony (ABC) [11].

The PSO algorithm was introduced by Kennedy and Eberhart in (1995). It mimics the social behavior of birds flocking and fish schooling and uses the widely known best solution for achieving the optimal solution. This algorithm is successfully used in different domains such as power flow analysis, text clustering, bioinformatics, wireless sensor networks, feature selection, and data clustering [13].

PSO is the common algorithm used to solve complex optimization problems [14]. In this study, the authors applied three models [10]. The first model uses the original PSO algorithm, the second model improves the PSO algorithm by inertia weight to optimize the feature selection model, and the third model includes new functions for the original PSO algorithm. Experimental results showed that the second PSO model is the best one for improving the performance of text feature selection. A new technique that uses PSO with an opposition-based mechanism is applied for text feature selection, and it begins with a set of promising and varied solutions to achieve an optimal solution [11]. The authors investigated the proposed method using three text datasets. The

experimental results show that the effectiveness of the feature selection is increased and the computational time is reduced.

A new GA was proposed to select the optimal subset of text features to improve the text clustering method. It used the term frequency-inverse document frequency (TFIDF) to reduce the document term relationships [15]. Experiments were conducted using spam e-mail documents to validate the feature selection performance. The authors found that the proposed GA for feature selection improves the performance of text clustering. A feature selection technique is suggested using TFIDF to reduce the execution time and to increase the performance of the clustering process [16]. The results show that the proposed method improved the performances of text clustering and text classification methods.

Cat swarm optimization (CSO) algorithm has been proposed to solve several optimization problems. However, CSO is restricted for long execution times. The authors modified this algorithm to improve the text feature selection in text classification [17]. The experiment was conducted using large data. The results show that the proposed modified CSO outperforms the traditional version and the TFIDF with CSO achieved better results in comparison with the TFIDF alone.

HS algorithm selects a subset of text features in discovering the entire dataset while preserving the intrinsic information. A novel feature selection method based on HS algorithm was proposed for selecting a new optimal subset of the informative text features [18]. The algorithm simplicity was used to reduce the computational time and to maximize the number of features. Experimental results show that the proposed modifications of the HS algorithm enhanced the efficacy of the text feature selection technique.

This paper chiefly aims to propose a new feature method and enhance the performance of feature selection methods in obtaining satisfactory results. The specific sub-objectives are as follows:

- To propose a new feature selection method using an optimization technique to find the more informative features in each document. This method is used to improve the performance of the text clustering and reduce its computational time.
- To propose a hybrid particle swarm optimization algorithm with genetic operators (H-FSPSOTC) to improve the performance of the original PSO algorithm and applied to the feature selection problem.
- To utilize the obtained redact subset in K-mean clustering algorithm and determine the optimal document clusters that are similar and relevant.

A wide group of experiments on text datasets is used in the domain of text mining to show the benefits and advantages of the proposed method and its modification in application to feature selection. The proposed algorithm is hybrid PSO with GOs. The proposed method and algorithm are evaluated using eight text benchmark datasets from the Laboratory of Computational Intelligence (LABIC)[1] in terms of MAD, precision, recall, F-measures, and accuracy. The results performed by the proposed method are compared with the results performed by same methods in the literature. H-(FSPSOTC) produced the optimal results. The proposed text feature selection method is impressive

[1] http://sites.labic.icmc.usp.br/text_collections/.

additions in the domain of the text mining for improving the performance of both text feature selection and text document clustering methods. The results show that the proposed method (H-FSPSOTC) not only reduces the number of uninformative features but also significantly enhances performance of the text clustering algorithm; the achieved results are comparatively better than the competitive methods

The rest of this paper is organized as follows: Sect. 2 illustrates the proposed method. Section 3 illustrates the steps of the feature selection problem. Section 4 discusses the proposed hybrid particle swarm optimization algorithm and its procedure. Section 5 shows the text preprocessing steps for text analysis. Section 6 shows the steps of text clustering problem. Section 7 contains the text clustering evaluations measurements. The experimental results of the proposed method are presented in Sect. 8. Finally, the conclusion is provided in Sect. 9.

2 The proposed method

Recently, the text preprocessing steps are used to select optimal informative features in order to reduce the computational time and enhance the performance of text clustering. Hence, this paper proposes an effective method to find accurate clusters by generating a new subset of more informative features. Algorithm 1 shows the methodology of the proposed text clustering method by using the proposed hybrid feature selection algorithm (H-FSPSOTC).

Algorithm 1 The proposed algorithm to find an optimal subset of features

1: **Input**: Set of the text documents.
2: **Output**: Clusters of the text documents.
3: **Algorithm**
4: **First step**
5: Preprocess the text documents with preprocessing steps.
6: Convert text documents into numerical matrix.
7: **Second step**
8: Apply the proposed Hybrid PSO for text feature selection.
9: PSO algorithm
10: Genetic operators
11: Return a new subset of informative text features.
12: **Third step**
13: Apply the k-mean text clustering on selected features.
14: Convert text documents into a numerical matrix.
15: Return the set of clusters.

In the first step, we applied the preprocessing steps to represent the text document in the numerical style and then converted the text documents into a numerical matrix. It is used to facilitate the dealing with the text document through the mathematical equations. In the second step, we proposed hybrid of the PSO for the feature selection problem to eliminate uninformative text features for improving the performance of text clustering. This step is combined PSO algorithm with the operators of the genetic algorithm (genetic operators) to adjust the solution of the PSO. This technique is considered a preprocessing step in pattern recognition, machine learning, and so on.

In the third step, the clustering algorithm (i.e., k-mean algorithm) is used to obtain optimal clusters after obtained a new subset of text features.

3 Text preprocessing

Any text analysis technique, such as text retrieval, text clustering, text feature selection, and so on, needs to convert document contents to become manageable in the underlying algorithm [1,19]. The preprocessing steps are used to convert the document contents in numerical form. These steps are divided in the following:

3.1 Tokenization

Tokenization is the process of splitting a stream of text documents into words or terms, and removing the empty sequence, in which each word or symbol is taken from the first character to the last character, which is called a token [11].

3.2 Stop words removal

The list of common popular words, such as an, this, that, when, be, and other common words that take small weighting, as well as high-frequency and short functional words in the text document clustering, are known as the stop words. These words must be removed from documents because they usually take some part of the document and affect the increasing number of features, thereby leading to the reduced performance of the text clustering technique. The stop word lists are available at http://www.unine.ch/Info/clef/, which consists of 571 words [20].

3.3 Stemming

Stemming[2] transforms the inflectional relevant forms of some words with same root by removing the prefixes and suffixes for each word. For instance, intersect, dissect, and section have the same common root, sect, which is considered a feature. In this paper, we use the Porter stemmer which is the most common stemming method being used [6,21].

3.4 Term weighting (TFIDF)

The term frequency-inverse document frequency (TFIDF) is the common weight scheme used to calculate the term weighting in the area of text mining for the document representation. Each document is represented as a vector of terms weights as Eq. (1):

$$d_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,j}, \ldots, w_{i,t}), \tag{1}$$

[2] Porter stemmer. website at http://tartarus.org/martin/PorterStemmer/.

The term weighting is assigned for each term according to the term frequency in each document and others factors. If the term frequency is high and the same term appears in a few documents, we conclude that this term is useful to distinguish among the documents [22]. The term weighting is calculated by Eq. (2).

$$w_{ij} = tf(i, j) * idf(i, j) = tf(i, j) * \log(n/df(j)), \tag{2}$$

where $w_{i,j}$ represents the weight of term j in document i, and $tf(i, j)$ represents frequencies of term j in a document i. $idf(i, j)$ is a factor used to improve the term which has low frequency and appears in a few documents as Eq. (3), $idf(i, j) = \log(m/df(j))$, where n is the number of all documents in the dataset, and $df(j)$ is the number of documents which contains the term j. The following expressions represent the documents in a common standard format using the vector space model:

$$\text{VSM} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,(t-1)} & w_{1,t} \\ \vdots & \ddots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots \\ w_{(n-1),1} & \cdots & \cdots & w_{(n-1),t} \\ w_{n,1} & \cdots & w_{n,(t-1)} & w_{n,t} \end{bmatrix} \tag{3}$$

4 Unsupervised feature selection problem

The following subsections explain the proposed feature selection method based on the proposed hybrid algorithm (H-FSPSOTC).

4.1 Mathematical model of the feature selection problem

The feature selection problem is formulated as an optimization problem by a new model to find an optimal subset of informative text features. Furthermore, it eliminates uninformative features. The following mathematical notation explains the proposed model for the feature selection problem.

Given F as a set of text document features $F_i = \{f_{i,1}, f_{i,2}, \ldots, f_{i,j}, \ldots, f_{i,t}\}$, where t is the number of all unique features for the documents, j is the feature number, and i is the document number. Let $NF_i = \{nf_{i,1}, nf_{i,2}, \ldots, nf_{i,j}, \ldots, nf_{i,m}\}$ is a new subset of informative features obtained by the feature selection algorithm with a new dimension space (length of features), m is the number of the new unique features (new length of features), and $nf_{i,j} \in \{0, 1\}$, $j = 1, 2, \ldots, m$. if $nf_{i,j} = 1$ which means that the jth feature is selected as an informative text feature in document i, if $nf_{i,j} = 0$ which means that the jth feature is not selected as informative feature in document i [6,23].

Table 1 Solution representation of the feature selection technique

X	0	1	1	−1	−1	1	0	−1	1	−1

4.2 Solution representation

In the proposed H-FSPSOTC for feature selection problem, each candidate solution represents a subset of text features for a document as the solution in Table 1. The swarm of the PSO is a collection of particles which is represented as vectors (row), in each particle has some positions and each position represents one feature. The jth position in the particle represents the status of the jth feature. Hybrid PSO begins with random initial solutions and improves its population to achieve a globally optimal solution [11].

If the value of position jth is 1, the jth feature is selected, if the value of position jth is 0, the jth feature is not selected. Otherwise, if the value of position jth is -1, the jth feature is not in the original document.

4.3 Fitness function

The fitness function is a type of objective functions used to evaluate each solution produced by the algorithm. In each iteration, the fitness function of each solution is calculated to decide if there is an improvement found in the solutions to accept or decline it. Finally, the solution with high fitness value is the optimal solution so far [20]. In this paper, the mean absolute difference (MAD) is used as a fitness function in H-FSPSOTC algorithm for feature selection problem. It is based on the weighting scheme (i..e, TFIDF) for evaluating the solution positions (features) [3,14,24]. MAD is used in the feature selection domain to assign a relevance score (weightiness) for each feature by calculating the difference of each feature from the mean value by Eq. (4).

$$\mathrm{MAD}_{(Xi)} = \frac{1}{a_i} \sum_{j=1}^{t} |x_{i,j} - \bar{x_i}|, \tag{4}$$

where,

$$\bar{x_i} = \left(\frac{1}{a_i}\right) \sum_{j=1}^{t} x_{i,j}, \tag{5}$$

where $\mathrm{MAD}_{(Xi)}$ represents the fitness function of the solution i, $x_{i,j}$ is the value (i.e., weighting value) of the feature j in document i, and a_i is the number of selected features in document i. t is the number of unique features in the original dataset, and $\bar{x_i}$ is the mean value of the vector i.

5 The proposed algorithm

This section presents a comprehensive description for the proposed hybrid PSO algorithm and feature selection problem. Algorithm 2 shows the pseudo-code of the

proposed algorithm (H-FSPSOTC). It consists of three stages: (i) population initialization to fill the initial solutions (ii) updating the solutions positions to provide an appropriate value (iii) binary PSO algorithm to adjust the positions value to be discrete values (IV) genetic operators for the PSO algorithm to improve the global search ability.

5.1 Hybrid particle swarm optimization with genetic operators

PSO is a population-based meta-heuristic optimization algorithm which simulates the social behavior of organisms, such as birds in a flock and fish in a school. The PSO population is called a swarm, and each solution in a swarm is referred to a particle. A population consists of S solutions, each solution represents the potential of solving the problem in a multi-dimensional search space [10,25]. PSO has used the global best (GB) solution and the local best (LB) solution for achieving the optimal solution. In each iteration, the global best solution remains the best solution. The pseudo-code of the proposed H-FSPSOTC is shown in Algorithm (2).

Algorithm 2 Hybrid particle swarm optimization

Input: Initialize the population and PSO parameters:
c_1, c_2, Cr, Mu, LB, and GB.
$Imax$: Number of iterations.
X: A set of current positions (i.e., a solution).
S: Number of solutions.
Output: Optimal solution (Subset of text features).
Algorithm
while Not reach the maximum number of iterations **do**
  for $i = 1$ to S do **do**
    for $j = 1$ to t do **do**
      Evaluate all solutions by Eq. (4).
      Update the LB and GB.
      Update the velocity using Eq. (9).
      Update the position using Eq. (8).
      Applying the genetic operators
      Crossover operator
      Mutation operator
      if $f(X\ best) > f(LB)$ **then**
        $LB = X\ best$
        if $f(GB) > f(LB)$ **then**
          Update LB
          Update GB
        end if
      end if
    end for
  end for
end while
Return a new subset of text features.

5.2 Particle swarm optimization algorithm search space

The meta-heuristic algorithm has generated the population (called a swarm) with random positions (i.e., an initial solution is a random number either 0 or 1 which means that if any position (feature) = 0, this feature does not select as an informative feature, and if any position = 1 the feature is selected as an informative feature). This algorithm enhances the swarm to achieve an optimal best solution (i.e., an optimal subset of the document features). Each position is a dimension of the search space. The solutions are evaluated by the cost function (fitness function), as shown in Eq. (4). The PSO algorithm includes a store of solutions [i.e., particle swarm optimization memory (PSOM)], which is filled by generating S random solutions shown in Eq. (7). The randomly generated solution for the H-FSPSOTC in the initialization step based on using Eq. (6).

$$x_{i,j} = rand \mod 2, \tag{6}$$

where $rand$ in (1 …INTMAX), $i = 1, 2, \ldots, S$ and $j = 1, 2, \ldots, t$. S is number of the candidate solutions, and t is the number of particle positions.

$$\mathbf{PSOM} = \begin{bmatrix} x_{(1,1)} & \cdots & \cdots & x_{(t-1,1)} & x_{(t,1)} & f(\boldsymbol{X}_1) \\ x_{(1,2)} & \cdots & \cdots & x_{(t-1,2)} & x_{(t,2)} & f(\boldsymbol{X}_2) \\ \vdots & \ddots & \ldots & \vdots & \vdots & \vdots \\ x_{(1,S-1)} & \cdots & \cdots & \cdots & x_{(t,S-1)} & f(\boldsymbol{X}_{S-1}) \\ x_{(1,S)} & \cdots & \cdots & x_{(t-1,S)} & x_{(t,S)} & f(\boldsymbol{X}_S) \end{bmatrix} \tag{7}$$

Population initialization is the first step in any meta-heuristic optimization algorithm, which affects the algorithm convergence speed and the solution quality. Due to the absence of any information about the solution of the problem, the random initialization is the most considerably approach used to generate the initial solutions [26,27].

5.3 Positions updating

The PSO algorithm generates particles with random positions according to Eq. (6). Each candidate solution called particle is evaluated by the fitness function as formulated in Eq. (4). In PSO, the solutions contain some single entities (features). PSO is placed in the search space of a feature selection problem and evaluates the fitness function at its current location. Each solution determines its movement by combining aspects of the historical information according to its own current and best fitness. The next iteration selects a locations after all solutions moved. Lastly, the solutions, which are similar to a flock of birds collectively searching for food, will likely go close to an optimal fitness function [10].

PSO works based on two main factors to update each particle position: velocity, as shown in Eq. (9) and particle position, indicated in Eq. (8). The velocity of each

particle is updated according to particle movement effect, and each particle attempts to move to the optimal position [14].

$$x_{ij} = x_{ij} + v_{ij} \tag{8}$$

where,

$$v_{i,j} = w * v_{ij} + c_1 * rand_1 * (LB_I - x_{i,j}) + c_2 * rand_2 * (GB_I - x_{i,j}), \tag{9}$$

The value of inertia weight often changes based on the iteration in the range of [0, 1]. LB_I is the current best local solution at iteration number I, and GB_I is the current best global solution at iteration number I. $rand_1$ and $rand_2$ are random numbers in the range of [0, 1], c_1 and c_2 are usually two constant. The inertia weight is determined by Eq. (10).

$$w = (w_{\max} - w_{\min}) * \left(\frac{I_{\max} - I}{I_{\max}} \right) + w_{\min} \tag{10}$$

where $w_{\max}$ and $w_{\min}$ are the largest and smallest inertia weights, respectively. The values of these weights are constants in the range of (0.5–0.9).

The proposed algorithms deal with binary optimization problems [14]. Hence, the algorithms are modified to update the solutions positions by discrete value for each dimension. Equation (11) represents the Sigmoid function that is used to determine the probability of the ith position, and Eq. (12) is used to update the new position. The Sigmoid function values of the updating process are presented in Fig. 1.

$$s_{i,j} = \begin{cases} 1 & \text{if } rand < \frac{1}{1+exp^{-v_{i,j}}} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where *rand* is a random number between [0, 1], $x_{i,j}$ represents the values of position j, and $-v_{i,j}$ denotes the velocity of particle i at position j, $j = 1, 2, \ldots, t$.

$$x_{i,j} = \begin{cases} 1, & rand < s_{i,j} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

5.4 The genetic operators of PSO

In this paper, the researchers combined the PSO algorithm with the genetic operators to improve the performance of the text feature selection problem. Genetic operators (i.e., crossover and mutation) are inspired from the genetic evolutionary algorithm [19,27].

Crossover operator of PSO algorithm

The crossover is taken from the genetic algorithm as an effective approach for improving the global optimum solution. The crossover probability Cr is used as a control

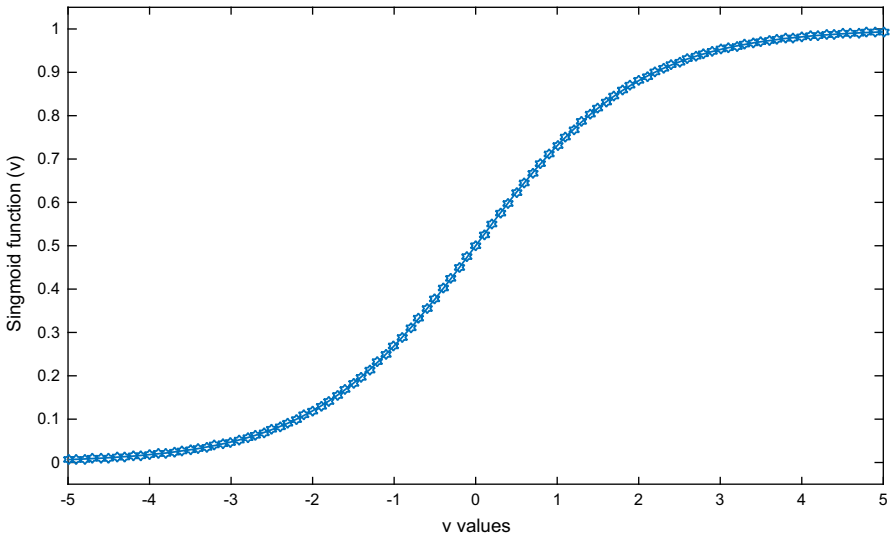

Fig. 1 Sigmoid function used in binary PSO algorithm

parameter for tuning the crossover operator by generating a uniformly distributed random value between [0, 1] [28], the m_th component of $x_{i,m}$ is determined as Eq. (13):

$$x_{i,j} = \begin{cases} x_{r,j} & if rand < Cr \\ x_{i,j} & else \end{cases}. \tag{13}$$

where, $Cr = 0.2, r \in \{1, 2, \ldots, i-1, i+1, \ldots, t\}$. This is a new probability for the global best solution, and it is changeable with increasing and decreasing the fitness function.

Mutation operator of PSO algorithm

Mutation operator plays an important role in genetic algorithms. Mutation probability *Mu* is used as the control parameter for tuning the mutation operator by generating a uniformly distributed random value between [0, 1] [28]. The mutation operator is formulated as:

$$x_{i,j} = \begin{cases} x_{GBj} + \mu(x_{pj} - x_{q,j}) & if rand < Mu \\ x_{i,j} & else \end{cases}. \tag{14}$$

where, $Cu = 0.05$, $p, q \in \{1, 2, \ldots, i-1, i+1, \ldots, t\}$ and μ is value between [0,1]. This probability is a new in the mutation operator to reach the global best solution and it is changeable with the increases and decreases the fitness function [28].

6 Text clustering problem

This section shows the steps of text clustering technique after obtained a new subset of text informative features using the proposed algorithm (H-FSPSOTC), k-mean algorithm, update the cluster centroid, and similarity measure in this research.

6.1 Mathematical model of the text clustering problem

The text clustering technique is defined as follows: given D a set of text documents $D = (d_1, d_2, \ldots, d_j, \ldots, d_n)$ is divided into a subset of clusters K. Where n represents the number of all documents in the given dataset, and d_1 represents the document number 1. Each cluster is represented by one centroid as a vector of terms weight $c_k = (c_{k1}, c_{k2}, \ldots, c_{kj}, \ldots, c_{kK})$, where ck is the cluster centroid number k, $ck1$ is the value of position 1 in the centroid number k, and t is the number of all unique terms in the centroids [29].

6.2 Compute clusters centroid

In order to partition a set of text documents into a subset of clusters, each cluster has one centroid, which must be updated in each iteration using Eq. (15). Each document is assigned to a similar cluster based on the similarity with the cluster centroid, where ck are cluster centroids of k clusters $c_k = (c_{k1}, c_{k2}, \ldots, c_{kj}, \ldots, c_{kK})$, and c_{kj} is the centroid of cluster j [29]. Equation (15) is used in order to calculate cluster centroids.

$$c_k = \frac{\sum_{i=1}^{n} (a_{ki}) d_i}{\sum_{i=1}^{n} a_{ki}}, \tag{15}$$

where d_i is the terms weights (vector) of the document i that belongs to c_j centroid of the cluster j. a_{kj} is a matrix used to find each document to which cluster it belongs. n is the number of all documents, and $\sum_{j=1}^{n} a_{kj}$ is used to find the number of documents in cluster i.

6.3 Similarity measure

Cosine similarity is the common similarity measurement that is used in the text document clustering technique to calculate the similarity between two vectors as d_1 is document number 1 and d_2 is the cluster centroid, which is defined as Eq. (16). Generally, cosine similarity is the common measurement that is used in the text mining area and particularly in the text clustering domain to compute the similarity value between each document with the clusters centroids [30].

$$Cos(d_1, d_2) = \frac{\sum_j w_{1,j} \times w_{2,j}}{\sqrt{\sum_j w_{1,j}^2} \times \sqrt{\sum_j w_{2,j}^2}}, \tag{16}$$

where, $w_{1,j}$ is the weight of term j in document 1 and $w_{2,j}$ is the weight of term j in document 2. $\sqrt{\sum_j w_{1,j}^2}$ is the summation of all terms weights square of the document 1. In the clustering technique, the cosine similarity is used to find the similarity between a document with a cluster centroid, which means that the d_2 represents the cluster centroid [1].

6.4 K-mean text clustering algorithm

K-mean is a common clustering algorithm used in text clustering. MacQueen James and other scholars introduced this algorithm in 1967. The k-mean initially assigns a random initial cluster centroids. It partitions a set of text documents $D = (d_1, d_2, d_3, \ldots, d_n)$ into a subset of clusters K. This algorithm uses the maximum similarity to assign each document for a similar cluster centroid calculated [1,21]. This procedure is presented in Algorithm 3.

K-mean text clustering uses the number of clusters K and the initial cluster centers to identify the related documents in each group or cluster using similarity equation [21]. The similarity value between each text document and clusters centroids iteratively updates the cluster centroids until the termination criterion is met [1].

Algorithm 3 K-mean clustering algorithm [21]

1: **Input**: A collection of text documents , K is the number of all clusters.
2: **Output**: Assign D to K.
3: **Termination criteria**
4: Randomly choosing K document as clusters centroid $C = (c_1, c_2, \ldots, c_K)$.
5: Initialize matrix X as zeros
6: **for** all d in D **do**
7: let $j = argmax_{k \epsilon \{1 to K\}}$, using the cosine similarity.
8: Assign d_i to the cluster j, A[i][j] = 1.
9:
10: Update the clusters centroids using Eq. (15).
11: **end for**

7 Evaluation measures

The comparative evaluations were conducted using one internal evaluation measures (i.e., similarity measure) and four external evaluation measures [i.e., accuracy (Ac), precision (P), recall (R), and F-measure (F)]. These measures are the common evaluation criteria used in the domain of the text clustering to evaluate the clusters accuracy [22].

7.1 Precision, recall, and F-measure

F-measure (F) is a common measurement used in the domain of the text clustering [16]. It depends on two measurements: precision and recall. Precision and recall are

the common measurements used in the area of text mining by Eqs. (17) and (18). F-measure is calculated based on using these two measures by Eq. (20).

$$P(i, j) = \frac{n_{i,j}}{n_j}, \tag{17}$$

$$R(i, j) = \frac{n_{i,j}}{n_i}, \tag{18}$$

where, $n_{i,j}$ is the number of members of class i in cluster j, n_j is the number of members of cluster j, and n_i is the number of members of class i.

$$F(j) = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) \times R(i, j)}, \tag{19}$$

where, $P(i, j)$ is the precision of members of class i in cluster j, $R(i, j)$ is the recall of members of class i in cluster j, and F-measure for all clusters is calculated by the following equation:

$$F = \sum_j \frac{n_j}{n} \max_i \{n(i, j)\}, \tag{20}$$

7.2 Accuracy

The accuracy (AC) measurement is one of the common external measurements used to compute the percentage of correct assigned documents to each cluster according to the following equation [31]:

$$\text{AC} = \frac{1}{n} \sum_{i=1}^{K} n_{i,i} \tag{21}$$

where, $n_{i,j}$ is the number of correct members of the class i in cluster i, n is the number of all documents and K is the number of all clusters.

8 Experimental results

We have programmed the proposed hybrid PSO algorithm for the feature selection problem and the k-mean clustering algorithm for text clustering problem using MATLAB (version 7.10.0) software in Windows 7 environment on a machine with 4GB RAM. This section provides the details of the given datasets, and experimental results and discussion.

Table 2 shows the eight standard benchmark text datasets which are used to investigate the performance of the proposed algorithm (H-FSPSOTC). Text clustering benchmark standard datasets are available at http://sites.labic.icmc.usp.br/text_collections/ by numerical form after the terms extraction. The first dataset (DS1), called Reuters21578, contains 200 random documents belong to four groups. The second dataset (DS2), called 20Newsgroups, contains 100 random documents belong to five groups. The third dataset (DS3), called Reuters21578, contains 100 documents belong

Table 2 Characteristics of the text document datasets

Datasets	# of documents	# of terms	# of clusters
DS1	200	2935	2
DS2	100	3263	4
DS3	100	2063	7
DS4	200	5773	9
DS5	300	312	10
DS6	1000	2650	12
DS7	5000	9641	15
DS8	10,000	19,350	20

to eight groups. The fourth dataset (DS4), called 20Newsgroups, contains 200 random documents belong to ten groups. The fifth dataset (DS5), called Dmoz-Business, contains 300 random documents belong to ten groups. The sixth dataset (DS6), called Dmoz-Science, contains 1000 random documents belong to twelve groups. The seventh dataset (DS7), called Reuters21578, contains 5000 random documents belong to fifteen groups. Finally, the eighth dataset (DS8), called 20Newsgroups, contains 10,000 random documents belong to twenty groups.

8.1 Parameter settings for the proposed algorithm

The experiments are carried out in order to compare four meta-heuristic optimization algorithms, including harmony search algorithm (FSHSTC) [24], genetic algorithm (FSGATC) [3], particle swarm optimization (FSPSOTC) [14], and the proposed hybrid particle swarm optimization (H-FSPSOTC). These algorithms are used various adjustable parameters. The parameters sitting for the comparative algorithms have been taken from the expert papers as researchers recommend them based on their experimental analysis. The H-FSPSOTC is a global search algorithm runs for 1000 iterations in each run. A total of 1000 iterations has been experimentally noted to be sufficient for the convergence of global search algorithm as shown in Table 3 [14]. The results from the proposed method (H-FSPSOTC) are recorded according to 20 replicated runs and compared with the results of three comparative algorithms. For

Table 3 Characteristic of the H-FSPSOTC

Parameter	Value
Number of solution	20
Number of generation	1000
$C1$	2.2
$C2$	2.2
w	Dynamic
Cr	0.9
Mu	0.20

text clustering, the k-mean clustering algorithm proceeds over 100 iterations in each run to obtain the local optimum solution as obtained in previous studies [21].

8.2 Results and discussion

This section shows the results obtained through the research experiments to validate the performance of the proposed algorithm in terms of clustering performance, dimensional space, and computational time and compare the results with that other comparative algorithm used in the domain of the feature selection.

Table 4 shows the performance of the proposed feature selection algorithm (H-FSPSOTC) on eight standard text datasets using four evaluation measurements. The proposed feature selection method using H-FSPSOTC algorithm improved the performance of the text clustering in almost all given datasets according to the evaluation measurements. The proposed H-FSPSOTC performs extremely well to improve the text clustering technique, and it reduced the number of features. The proposed H-FSPSOTC overcomes the other comparative algorithm to deal with a huge collection of text documents with multi-dimensional space. Clearly, it balances these essential components in H-FSPSOTC owing to the combination that applied the GOs after updating the particle positions of the PSO algorithm.

In terms of the feature selection, the proposed H-FSPSOTC performed extremely well and overcame the other comparative algorithm (i.e., FSGATC and FSHSTC). In terms of the text clustering, the proposed H-FSPSOTC performed better results than the other comparative algorithm (i.e., k-mean algorithm without using the feature selection). Lastly, it is clear that the proposed algorithm H-FSPSOTC performs better than the original PSO in all cases.

The statistical analysis (Nemenyi test) is performed using the F-measure values. The average rankings of the text feature selection algorithms are reported in Table 5. The proposed H-FSPSOTC is ranked the highest, which is followed by FSPSOTC, FSGATC, FSHSTC, and K-mean alone without using the feature selection, among the eight datasets.

This section illustrates the experimental results to empirically examine the effectiveness of the proposed hybrid PSO algorithm (H-FSPSOTC) in the application of the feature selection and compare it with the most successful comparative algorithms used to solve that problem. H-FSPSOTC obtained the best performance according to F-measure compared with the comparative algorithms. It performs better on all eight benchmark datasets compared with the comparative algorithms consistent with all evaluation measures. A fit balance between exploration search ability and exploitation search ability enhances the performance of the proposed hybrid PSO algorithm.

Table 6 illustrates that the proposed algorithm (H-FSPSOTC) reduced the dimensions in the feature space over the given dataset. It obtained the lowest number of informative text features effectively in comparison with well-known algorithms. for example, In terms of the first dataset, H-FSPSOTC obtained the lowest number of feature (i.e., 705), followed by FSPSOTC selected 703 features, FSHSTC selected 738 features, and FSGATC selected 805 features. This improvement reduced the computational time and make the k-mean clustering algorithm more effective.

Table 4 Algorithm performance based on clusters quality

Dataset	Method	Text clustering [21]	FSHSTC [24]	FSGATC [3]	FSPSOTC [14]	H-FSPSOTC Our proposed
DS1	Accuracy	0.5565	0.5365	0.5955	0.5845	**0.5999**
	Precision	0.5201	0.5274	0.5690	**0.5754**	0.5712
	Recall	0.5077	0.5046	0.5681	0.5518	**0.5702**
	F-measure	0.5244	0.5011	0.5679	0.5690	**0.5701**
DS2	Accuracy	0.3520	0.3595	0.4070	0.4040	**0.4211**
	Precision	0.2852	0.3166	0.3346	0.3551	**0.3699**
	Recall	0.2718	0.3159	0.3446	**0.3595**	0.3509
	F-measure	0.3057	0.3150	0.3386	0.3559	**0.3586**
DS3	Accuracy	0.5070	0.5025	0.4705	0.5170	**0.5221**
	Precision	0.4721	0.4611	0.4262	0.4768	**0.4852**
	Recall	0.4709	0.4644	0.4261	0.4758	**0.4886**
	F-measure	0.4751	0.4610	0.4262	0.4844	**0.4905**
DS4	Accuracy	0.2707	0.2692	0.2762	0.2862	**0.2889**
	Precision	0.2422	0.2502	0.2578	0.2581	**0.2619**
	Recall	0.2514	0.2499	0.2479	**0.2626**	0.2606
	F-measure	0.2349	0.2491	0.2526	0.2607	**0.2661**
DS5	Accuracy	0.4657	0.4732	0.4655	0.4801	**0.4864**
	Precision	0.4552	0.4623	0.4578	0.4756	**0.4802**
	Recall	0.4351	0.4410	0.4398	0.4562	**0.4664**
	F-measure	0.4487	0.4598	0.4497	0.4665	**0.4723**
DS6	Accuracy	0.6235	0.6435	0.6481	0.6648	**0.6661**
	Precision	0.6201	0.6590	0.6617	0.6801	**0.6912**
	Recall	0.6008	0.6175	0.6199	0.6327	**0.6505**
	F-measure	0.6332	0.6390	0.6452	0.6508	**0.6685**
DS7	Accuracy	0.3354	**0.3498**	0.3248	0.3395	0.4390
	Precision	0.3047	**0.3385**	0.3231	0.3289	0.3375
	Recall	0.2912	0.3261	0.3201	0.3158	**0.3264**
	F-measure	0.2975	**0.3350**	0.3186	0.3232	0.3304
DS8	Accuracy	0.4326	0.4215	0.4375	**0.4416**	0.4414
	Precision	0.4215	0.4278	0.4289	0.4227	**0.4547**
	Recall	0.4105	0.4014	0.4129	0.4154	**0.4478**
	F-measure	0.4175	0.4112	0.4201	0.4187	**0.4519**

The best results are highlighted in bold

8.3 Computational time

This section illustrates that the proposed H-FSPSOTC obtained the highest clustering quality and outperforms other comparative methods. Table 7 shows the received computational time for the four feature selection algorithms. This section focuses on a number of the selected features because this advantage primarily reduces the exe-

Table 5 The average ranking of the feature selection algorithm based on the average F-measure

Method no.	Description	Dataset								Mean rank	Ranking
		1	2	3	4	5	6	7	8		
01	K-mean	04	05	03	05	05	05	05	04	04.50	5
02	FSHSTC	05	04	04	04	03	04	01	05	03.75	4
03	FSGATC	03	03	05	03	04	03	04	02	03.37	3
04	FSPSOTC	02	02	02	02	02	02	03	03	02.25	2
05	H-FSPSOTC	01	01	01	01	01	01	02	01	01.12	1

The lowest ranking value is the best method

Table 6 Number of selected features along number of iterations

Dataset	Number of iterations	Algorithm			
		FSHSTC	FSGATC	FSPSOTC	H-FSPSOTC
DS1 (2935)	100	2935	2259	2196	2198
	200	2150	1954	2310	1999
	300	1430	1235	1395	1248
	400	745	811	370	686
	500	738	805	730	705
DS2 (3263)	100	2641	2568	2418	2414
	200	2154	2293	2097	2169
	300	1745	1654	1565	1541
	400	354	399	432	355
	500	328	382	460	312
DS3 (2063)	100	1451	1265	1554	1352
	200	1240	1021	1145	1234
	300	756	768	801	769
	400	489	576	599	578
	500	469	547	530	522
DS4 (5773)	100	3657	3266	3759	3985
	200	2650	2549	2694	2458
	300	1640	1791	1785	1690
	400	901	928	1089	920
	500	779	869	833	746
DS5 (312)	100	245	263	257	245
	200	214	242	236	201
	300	189	196	194	176
	400	176	199	184	166
	500	165	176	169	153

Table 6 continued

Dataset	Number of iterations	Algorithm			
		FSHSTC	FSGATC	FSPSOTC	H-FSPSOTC
DS6 (2650)	100	2012	1978	2154	1896
	200	1423	1369	1546	1310
	300	1120	1234	1325	1077
	400	962	956	990	910
	500	894	871	860	846
DS7 (9641)	100	8564	8475	8475	8310
	200	7894	7841	7748	7640
	300	7654	7668	7650	7612
	400	7548	7612	7541	7452
	500	7515	7588	7522	7420
DS8 (19,350)	100	17,548	17,486	16,450	16,120
	200	17,320	17,365	16,352	16,014
	300	17,265	17,121	16,210	15,795
	400	16,980	16,899	16,021	15,683
	500	16,945	16,841	16,012	15,616

cution time of the text clustering method. A rigorous validation of the computational execution time analysis shows that the proposed H-FSPSOTC recorded the shortest time and obtained accurate clusters. The proposed H-FSPSOTC method outperformed the comparative algorithms (i.e., FSHSTC, FSGATC, and H-FSPSOTC) regarding the shortest execution time in almost all of the given datasets. Thus, the obtained subset of text features is a useful subset, which improves the performance of the k-mean text clustering with the reduction of computational time in comparison with related methods. The proposed H-FSPSOTC reduced the computational time significantly more than other algorithms while preserving clustering improvement.

8.4 Convergence behavior

In order to present convergence properties of the comparative algorithms. This section shows a comparative analysis for the feature selection algorithms performed according to the fitness function (MAD) values. This behavior related to a number of iterations and the computational execution time. Figure 2 shows the average solution history graph along with the number of iterations. Clearly, the proposed H-FSPSOTC starts with a set of reliable solutions and it is able to improve the solutions along the number of iterations. H-FSPSOTC obtained the better results in comparison with the other comparative algorithms.

Finally, it is clear that the convergence of the proposed algorithm in comparison with the original PSO algorithm is more controlled because the PSO trapped in the local optimal solution and sometimes it got premature convergence. However, H-

Table 7 Computational time of the feature selection algorithms

Dataset	Algorithm	Time (in second)	Ranking
DS1	FSGATC	2050.215	2
	FSHSTC	1921.452	1
	FSPSOTC	1964.568	3
	H-FSPSOTC	2036.411	4
DS2	FSGATC	1230.648	4
	FSHSTC	1123.942	2
	FSPSOTC	1146.510	3
	H-FSPSOTC	1144.011	1
DS3	FSGATC	1470.826	4
	FSHSTC	1201.254	1
	FSPSOTC	1299.699	3
	H-FSPSOTC	1264.258	2
DS4	FSGATC	3204.102	2
	FSHSTC	2953.214	1
	FSPSOTC	3325.012	4
	H-FSPSOTC	3251.218	3
DS5	FSGATC	4602.188	4
	FSHSTC	4329.092	3
	FSPSOTC	4125.195	2
	H-FSPSOTC	3998.548	1
DS6	FSGATC	14,052.971	4
	FSHSTC	13,225.252	1
	FSPSOTC	13,697.555	2
	H-FSPSOTC	13,954.214	3
DS7	FSGATC	61,224.564	4
	FSHSTC	54,936.650	1
	FSPSOTC	58,108.392	3
	H-FSPSOTC	55,780.000	2
DS8	FSGATC	133,494.214	4
	FSHSTC	122,992.500	2
	FSPSOTC	132,273.000	3
	H-FSPSOTC	120,452.154	1

FSPSOTC is more productive than the PSO algorithm and the other comparative algorithms in terms of the algorithms convergence, performance, execution time, and better subset of informative feature than the comparative algorithms. We conclude that the proposed hybrid PSO algorithm enhances the performance and prevent the premature convergence of the original PSO. It is used successfully for solving the feature selection problem in the form of the proposed method.

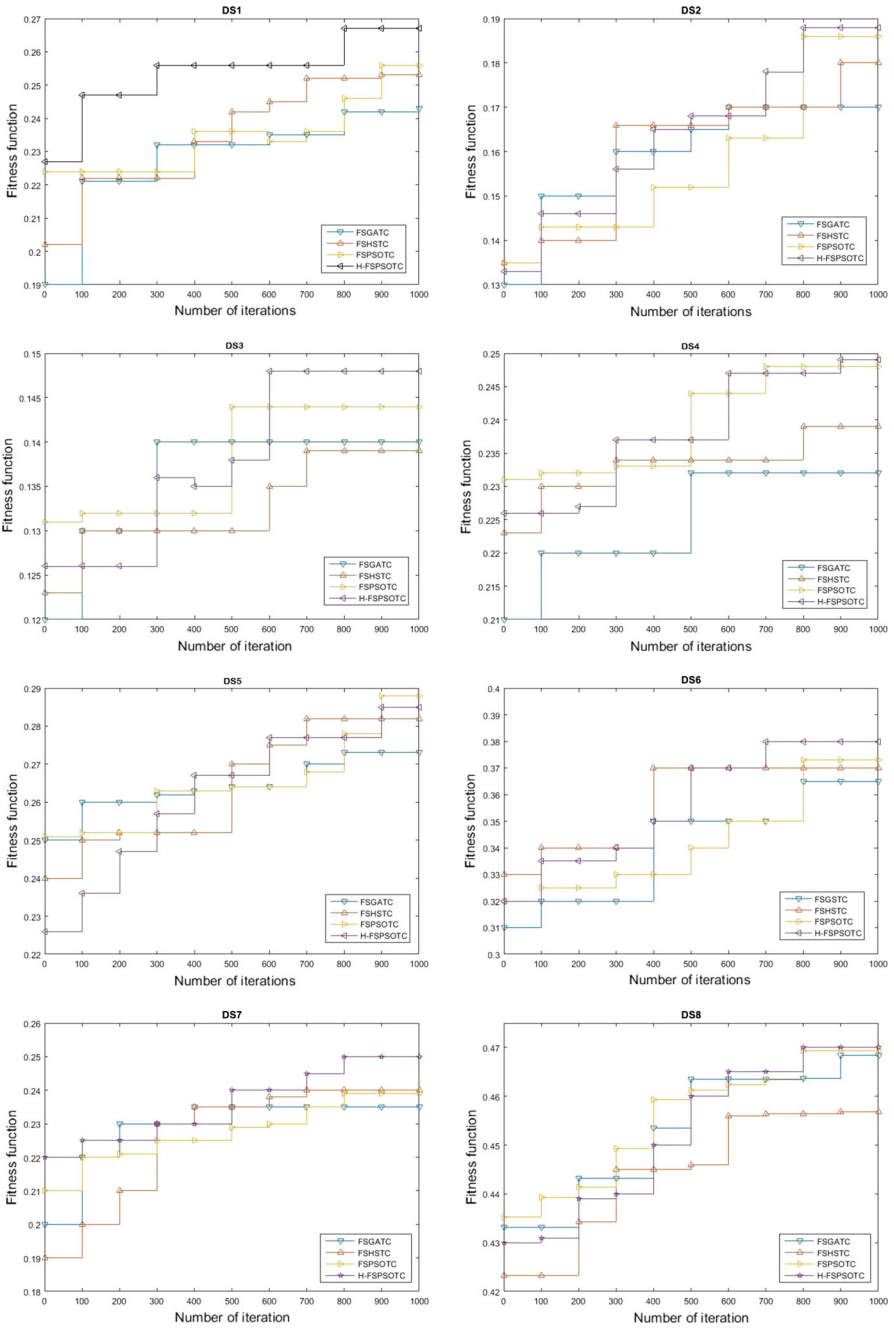

Fig. 2 Comparison of convergence properties for all feature selection algorithms

9 Conclusion

This paper presents a new feature selection method based on the hybrid PSO algorithm with the GOs (H-FSPSOTC). Text document clustering includes the problem of grouping documents into proper clusters based on shared characteristics. Before the clustering performed, the informative features will be defined to yield accurate groups. Therefore, the hybrid PSO algorithm with the GOs (H-FSPSOTC) is adapted to text feature selection methods. The results of the text feature selection method are used by the k-mean clustering algorithm to yield accurate clusters.

Eight standard benchmark text datasets related to text mining are used for performance and comparative evaluations, which were selected from the Laboratory of Computational Intelligence. A total of four comparatives of the text feature selection methods are examined to determine the best algorithm. The results from H-FSPOSTC are the best among those of other comparatives. Thus, using hybrid PSO for the proposed feature selection method will increase the performance of produced text features and text clustering technique, such as the k-mean algorithm, will be more accurate and result in better accuracy and F-measure.

The hybrid PSO algorithm and feature selection method contributions in this paper can be very useful for the text mining research community. Feature selection method can be further improved by introducing a new fitness function combined the features weighting and the number of the selected features. The used text mining datasets are known datasets. However, other more rigorous text or date datasets can be used for the evaluation process. Another recently developed and successful meta-heuristic method can be employed.

References

1. Abualigah LM, Khader AT, Al-Betar MA, Awadallah MA (2016) A krill herd algorithm for efficient text documents clustering. In: 2016 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE). IEEE, pp 67–72
2. Rao B, Mishra BK (2017) An approach to clustering of text documents using graph mining techniques. International Journal of Rough Sets and Data Analysis (IJRSDA) 4(1):38–55
3. Abualigah LM, Khader AT, Al-Betar MA (2016) Unsupervised Feature Selection Technique Based on Genetic Algorithm for Improving the Text Clustering, pp 1–6
4. Li C, Lin M, Yang LT, Ding C (2014) Integrating the enriched feature with machine learning algorithms for human movement and fall detection. J Supercomput 67(3):854–865
5. Xu S, Zhang J (2004) A parallel hybrid web document clustering algorithm and its performance study. J Supercomput 30(2):117–131
6. Bharti KK, Singh PK (2015) Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. Expert Syst Appl 42(6):3105–3114
7. Bu F, Chen Z, Zhang Q, Yang LT (2016) Incomplete high-dimensional data imputation algorithm using feature selection and clustering analysis on cloud. J Supercomput 72(8):2977–2990
8. Xu J, Xu B, Wang P, Zheng S, Tian G, Zhao J (2017) Self-taught convolutional neural networks for short text clustering. Neural Netw 30(2):117–131
9. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
10. Lu Y, Liang M, Ye Z, Cao L (2015) Improved particle swarm optimization algorithm and its application in text feature selection. Appl Soft Comput 35:629–636
11. Bharti KK, Singh PK (2016) Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. Appl Soft Comput 43:20–34

12. Kabir MM, Shahjahan M, Murase K (2012) A new hybrid ant colony optimization algorithm for feature selection. Expert Syst Appl 39(3):3747–3763
13. Ghamisi P, Benediktsson JA (2015) Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci Remote Sens Lett 12(2):309–313
14. Abualigah LM, Khader AT, AlBetar MA, Hanandeh ES (2017) Unsupervised Text Feature Selection Technique Based on Particle Swarm Optimization Algorithm for Improving the Text Clustering. EAI
15. Shamsinejadbabki P, Saraee M (2012) A new unsupervised feature selection method for text clustering based on genetic algorithms. J Intell Inf Syst 38(3):669–684
16. Hong SS, Lee W, Han MM (2015) The feature selection method based on genetic algorithm for efficient of text clustering and text classification. Int J Adv Soft Comput Appl 7(1):22–40
17. Lin KC, Zhang KY, Huang YH, Hung JC, Yen N (2016) Feature selection based on an improved cat swarm optimization algorithm for big data classification. J Supercomput 72:1–12
18. Diao R (2014) Feature selection with harmony search and its applications. Aberystwyth University, Aberystwyth
19. Abualigah LMQ, Hanandeh ES (2015) Applying genetic algorithms to information retrieval using vector space model. Int J Comput Sci Eng Appl 5(1):19
20. Uğuz H (2011) A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowl Based Syst 24(7):1024–1032
21. Abualigah LM, Khader AT, Al-Betar MA (2016) Multi-objectives-Based Text Clustering Technique Using K-Mean Algorithm. 2016 July, pp 1–6
22. Bharti KK, Singh PK (2014) A three-stage unsupervised dimension reduction method for text clustering. J Comput Sci 5(2):156–169
23. Bharti KK, Singh PK (2013) A two-stage unsupervised dimension reduction method for text clustering. In: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012) Volume 2. Springer, 2013, pp 529–542
24. Abualigah LM, Khader AT, Al-Betar MA (2016) Unsupervised Feature Selection Technique Based on Harmony Search Algorithm for Improving the Text Clustering. 2016 July, pp 1–6
25. Liu Y, Wang G, Chen H, Dong H, Zhu X, Wang S (2011) An improved particle swarm optimization for feature selection. J Bionic Eng 8(2):191–200
26. Nekkaa M, Boughaci D (2015) Hybrid harmony search combined with stochastic local search for feature selection. Neural Process Lett 44:1–22
27. Bolaji AL, Al-Betar MA, Awadallah MA, Khader AT, Abualigah LM (2016) A comprehensive review: Krill Herd algorithm (KH) and its applications. Appl Soft Comput 49:437–446
28. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
29. Forsati R, Mahdavi M, Shamsfard M, Meybodi MR (2013) Efficient stochastic algorithms for document clustering. Inf Sci 220:269–291
30. Zhao Z, Wang L, Liu H, Ye J (2013) On similarity preserving feature selection. IEEE Trans Knowl Data Eng 25(3):619–632
31. Nassirtoussi AK, Aghabozorgi S, Wah TY, Ngo DCL (2015) Text mining of news-headlines for FOREX market prediction: a multi-layer dimension reduction algorithm with semantics and sentiment. Expert Syst Appl 42(1):306–324