Cognitive Computation (2019) 11:271–293
https://doi.org/10.1007/s12559-018-9611-8

Automatic Scientific Document Clustering Using Self-organized Multi-objective Differential Evolution

Naveen Saini[1] · Sriparna Saha[1] · Pushpak Bhattacharyya[1]

Received: 6 April 2018 / Accepted: 12 November 2018 / Published online: 19 December 2018

Abstract

Document clustering is the partitioning of a given collection of documents into various K- groups based on some similarity/dissimilarity criterion. This task has applications in scope detection of journals/conferences, development of some automated peer-review support systems, topic-modeling, latest cognitive-inspired works on text summarization, and classification of documents based on semantics, etc. In the current paper, a cognitive-inspired multi-objective automatic document clustering technique is proposed which is a fusion of self-organizing map (SOM) and multi-objective differential evolution approach. The variable number of cluster centers are encoded in different solutions of the population to determine the number of clusters from a data set in an automated way. These solutions undergo various genetic operations during evolution. The concept of SOM is utilized in designing new genetic operators for the proposed clustering technique. In order to measure the goodness of a clustering solution, two cluster validity indices, Pakhira-Bandyopadhyay-Maulik index, and Silhouette index, are optimized simultaneously. The effectiveness of the proposed approach, namely self-organizing map based multi-objective document clustering technique (SMODoc_clust) is shown in automatic classification of some scientific articles and web-documents. Different representation schemas including tf, tf-idf and word-embedding are employed to convert articles in vector-forms. Comparative results with respect to internal cluster validity indices, namely, Dunn index and Davies-Bouldin index, are shown against several state-of-the-art clustering techniques including three multi-objective clustering techniques namely MOCK, VAMOSA, NSGA-II-Clust, single objective genetic algorithm (SOGA) based clustering technique, K-means, and single-linkage clustering. Results obtained clearly show that our approach is better than existing approaches. The validation of the obtained results is also shown using statistical significant t tests.

Keywords Clustering · Cluster validity indices · Self Organizing Map (SOM) · Differential Evolution (DE) · Polynomial mutation · Multi-objective Optimization (MOO)

Introduction

Background

Document clustering [1] refers to partitioning of a given collection of documents into various K-groups based on some similarity/dissimilarity criterion so that each document in a group is similar to other documents in the same group. Various applications of document clustering include: extraction of relevant topics [12], organization of documents as in digital libraries [63], creation of document taxonomy [22] such as in Yahoo, document summarization [25] etc. For the purpose of clustering, the value of K may or may not be known a priori. To determine the value of K in the collection of documents, traditional clustering approaches [44] like K-means [31], bisecting K-means [59], hierarchical clustering techniques [31] are required to be executed multiple times with various values of K. The qualities of different partitionings are measured with respect to some cluster validity indices, measuring the goodness of a partitioning by monitoring different intrinsic properties of clusters. Finally, the partitioning which corresponds to the optimal value of any cluster validity index is selected

✉ Naveen Saini
naveen.pcs16@iitp.ac.in; naveen.pcs16@gmail.com

Sriparna Saha
sriparna@iitp.ac.in

Pushpak Bhattacharyya
pb@iitp.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, 801103 Bihar, India

as the final partitioning. Davies-Bouldin (DB) index [17], Silhouette index (SI) [53, 58], Xie-Beni (XB) index [51], Pakhira-Bandyopadhyay-Maulik (PBM) [47] index etc. are some popularly used cluster validity indices. Cluster validity indices which do not support overlap between clusters are called crisp indices. Examples include Davies-Bouldin index [17], Silhouette index [53, 58]. Some cluster validity indices are called fuzzy indices which support the overlap between clusters, for example, Xie-Beni index [51], Pakhira - Bandyopadhyay-Maulik index [47].

The Existing traditional clustering techniques implicitly optimize an internal evaluation function or objective function. These objective functions in general measure the compactness of clusters [37], spatial separation between clusters [37], connectivity between clusters [52], density or cluster symmetry [51]. But in real life, all these properties cannot be captured using a single objective function. Also, for a given data set possessing clusters of different geometrical shapes (like hyper-spherical, convex etc.), use of a single objective function measuring the cluster quality may not be suitable for determining all types of clusters. Application of any multi-objective optimization technique [10, 18] optimizing different cluster validity indices appears to be an alternative and promising direction in clustering research in recent years. This motivates researchers to develop some multi-objective based clustering algorithms [4, 53, 60]. Also, determining the appropriate number of clusters from a given data set in an unsupervised way is another important consideration. Simultaneous optimization of multiple cluster validity indices can also address this issue. Most of the existing multi-objective clustering approaches utilize different types of evolutionary techniques (EAs) as the underlying optimization strategies. Some of the examples of EAs are particle swarm optimization (PSO) [33], genetic algorithm (GA) [35], differential evolution (DE) [60] etc.

In [5], GCUK (genetic clustering with unknown K), an automatic clustering approach was proposed. It optimizes a single cluster validity index, *Xie-Beni* index [44] and is able to detect only hyperspherical shaped clusters. In [8], a symmetry distance-based automatic genetic clustering algorithm, namely VGAPS clustering, was proposed, which detects the number of clusters as well as the optimal partitioning from a data set in an automated way. However, it also optimizes a single cluster validity measure, point symmetry distance based *Sym* index [6] and can detect only point symmetric clusters. Both GCUK [5] and VGAPS [8] are popular automatic clustering techniques. They are rarely applicable to different kinds of data sets having various characteristics. In order to detect clusters having different shapes/sizes/convexities, in recent years, some symmetry based automatic multi-objective clustering techniques [51, 53] are proposed by one of the co-authors of this paper. These algorithms utilize archived multi-objective simulated annealing [10] process as the underlying optimization technique. All these clustering techniques utilized a newly developed symmetry based distance [6] for assigning points to different clusters. Handl et al. [28] developed an automatic multi-objective clustering technique, MOCK. The major limitation of MOCK is that it can determine only some well-separated and hyper-spherical shaped clusters and was not able to detect overlapping clusters. Moreover, the complexity of MOCK increases linearly with the increase in the number of data points. Some multi-objective clustering techniques using differential evolution algorithm [49] as the underlying optimization strategy were proposed in [16] and [60]. Experimental results reported in those works clearly showed that differential evolution has *faster convergence* rate as compared to other evolutionary algorithms and can serve as better optimization strategy for devising any multi-objective clustering technique. Although all the above discussed clustering techniques are automatic in nature, their applications are shown only for partitioning some artificial and real-life numeric data sets. Also, all these algorithms used normal reproduction operators as used in the single objective differential evolution process.

Recent years have witnessed some works on document classification. Steinbach et al. [59] had made a comparative performance study of different document clustering techniques, including K-means [31] and bisecting K-means [32], for clustering different document data sets. Xu et al. [64] have used the non-negative matrix factorization of the term-document matrix for document clustering where the number of topics is required to be known beforehand. Authors of [64] assumed that the number of clusters and the number of topics are known in advance. However, these assumptions are not realistic as the correct value of the number of clusters/topics depends on data distribution which is difficult to approximate in case of document-collection. Moreover, domain knowledge should be acquired for correctly estimating the value of the number of clusters/topics.

Recently, [2, 27] reported some bio-inspired works of text-summarization. They have developed single document text summarization systems for Arabic and Punjabi texts. The proposed clustering technique of the current paper can be easily applied for text-summarization by first performing clustering of sentences (considering each sentence as a document) present in the document and then extracting the most important sentences from each cluster to obtain the summary. In [54], a plagiarism detection system is developed using semantic and syntactic information present in text documents. Chen et al. [15] developed an approach for Chinese text document classification based on semantic topics. Our proposed approach which is unsupervised in nature can also be used for the similar task.

In [55], an algorithm similar to the proposed automatic clustering technique is developed by co-authors of this paper. But the application of the approach [55] was shown only for clustering some artificial low-dimensional numeric data sets. The current paper has proposed a cognitive-inspired multi-objective clustering framework for automatically partitioning a given collection of scientific documents exploiting syntactic and semantic information to identify possible subtopics. In other words, the approach discussed in [55] is extended to solve a real-life problem, scientific document clustering. An automatic categorization of scientific documents is important for several tasks including scope detection of journals/conferences, development of some automated peer-review support systems, topic-modeling, etc. Scientific documents in general, are of varying complexities and categories are highly overlapping in nature. Various pre-processing steps are required to be applied to clean the documents. For example, removal of most frequent words, e.g., *is, am, are etc.*, stemming [57] etc. In order to further process the data, various representation schemas like tf-idf [43], word2vec [39, 45] and Glove [48] are applied to convert documents into numeric vectors. These representations are popular and were used in several recently published [38, 40] cognitive-inspired works on sentiment analysis. Finally, these vectors are grouped into different categories using a newly developed clustering technique.

Motivation

In this section, we describe the motivation behind developing the current automatic document clustering technique utilizing the power of SOM in designing some new reproduction operators.

1) Literature survey reveals that in the field of document clustering, there is no work which can automatically estimate the number of clusters and the appropriate partitioning from a document collection of varying complexities.
2) In recent years, the researchers are working towards utilizing the potentiality of self-organizing map [29, 34] in developing some new reproduction operators as opposed to traditional reproduction operators used in evolutionary techniques. Some evolutionary algorithms like SOMEA/D [65] and SMEA [66], are developed in recent years utilizing the above concepts and are successfully validated on standard benchmark datasets [26]. It was shown that these algorithms perform better than other state-of-the-art evolutionary algorithms.

Motivated by these, current paper proposes a novel self-organizing map based automatic multi-objective document clustering technique, namely SMODoc_clust. Some new genetic operators utilizing the neighborhood information extracted using SOM are incorporated in the proposed approach. SOM [29, 34] is a special type of artificial neural network which learns from the data in an unsupervised way. It maps high dimensional input space to low dimensional output space and preserves the topological properties of the input data. In our proposed clustering based framework, first SOM is trained using the solutions present in the current population. In order to apply genetic operator on a given solution, the closer (neighboring) solutions identified by SOM in the topographical map are extracted and only these extracted solutions can take part in generating high-quality new solutions.

The proposed clustering approach is automatic in nature as it can determine the number of clusters present in a dataset automatically. Center-based encoding is used in the current approach where a set of cluster centers are coded in the form of a chromosome. The number of cluster centers present in different chromosomes varies over a range. In order to measure the quality of a partitioning, different internal cluster validity measures are deployed. The values of these different cluster validity indices are simultaneously optimized using the search capability of multi-objective DE. In order to show the efficacy of the proposed clustering technique, the problem of document classification is considered. Two data sets containing some scientific articles with varying complexities and a data set containing some web-documents are chosen for the purpose of evaluation of the proposed clustering technique. In order to represent the articles in the form of vectors, different representation schemas like tf [43], tf-idf [43], word embeddings [39, 45, 48] are exploited. Similar to any MOO-based approach, our proposed clustering approach also generates a set of solutions on the final Pareto optimal front. A single solution can be selected by the user depending on the requirement. In the current study, a single best solution is selected using some internal cluster validity indices, namely Dunn Index [44] and Davies-Bouldin index [17]. The obtained partitioning results are compared with those obtained by some existing state-of-the-art clustering techniques namely, MOCK [28], AMOSA based multi-objective clustering (VAMOSA) [51], NSGA-II based multi-objective clustering technique (NSGA-II-Clust) [9, 23], single objective genetic algorithm (SOGA) based clustering [7], K-means [31] and single-linkage [31] clustering approach with respect to different performance measures.

In a part of the paper, we have also shown the utility of incorporating SOM-based genetic operators in the clustering process. A multi-objective DE-based clustering approach (without using SOM-based operators), MODoc_clust, is implemented and the results by this approach are compared with the results obtained by the proposed SMODoc_clust (with SOM based operators). The comparative study evidently indicates the effectiveness of SOM based operators in the proposed clustering framework. Furthermore, in order to show the superiority of our proposed clustering approach, statistical *t* tests guided by [21] are also conducted.

Key Contributions

The key contributions of the proposed clustering technique are summarized below :

1. The proposed clustering approach, namely SMOD-oc_clust is the fusion of self-organizing map and multi-objective differential evolution approach [60].
2. The proposed approach with variable length chromosomes is capable of automatically detecting the number of clusters from any given data set.
3. In the proposed framework, two cluster validity indices, PBM index [47] and Silhouette index [53, 58] are simultaneously optimized for the automatic determination of the appropriate number of clusters and also to improve the quality of clusters.
4. Some new genetic operators are proposed in the framework of multi-objective DE. The mating pool constructed for crossover operation given a solution only contains the neighboring solutions identified by SOM. For the training of SOM, the solutions of the current population are utilized. The constructed mating pool takes part in generating some new solutions.
5. The results of the proposed technique are shown for clustering two document data sets containing scientific articles with varying complexities and a document data set containing some web-documents. The experimental results evidently prove that the proposed clustering technique performs well for document classification.

The rest of the sections are organized as follows. “Background” briefly reports on the self-organizing map and the definitions of cluster validity indices used in this paper. “Proposed Methodology” demonstrates the proposed methodology. “Data Sets Used” discusses the data sets used. “Comparing Methods” describes state of the art techniques used for comparison. The experimental results and the significance of proposed approach are summarized in “Experimental Setup and Results”. Finally, “Conclusions and Future Works” concludes the paper.

Background

Self Organizing Map

Self Organizing Map [29, 34] or SOM developed by Kohonen is a type of artificial neural network which learns the data presented to it in an unsupervised way. It generates a low-dimensional output space for the given input space which is consisting of high-dimensional training data. Usually, low-dimensional space (also called an output space) consists of a 2-D regular grid of neurons. These neurons are called as map units. Let S be a set of training data in n-dimensional space, then each map unit u ∈ D (number of map units) has:

1. a predefined position in the output space: z^u=(z^u_1, z^u_2)
2. a weight vector $w^u = [w^u_1, w^u_2....w^u_n]$, where n is the input vector dimension, u is the index of map unit in 2-dimensional Map

Figure 1 shows the typical architecture of SOM. In this example, input space and output space are $n-$ dimensional and 2-dimensional, respectively.

The main principle of SOM is to create a topographical map such that input patterns which are similar in nature in the input space map to neurons next to each other. In our work, the sequential learning algorithm [29] is utilized for the training of SOM as shown in Algorithm 1. This algorithm returns the updated weight vectors of different map units at the output. Before training of SOM, there is a need to assign a weight vector to each neuron, randomly chosen from the available training data. At each iteration, when an input pattern is presented to the grid, then weight vector of the winning neuron (closer to presented input pattern) and neighboring neurons are updated to make them close to the input pattern.

Algorithm 1 SOM Framework(η_0, σ, S, T)

```
Initialize learning constant $\eta_0$ and neighborhood size $\sigma_0$; maximum iteration count T for SOM training; Initialize each map unit by assigning a weight vector randomly chosen from training data S.
while $t \neq T$ do                ▷ t is the current iteration no.
    Adjust learning rate ($\eta$) and neighborhood size ($\mu$) as $\eta = \eta_0 * (1 - \frac{t}{T}), \sigma = \sigma_0 * (1 - \frac{t}{T})$.
    Randomly select a training input pattern x ∈ S
    Find winning map unit: $u^{`} = arg\min_{1 \leq u \leq D} \parallel x - w^u \parallel_2$
    Find the neighboring neurons: $U = \{u|1 \leq u \leq D \parallel z^u - z^{u^{`}} \parallel_2 < \sigma\}$
    Update all neighboring neurons u∈U : $w^u = w^u + \eta * \exp(- \parallel z^u - z^{u^{`}} \parallel_2) * (x - w^u)$
return The weight vectors corresponding to map units, $w^u, u = 1, 2 \ldots, D$
```

Cluster Validity Indices

Cluster validity indices measure the quality of a partitioning obtained using a given clustering technique. These indices

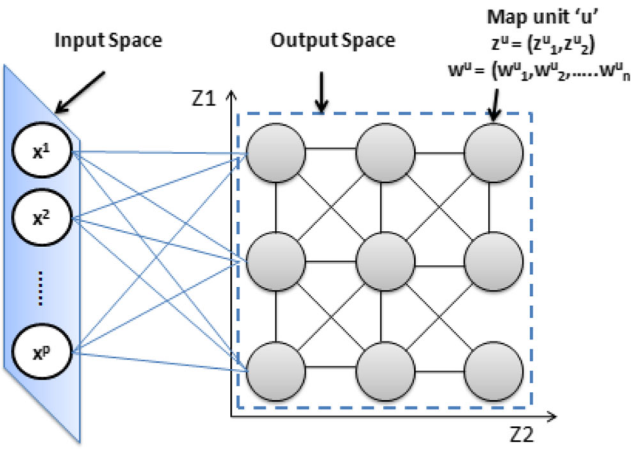


Fig. 1 SOM Architecture (taken from [56]). Here $x^p = x_1^p, x_2^p .....x_n^p$ is the input vector, $Z1$ and $Z2$ denote the axis of 2-D Map, w^u is the weight vector of uth neuron

also help in determining the correct number of clusters from a dataset in an iterative way. Generally, there are two types of cluster validity indices:

1. External Cluster Validation Indices: These indices require external knowledge provided by the user (ground truth/ original labels) to measure the goodness of obtained partitioning. Minkowski Scores [51], Adjusted Rand Index [60] etc. are some examples of external validity indices.
2. Internal Cluster Validation Indices: These indices generally rely on the intrinsic structures of the data and do not require ground truth labels. Most of the internal validity indices measure the intra-cluster distance (compactness within clusters) and inter-cluster separation (separation between clusters). Silhouette index (SI) [53, 58], Dunn index (DI) [44], Davies-Bouldin index (DB)[17], Xie-Beni (XB) index [51], PBM index [47] etc. are some popular internal cluster validity indices.

Out of these indices, PBM index [47], SI [53], DI [44] and DB [17] index are used in this paper. Note that all these indices are internal cluster validity measures. The formal definitions of these indices are presented in Table 1.

Proposed Methodology

In this paper, we have proposed a new multi-objective document clustering (SMODoc_clust) technique to automatically determine the appropriate partitioning from a collection of text documents. The flow-chart of the proposed architecture is shown in Fig. 2. Several new concepts are incorporated in the framework of the proposed clustering technique. SMODoc_clust utilizes the DE [66] framework as the underlying optimization technique for determining the optimal partitioning. The basic operations of SMODoc_clust are described below.

Solution Representation and Population Initialization

In SMODoc_clust, solutions encode a set of different cluster centers. As the proposed algorithm attempts to determine the optimal set of cluster centers that can partition the document dataset appropriately, the number of cluster centers encoded in different solutions are varied over a range. The number of clusters is varied between 2 and $\sqrt{N}$, where N is the total number of points (documents). To generate ith solution, a random number (K_i) is selected between two values, i.e., $K_{min} = 2$ and $K_{max} = \sqrt{N}$ and these K_i number of initial cluster centers are chosen randomly from the dataset. As these solutions take part in SOM training to learn the distribution pattern of the population, lengths of input vectors (solution) and weight vectors of neurons are kept equal. Therefore, variable length solutions are converted to some fixed length vectors by appending zeros at the end. If F indicates the number of features in the dataset, then maximum length of the solution can be $(K \times F + l)$, where K is the number of clusters present in a solution, l is the number of appended zeros lying between '$0'$ and $(K \times F - 2 \times F)$. Here, we have subtracted $2 \times F$ because there must exist at least two clusters in the dataset. In terms of data points, the maximum length of a solution can be $\sqrt{N}$*F.

This set of solutions with the varying number of clusters forms the initial population. In order to obtain a partitioning corresponding to a solution in the population, steps of K-means clustering technique [31] are executed on the whole data set considering the cluster centers encoded in the solution as initial cluster centers. Each point is assigned to that center which is at a minimum Euclidean distance among all the centers encoded in the chromosome. Finally, clusters are identified and the averages of the points belonging to individual clusters are calculated. These are used to replace the old centers present in a solution/chromosome. Population (P) initialization step is shown in Fig. 3 and an example of solution encoding is given below.

Example Let K=3, F=2, N=16. Let three centers be $C_1 = (2.3, 1.4), C_2 = (7.6, 12.9)$ and $C_3 = (2.1, 3.4)$. Here, maximum length of solution=$\sqrt{N} \times F$=(4 $*$ 2)=8. Then, solution will be represented as {(2.3, 1.4, 7.6, 12.9, 2.1, 3.4, 0.0, 0.0)} which encodes three cluster centers, with $l = 2$.

Table 1 Definitions of Cluster validity measures/indices

Measure	Definition	Description	Optimization type
PBM index [47]	$PBM = \left(\frac{1}{K} X \frac{E_1}{E_K} X D_K\right)^2$ $E_K = \sum_{s=1}^{K} E_s$ $E_s = \sum_{j=1}^{N} \mu_{sj} \parallel x_j - c_s \parallel^2$ $E_1 = \sum_{x \in X} \parallel x - c \parallel^2$ $D_K = \max_{i,j=1,i \neq j}^{K} \parallel c_i - c_j \parallel^2$	-K : number of clusters; -E_K: total within-cluster scatter; -N: number of data points; -$[\mu_{sj}]_{K \times N}$: membership matrix of the data; -c_s: s^{th} cluster center; -c: cluster center of the whole data set; -D_K: maximum separation between clusters	Maximum
Silhouette Index (SI) [53]	$SI = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{z_{i2} - z_{i1}}{max(z_{i2}, z_{i1})} \right)$	-N: number of data points; -z_{m1}: average distance of a point x_m belonging to kth cluster to the remaining points of the same cluster; -z_{m2}: minimum of the average distances of the same point x_m from points belonging to other clusters.	Maximum
Dunn Index (DI) [44]	$DI = \frac{\min_{C_k, C_l \in \Im, C_k \neq C_l} (\min_{i \in C_k, j \in C_l} dist(i, j))}{\max_{C_m \in \Im} diam(C_m)}$	-i and j denote the data points; -$\Im$: any clustering algorithm; C_k, C_l, C_m: different clusters; -$diam(C_m)$: the diameter of mth cluster calculated using the Euclidean distance between two points of the same cluster.	Maximum
Davies-Bouldin Index (DB) [17]	$DB = \frac{1}{K} \sum_{i=1}^{K} D_i$ $D_i = \max_{i \neq j} R_{i,j}$ $R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$	-$M_{i,j}$ be the separation between the ith and the jth cluster; -S_i: within-cluster scatter for cluster i; -K: number of clusters	Minimum

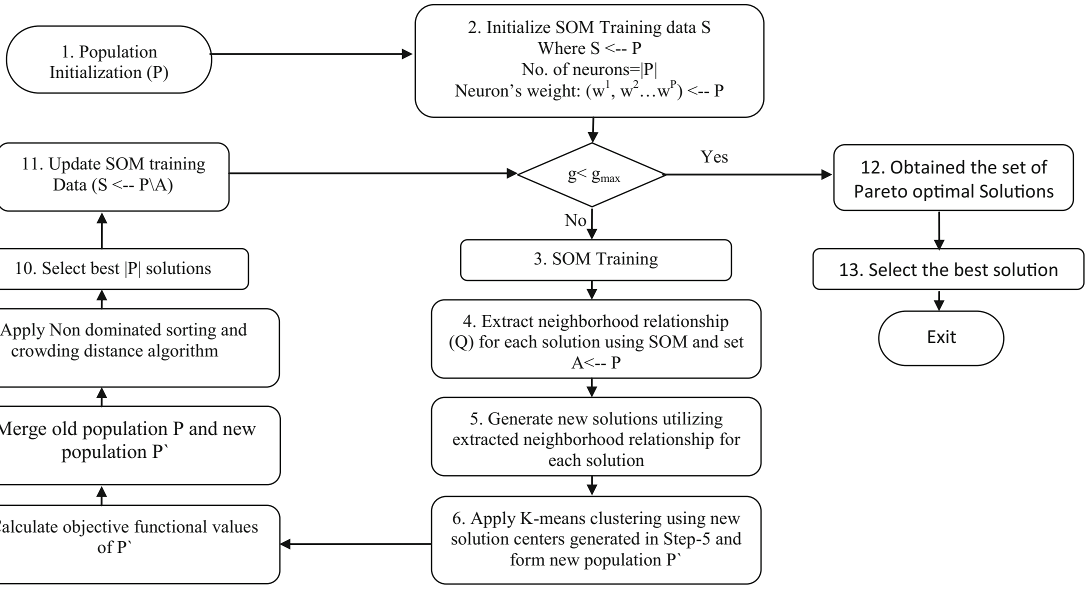

1. Population Initialization (P)
2. Initialize SOM Training data S Where S <-- P No. of neurons=|P| Neuron's weight: (w1, w2…wP) <-- P
g< gmax
Yes
No
12. Obtained the set of Pareto optimal Solutions
13. Select the best solution
Exit
3. SOM Training
4. Extract neighborhood relationship (Q) for each solution using SOM and set A<-- P
5. Generate new solutions utilizing extracted neighborhood relationship for each solution
6. Apply K-means clustering using new solution centers generated in Step-5 and form new population P`
7. Calculate objective functional values of P`
8. Merge old population P and new population P`
9. Apply Non dominated sorting and crowding distance algorithm
10. Select best |P| solutions
11. Update SOM training Data (S <-- P\A)

Fig. 2 Flow chart of proposed algorithm for automatic multi-objective document clustering. Here, P: population containing solutions, $|P|$: size of the population, w^i: weight vector of ith neuron, g_{max}: maximum number of generations, A: archive (copy of population P), Q: Mating pool

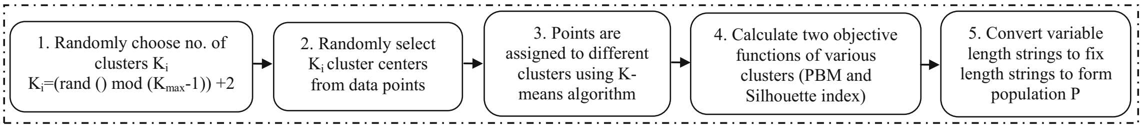

1. Randomly choose no. of clusters K_i
K_i=(rand () mod (K_{max}-1)) +2
2. Randomly select K_i cluster centers from data points
3. Points are assigned to different clusters using K-means algorithm
4. Calculate two objective functions of various clusters (PBM and Silhouette index)
5. Convert variable length strings to fix length strings to form population P

Fig. 3 Steps of population initialization

Calculation of Euclidean Distance and Neuron's Weight Updation

To learn the distribution pattern of the population and to find the neighborhood relationship among these solutions, SOM is utilized in our approach. It is trained using the solutions in the population. As the lengths of different solutions are same in the population after padding zeros between "0" and $(K \times F - 2 \times F)$, therefore, during Euclidean distance calculation between input solution and neuron's weight vector, only minimum number of features available in both the vectors are considered.

Example Let $F = 2$ and the maximum length of the solution be 8 for N=16. Consider a vector be $\{(m, n, q, p, 0, 0, 0, 0)\}$ having $K_1 = 2$ and second vector be $\{(w, x, y, z, a, b, 0, 0)\}$ having $K_2 = 3$. Then during distance calculation or weight updation, only $\{min(K_1, K_2) * F\}$ number of features are considered and other features are ignored.

Objective Functions Used

Proposed clustering framework follows the concepts of multi-objective optimization which is capable of optimizing more than one objective functions (cluster validity measures) simultaneously. In order to measure the goodness of the partitioning encoded in a solution, two internal cluster validity indices, Pakhira-Bandyopadhyay-Maulik (PBM) index [47] and Silhouette index (SI) [53, 58] are calculated and those are used as the objective functions of the current solution. Note that these two objective functions measure separation and compactness between the partitionings in two different ways. The superiority of PBM index over other cluster validity indices, namely, Dunn index [44], Davies−Bouldin index [17] and Xie–Beni index [51] in determining the appropriateness of clusters is established in [47]. While in [3], Silhouette index is compared with 29 other cluster validity measures (excluding PBM index) namely Davies−Bouldin index [17], Gamma index, C index, Dunn index [44], Xie–Beni index [51] etc. and it was found that Silhouette index achieved highest success rate compared to others. Inspired by these existing literature, PBM Index and Silhouette index are incorporated in our proposed framework as the objective functions. Formal definitions of these objective functions are available in Table 1.

Extracting Closer Solutions using Neighborhood Relationship of SOM

The nearby solutions for the current solution are identified using neighborhood relationship (NR) of SOM which is trained using the solutions in the population. This set of nearby solutions form the mating pool, Q, for the current solution. Only these solutions can take part in mating to generate a new solution from the current solution. Series of steps to construct the mating pool, Q, for the $\overline{x_{current}} \in P$ are described in Algorithm 2 [55]. Firstly winning neuron "b" for the current solution needs to be selected (Line 1). Thereafter neighboring neurons near to "b" and the corresponding mapping solutions ∈ P are extracted to form the mating pool (Line 2). The neighboring (closer) solutions present in the mating pool for the current solution can take part in the reproduction operation to generate a new solution. Different parameters used in the algorithm are- P: the population containing solutions $(\overline{x^1}, \overline{x^2}, \ldots, \overline{x^{|P|}})$, γ: threshold probability for selecting the neighboring solution, D: distance matrix formed using position vectors of neurons in the grid, H: mating pool size and $\overline{x_{current}}$: current solution for which the mating pool is generated.

```
Algorithm 2 Construct_MatingPool($\overline{x_{current}}$, $\gamma$, P, H, D)
1: Find the winning neuron 'b' in SOM architecture
   corresponding to solution $\overline{x_{current}}$ based on minimum
   Euclidean distance.
2: Sort $b$th row of D in ascending order and store the sorted
   indices in J.
   $Q = \begin{cases} \cup_{m=1}^{H}\{\overline{x^k}\} & \text{if rand()} < \gamma, \text{and m} < H \\ P & \text{otherwise} \end{cases}$
   Where rand() gives a random number lying between 0
   and 1. $x^k$ is the solution ∈ P mapped to neuron $k$ and
   $k$ ∈ J .
3: return mating pool $Q$ for solution $\overline{x_{current}}$
```

Example Let us assume that we have to generate a new solution for the current solution, $\vec{x}_{current}$. Firstly a mating pool is required to be constructed. The number of neurons in SOM grid are 8 having index values {0, 1, 2, 3, 4, 5, 6, 7} with position vectors {(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)}, respectively. To build the mating pool, firstly the winning neuron corresponding to $\vec{x}_{current}$ is determined using the shortest Euclidean distance criterion. Let it be the $4th$ neuron. Secondly, the Euclidean distances between $4th$ neuron and other neurons are required to be calculated using position vectors of the neurons, which are [1.414, 1, 1.414, 1, 0, 1, 1.414, 1, 1.414] (with respect to neuron indices {0, 1, 2, 3, 4, 5, 6, 7}). After that the calculated distances are sorted in ascending order and correspondingly neuron indices are recorded, i.e., after sorting we obtain the list of distances as [0, 1, 1, 1, 1.414, 1.414, 1.414, 1.414] with corresponding neuron index values as J=[4, 1, 3, 5, 0, 2, 6, 7]. Consider the mating pool size (H) as 4. Now a random probability "r" is generated. If "r" is less than some threshold probability, γ, then solutions mapped to H neurons having indices [1, 3, 5, 0] will form the mating pool. This further helps in exploitation. Note that here we have excluded first neuron index in the sorted list as it represents the winning neuron and distance of winning neuron with itself will always be zero. If "r" is greater than some threshold probability, γ, then all solutions in the population will form the mating pool. This step helps in the exploration of the search space to find the optimal solution. In our approach, it is assumed that each neuron should map to one solution so that similar input samples can be nearby to each other.

Offspring Reproduction

In the previous step, the mating pool was constructed which can take part in crossover and mutation operations to generate a new solution. The detailed algorithm for generation of the new solution is shown in Algorithm 3. First, the crossover operator of differential evolution (DE) [49, 55] is used to generate the trial solution (Line 2) and then a repair mechanism is adapted to ensure the feasibility of the generated solution (Line 3). The lower and upper boundaries of the solutions present in a population are utilized in converting a solution into a feasible one. Finally, mutation operation is applied to that solution (Line 4). Some modifications are incorporated in DE algorithm. Firstly during trial solution generation $\overline{y\,'}$, only $\{K_{x_{current}} * F\}$ feature values of the current solution are considered for distance computation while others are treated as zero, where $K_{x_{current}}$ is the number of clusters for the current solution, F is the number of features in the data set. Trial solution generation process is shown in Fig. 4. Secondly, instead of a single mutation operator, three types of mutation operations are used which are—normal mutation (here polynomial mutation [19] is used as normal mutation), insert mutation and delete mutation. Polynomial mutation operator is used in generating a highly disruptive mutated vector to explore the search space in any direction. This further assists in converging towards an optimal set of cluster centers.

Algorithm 3 y=Generate(Q, CR, MP, $\overline{x_{current}}$, $K_{x_{current}}$)

```
1: Randomly select two solutions $\overline{x^1}$ and $\overline{x^2}$ as the parent solutions from the mating pool Q of current solution $\overline{x_{current}}$ in such a way that $\overline{x^1} \neq \overline{x^2} \neq \overline{x_{current}}$
2: Generate a trial solution $\overline{y\,'}$ as per the following equation
     if rand() $\leq CR$, then $y_i^{'} = x_{current_i} + F_1 \times (x_i^1 - x_i^2)$,
     Otherwise $y_i^{'} = x_{current_i}$
   $i = 1\ldots, nz$. Here $nz = (K_{x_{current}} \times F)$ and only $nz$ feature values of $\overline{x_{current}}$ are considered with corresponding $nz$ values of $\overline{x^1}$ and $\overline{x^2}$ during computation while keeping remaining values from ($nz$+1) to $n$ (length of solution) unchanged.
3: Repair the trial solution using the lower ($x_L$) and upper ($x_U$) boundaries of population to generate $\overline{y\,''}$
     if $y_i^{'} < x_{L_i}$, then $y_i'' = x_{L_i}$, elseif $y_i^{'} > x_{U_i}$,
     then $y_i'' = x_{U_i}$, Otherwise , $y_i'' = y_i^{'}$
   where, $i = 1, 2, \ldots, nz$.
4: Now, trial solution is mutated to generate $\overline{y}$ using the following equation. Here $i = (1, 2, \ldots nz)$
     (i)   if $0 \leq MP < 0.6$:
             if rand()$\leq p_m$, then $y_i = y_i'' + \Delta_i \times (x_{U_i} - x_{L_i})$,
             otherwise $y_i = y_i''$
           where $r_1 = rand()$ is a random number generated between 0 and 1, and
           $\Delta_i = \begin{cases} \left[2r_1 + (1-2r_1)(\frac{x_{U_i} - y_i''}{x_{U_i} - x_{L_i}})^{\eta_m+1}\right]^{\frac{1}{\eta_m+1}} - 1, & \text{if } r_1 < 0.5 \\ 1 - \left[2 - 2r_1 + (1-2r_1)(\frac{y_i'' - x_{L_i}}{x_{U_i} - x_{L_i}})^{\eta_m+1}\right]^{\frac{1}{\eta_m+1}} & \text{Otherwise} \end{cases}$
     (ii)  if $0.6 \leq MP < 0.8$:
           A random input pattern is picked from the dataset and that pattern is added to solution $\overline{y}$ starting from position $(nz + 1)$.
     (iii) if $0.8 \leq MP \leq 1.0$:
           Select last cluster center and then delete that from solution $\overline{y}$.
5: **return** solution $\overline{y}$
```

Use of different types of mutation operators aids in locating the appropriate number of clusters and the appropriate partitioning efficiently. Any of these mutation operations can be selected based on probability MP which is generated with a uniform distribution lying in a range [0, 1] as similar to Ref. [51]. If $MP < 0.6$ then normal mutation is selected, else if $0.6 \leq MP < 0.8$ then insert-mutation is adopted, else deletion mutation is applied. Details about these mutation operations are discussed in

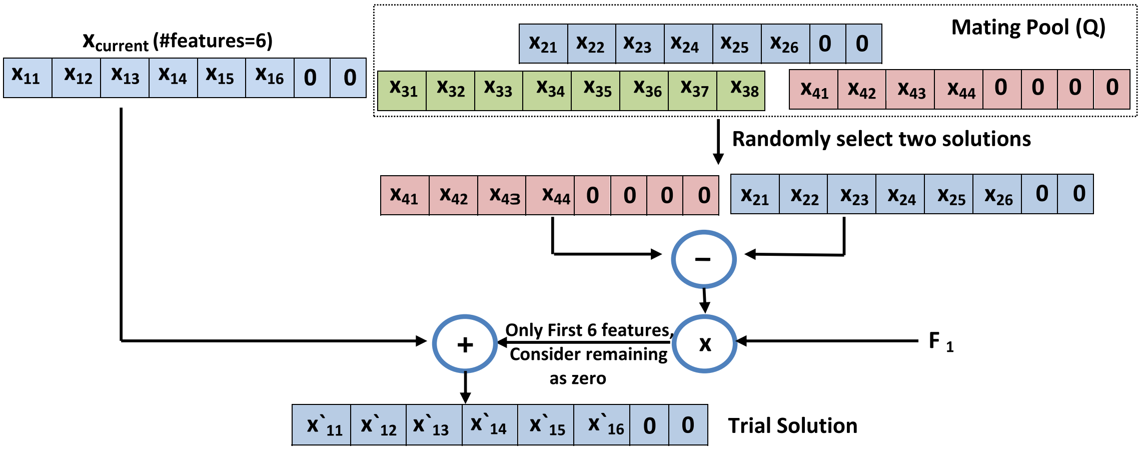

x_current (#features=6)
x11 x12 x13 x14 x15 x16 0 0
Mating Pool (Q)
x21 x22 x23 x24 x25 x26 0 0
x31 x32 x33 x34 x35 x36 x37 x38
x41 x42 x43 x44 0 0 0 0
Randomly select two solutions
x41 x42 x43 x44 0 0 0 0
x21 x22 x23 x24 x25 x26 0 0
–
x
F 1
+
Only First 6 features, Consider remaining as zero
x`11 x`12 x`13 x`14 x`15 x`16 0 0
Trial Solution

Fig. 4 Generation of trial solution

Line-4 of Algorithm 3 and examples of these different types of mutation operations are shown in Fig. 5.

It should be noted that in case of (a) normal mutation, the number of clusters for new solution $\overline{y}$ will remain same as $K_{x_{current}}$, i.e., $K_y = K_{x_{current}}$. (b) insertion mutation: number of clusters for new solution increases by 1, i.e., $K_y = \{K_{x_{current}} + 1\}$. (c) delete mutation: number of clusters for new solution decreases by 1, i.e., $K_y = \{K_{x_{current}} - 1\}$. After generating the new solution, the following additional steps are required to be applied to obtain the final solution.

1. The steps of K-means clustering algorithm are applied to the new solution generated using Algorithm 3. The centers present in the new solution will be considered as the initial set of cluster centers before application of K-means algorithm.
2. Cluster centers obtained after execution of the K-means algorithm are encoded into the new solution. Next, PBM and SI index values are calculated as the objective functions.

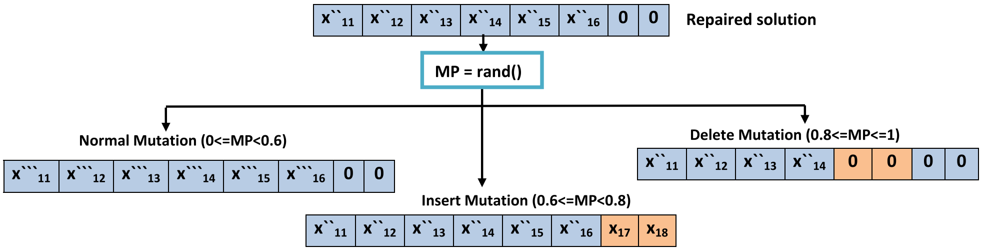
The following symbols are used in the algorithms : (a) F_1 and CR (crossover probability) which are control parameters of DE. The ranges for F_1 and CR are [0, 2] and [0, 1], respectively. (b) p_m is the normal mutation probability for each component of a solution; MP is the current solution ($\overline{x_{current}}$)'s mutation probability and it decides the type of mutation to be performed, η_m denotes the distribution index of polynomial mutation. Note that higher the distribution index, more diverse is the generated solution.

Example Let F=2, $\overline{x_{current}} = \{x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, 0, 0\}$, $K_{x_{current}} = 3$ and Q (Mating Pool) consists of three solutions which are $\{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, 0, 0\}$, $\{x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}, x_{38}\}$, $\{x_{41}, x_{42}, x_{43}, x_{44}, 0, 0, 0, 0\}$. Then at the time of generating a trial solution y' (Step-2), only $K_{x_{current}} \times F = 3 \times 2 = 6$ features of all the solutions are considered as the current solution has only 6 features. The remaining features are treated as zero as shown in Fig. 4. To make the solution feasible, trail solution undergoes repairing using the lower and upper boundaries of the population and then mutation is applied based on some random probability, MP, as shown in Fig. 5.

Selection Operation

In "Offspring Reproduction," after generating offspring (new solution) for each solution in the population P, a new population P' is formed. This is further merged with the old population, P. As $|P|$=$|P'|$, size of the merge population will be $2 \times |P|$. In the next generation, only best $|P|$ solutions (in terms of diversity and convergence [20]) of the merged population are retained, while the rest of the solutions are discarded. This operation is performed using non-dominated sorting and crowding distance algorithm of

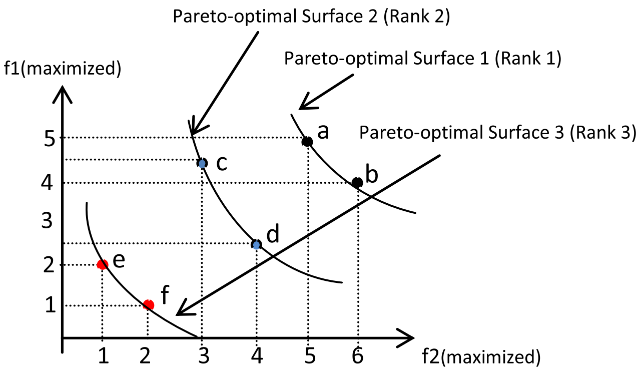
Fig. 5 Generation of new solution. Here rand() is a function which generates some random number between 0 to 1


x``11 x``12 x``13 x``14 x``15 x``16 0 0
Repaired solution
MP = rand()
Normal Mutation (0<=MP<0.6)
x```11 x```12 x```13 x```14 x```15 x```16 0 0
Insert Mutation (0.6<=MP<0.8)
x``11 x``12 x``13 x``14 x``15 x``16 x17 x18
Delete Mutation (0.8<=MP<=1)
x``11 x``12 x``13 x``14 0 0 0 0

the Non-dominated sorting genetic algorithm (NSGA-II) [20].

1. Non-dominated sorting algorithm: It sorts the solutions based on the concepts of domination and non-domination relationships in the objective functional space and ranks the solutions. It divides the solutions into k-fronts, $F = \{Front_1, Front_2 ... Front_k\}$ such that $Front_1$ contains higher ranked solutions and $Front_k$ contains lower ranked solutions. Each front contains a set of non-dominated solutions. For example, in Fig. 6, solutions are ranked as shown in the Pareto-optimal front (or surface). After this step, top ranked solutions are selected and those are added to the population to proceed for the next generation. This process is continued until the number of solutions added equals to $|P|$. If the number of solutions added exceeds $|P|$, then crowding distance algorithm is applied to select the required number of solutions.
2. Crowding distance algorithm: The crowding distance cd_i of i^{th} solution in a $Front_k$ is computed as follows:
   (a) for $i = 1, 2 ...... |Front_k|$, initialize $cd_i = 0$
   (b) For each objective function f_m, $m = 1, 2 ... M$, do the following:
      i. Sort the set $Front_k$ according to f_m in ascending order.
      ii. Set $cd_1 = cd_{|Front_k|} = \infty$
      iii. for $j = 2$ to $(|Front_k| - 1)$, set $cd_j = cd_j + (f_m(j+1) - f_m(j-1)/(f_m^{max} - f_m^{min})$

   Where f_m^{max} and f_m^{min} are the maximum and minimum m^{th} objective functional values, respectively, M is the total number of objective functions.


Pareto-optimal Surface 2 (Rank 2)
f1(maximized)
Pareto-optimal Surface 1 (Rank 1)
Pareto-optimal Surface 3 (Rank 3)
5
4
3
2
1
a
b
c
d
e
f
1
2
3
4
5
6
f2(maximized)

Rank 1: Solutions *a* and *b* are non-dominating to each other because in terms of objective *f1*, solution *a* is better. While in terms of *f2*, solution *b* is better.
Rank 2: Solutions *c* and *d* are non-dominating but dominated by at least one solution of Rank 1 solutions. For example, here, solution *c* dominated by solution *a* because in terms of *f1* and *f2*, solution *a* is better than *c*.
Rank 3: Solutions *e* and *f* are non-dominating but dominated by at least one solution of Rank 1 and Rank 2 solutions. For example, here, solution *e* is dominated by solution *c* and *a*.

Fig. 6 Representation of dominated and non-dominated solutions.

Example Let $|P|$=3 and the two objective functional values are (1, 2), (4, 2.5), (3, 4.5) for solutions e, d and c, respectively. After generating 3 new solutions f, a and b, let their objective functional values be (2, 1), (5, 5), (6, 4) respectively. Suppose both the objective functions have to be maximized. After merging, total number of solutions will become 6 and for next generation, 3 solutions have to be selected. First these solutions are ranked based on dominance and non-dominance concept. Thus, ranked solutions are $\{(5, 5), (6, 4)\}$ for rank-1; $\{(3, 4.5) , (4, 2.5)\}$ for rank-2 and $\{(1, 2), (2, 1)\}$ for rank-3. As rank-1 includes two solutions, therefore they will be propagated to the next generation. Out of all rank-2 solutions, $(3 - 2) = 1$ solution needs to be included in the next generation. Therefore, to select $(3 - 2) = 1$ solution, crowding distance operator is applied to rank-2 solutions and thus $(3 - 2) = 1$ solution is selected having highest crowding distance.

Termination Condition

The process of generating new solutions and then selection of best $|P|$ solutions for next generation will continue until a maximum number of generations, g_{max}, is reached. The final Pareto optimal set contains a set of optimal solutions.

Selection of a Single Solution Based on User Requirement

Any multi-objective algorithm produces a large number of equally important (called as non-dominated) solutions on the final Pareto optimal front. All these solutions represent different ways of clustering the given data set. But sometimes decision-maker wants to select only a single solution based on his requirement or to report the performance of the algorithm. Therefore, in this paper to select a single solution from the Pareto optimal front, we have used some internal cluster validity indices. Two experiments are conducted. In the first experiment, *Dunn Index (DI)* [44] is used to select the single solution from the final Pareto front. Definition of Dunn Index suggests that a higher value indicates better partitioning. Thus we have calculated the DI values for all the partitioning solutions present on the final Pareto front and the solution having the highest value of DI is reported here. Formal description of Dunn index is given in Table 1. In another experiment, Davies-bouldin index (DB) [17] is utilized for selecting a single solution. DB-index value should be minimized for

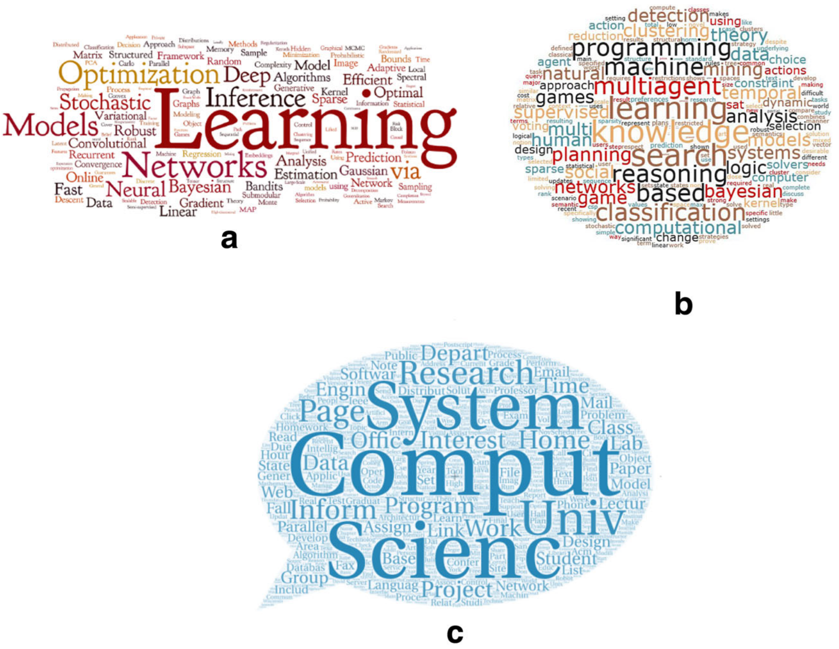


Fig. 7 Word Cloud of **a** *NIPS 2015*; **b** *AAAI 2013*; **c** *WebKB* datasets

getting the optimal partitioning. Thus, we have reported that solution which corresponds to the minimum value of DB-index. Selection of the best solution is shown in step-13 of Fig. 2. This step is different from step-10 which shows that after merging old P and the new population P', only those solutions are selected for the next generation which are non-dominated to each other and are well-distributed over different fronts.

Data Sets Used

In order to show the efficacy of the proposed algorithm, we have chosen the problem of clustering of *Scientific Articles* [14] which is a type of natural language processing task. Researchers submit their articles to different conferences/journals. After that, it is essential to cluster the documents into some groups based on their contents/research topics. This can help the editor to decide about the reviewers. Some conferences/journals ask about the general keywords to decide about the reviewers during submission. But if authors have not selected the keywords correctly, the approach can fail. The current work provides an alternative solution by forming the partitioning of similar journal articles based on their topics/contents. In order to represent an article in the form of a vector, different encoding schemas like tf [43], tf-idf [43], word2vec [39, 45, 61], glove [48] are employed. These scientific articles consist of Title, Abstract, Keywords etc.

In order to show the efficacy of the proposed clustering technique in handling some other type of documents, a data set containing some web-documents is also considered during experimentation. Detailed descriptions of the data sets used in the current study are given below:

NIPS 2015

This data set is taken from kaggle site.[1] This contains 403 articles published in Neural Information Processing Systems (NIPS) conference which is an important core ranked conference in the machine learning domain. It has topics ranging from deep learning, computer vision to cognitive science and reinforcement learning. This dataset includes paper id, title of the paper, event type (poster/oral/spotlight presentation), name of the pdf file, abstract, paper text; out of which only, title, abstract and paper text are used during our experimentation. Here, most of the articles are related to machine learning and natural language processing. The corresponding word cloud is shown in Fig. 7a.

AAAI 2013

This data set is taken from UCI repository [41] which contains 150 accepted articles from another core ranked conference of AI domain, namely AAAI 2013. Each of the papers is having the following information: title of the paper, topics (author-selected low-level keywords from conference-provided list), keywords (author-generated keywords), abstract and high-level keywords (author-selected high-level keywords from conference-provided

[1] https://www.kaggle.com/benhamner/exploring-the-nips-2015-papers/data

list). Most of the articles are related to artificial intelligence like multiagent system, reasoning and machine learning like data mining, knowledge discovery etc. The corresponding word cloud is shown in Fig. 7b.

WebKB

In order to show the potentiality of our approach, we have also used the out-domain dataset such as WebKB, in which documents are web pages, not scientific articles. WebKB [13] data set is consisting of web pages collected from computer science departments of 4 different universities, which are Texas, Cornell, Wisconsin, and Washington. In this paper, we have used total of 2803 documents out of 4199 documents. The corresponding word cloud is shown in Fig. 7c.

Comparing Methods

In order to illustrate the efficacy of the proposed clustering technique, SMODoc_clust, results are compared with several existing clustering techniques having different complexity levels. The approaches we have selected for comparison are traditional clustering techniques like K-means [31], single-linkage [31], SOGA (Single Objective Genetic Algorithm) based clustering [7], MOO-based clustering approach namely, MODoc_clust without using SOM based reproduction operator, MOCK [28], AMOSA based multi-objective clustering technique, VAMOSA [51] and NSGA-II based multi-objective clustering technique [9]. K-means and single-linkage clustering algorithms are some simple and well-known clustering algorithms having limited computational complexity and they assume that the number of clusters present in a data set is known beforehand. Note that our proposed clustering technique is automatic in nature. It determines the number of clusters automatically from a given data set. For K-means and single linkage clustering algorithms, the number of clusters is fixed to K where K is the value of the optimal number of clusters determined by the proposed approach, SMODoc_clust.

MODoc_clust

MODoc_clust, multi-objective based evolutionary algorithm for document clustering, is developed similar to our proposed clustering approach without utilizing the SOM-based genetic operators. It is also able to detect the appropriate number of clusters automatically from a given data set and optimizes PBM [47] and Silhouette index [53], simultaneously. Normal DE-based genetic operators are used during the clustering process. It is developed to show the effectiveness of our newly designed genetic operators utilizing SOM-based neighborhood information.

MOCK

MOCK [28] is a multi-objective clustering algorithm with automatic K-determination and it optimizes two objective functions (compactness and connectedness) simultaneously, where K is the number of clusters. Note that here we have executed MOCK with those document representations for which our proposed approach attains good results.

VAMOSA

VAMOSA [51] is a multi-objective clustering technique which optimizes cluster quality by utilizing two cluster validity indices as the objective functions, namely, PBM Index and Xie-Beni index. It is also able to determine the number of clusters, K, in an automated manner. Here, K lies between [2, $\sqrt{N}$], N is the number of data points. It uses AMOSA [10] as the underlying optimization technique, which was developed inspired by annealing behavior of metals. In original VAMOSA, a point symmetry based distance was utilized for assigning data samples to different clusters. As computation of point symmetry-based distance is a time-consuming task, and also to make a fair comparison with other approaches used in the current study, we have used Euclidean distance in VAMOSA for the purpose of distance computation.

NSGA-II-Clust

NSGA-II-Clust [9, 23] is a multi-objective clustering technique similar to VAMOSA [51] which optimizes PBM-index and Silhouette-index, simultaneously, to determine clusters having good quality in an automated way. It is also capable of determining the number of clusters, K, without human participation. The value of K varies between [2, $\sqrt{N}$], N is the number of data points. It uses NSGA-II [20] as the underlying optimization strategy. In [9], this algorithm was successfully applied to solve image segmentation problems.

SOGA

SOGA [7] is a single objective clustering technique utilizing the search capabilities of genetic algorithm (GA). GA is utilized in optimizing a single cluster validity index. In our experiments, SOGA-based clustering was executed multiple times with the number of clusters varying between 2 to $\sqrt{N}$, where N is the number of articles/documents. The final partitioning is selected based on the maximum value of Dunn index as well as the minimum value of Davies-Bouldin index.

K-means

K-means [31] is a well-known unsupervised clustering algorithm. It assumes that the number of clusters (K) is known *apriori*. Here the given dataset is partitioned into K clusters by using the procedure of minimum center distance-based criterion. A particular point is allocated to that cluster with respect to which it is having the minimum distance.

Single-linkage

Single-linkage clustering [31] is a type of hierarchical clustering technique, whose objective is to build a hierarchy of clusters. Hierarchical clustering techniques can be further divided into agglomerative and divisive algorithms corresponding to bottom-up and top-down strategies to build some clustering trees. In our experiment, agglomerative single linkage clustering algorithm is used.

Experimental Setup and Results

This section presents the evaluation and comparison of proposed approach with other state of the art techniques. In addition, this section also discusses about various preprocessing steps applied, different representation schemas used to convert a document into a vector form, parameter settings followed by discussion of results. Final clustering solution is determined as per steps discussed in "Selection of a Single Solution Based on User Requirement". The results reported in this section are the average values over 20 runs. All the approaches were implemented on a Intel Core i7 CPU 3.60 GHz with 4 GB of RAM on Ubuntu. Various preprocessing steps employed to clean the data sets are explained below:

Preprocessing

In order to clean the text data corresponding to these scientific articles and web-documents, we have executed several preprocessing steps including stop word removal[2] (e.g., is, am are etc.), removal of special characters (like @, ! etc.), punctuation symbols, numbers and white spaces, removal of words having length less than three, lower case conversion (like Computer to computer) and stemming[3] [36]. Stemming [36] is the process of converting inflected words into their morphological base forms called word stems, base or root forms. Reason for performing stemming is to group together the inflected forms of a word so that they can be analyzed as a single item and can help in clustering of documents. In addition to these preprocessing steps, words which appear in less than 5% and in more than 95% articles are removed. Moreover, for NIPS dataset, we have considered title, abstract and paper texts as the attributes for the given papers. For that purpose, topmost 5, 30, and 150 words are selected from title, abstract and paper text, respectively, which make vocabulary size as 183. While in case of AAAI 2013 data set, all the attributes are used. This makes vocabulary size as 673. For WebKB datasets, preprocessed text documents are already available in [13] having total vocabulary of size 7229.

Representation Schemas Used

To represent the scientific/web articles in vector forms, tf (bag-of-word model using 1-gram) [43], tf-idf [43] and most popular representation schema, word2vec [39, 45, 61] and Glove [48] both with varying dimensions of 50, 100, 200, 300 are used in the current study. Note that article vector is obtained by averaging the word2vec/Glove representations of all the vocabulary words present in the article.

Term-frequency or Term-document Count (tf)

Term-document count [43] is a type of representation for representing text documents or any object in the form of real vectors in which each component corresponds to the frequency of occurrence of a particular word (called as the weight of word) in the document. It is denoted as $tf_{t,d}$, the number of times term "t" appears in document "d").

Example Let two documents contain the following texts:

Doc1: John likes to watch movies. Mary likes movies too.
Doc2: John likes to watch football games.

Here vocabulary comprises of list of words (excluding stop words and ".") like: <John, likes, watch, movies, Mary, football, games>. Now document vector is represented as:

Doc1: $< 1, 2, 1, 2, 1, 0, 0 >$
Doc2: $< 1, 1, 1, 0, 0, 1, 1 >$

tf-idf

tf-idf [24] pair is another well known scheme for weighting the terms in a document by utilizing the concept of vector space model [43]. After assigning tf-idf weight to each term, document vector "v" of a document "d" can be represented as

$$v_d = [w_{1d}, w_{2d}, w_{3d}, \ldots\ldots, w_{nd}] \quad (1)$$

[2] We have used python nltk toolkit [42] to remove the stop words which are 153 in numbers.

[3] Here SnowballStemmer [42] of nltk is used.

where

$$w_{t,d} = tf_{t,d}.\left(1 + \log \frac{1 + |D|}{1 + \{d' \in D | t \in d'\}}\right) \quad (2)$$

and

- $tf_{t,d}$ is the term frequency of term t in document d in normalized form;
- $\log \frac{1+|D|}{1+\{d' \in D|t \in d'\}} + 1$ is the inverse document frequency. $|D|$ is the total number of documents in collection and $\{d' \in D|t \in d'\}$ is the number of documents containing term t. Here 1 is added in the numerator and in the denominator to avoid division by zero error.

Example Consider a document consisting of 300 words where the word *cat* appears 5 times. The term frequency (i.e., "tf") for *cat* is (5/300) $= 0.016$ (Using "l1" normalization). Now, assume that we have 20 million documents (D) and the word *cat* appears in two thousand (df) of D documents. Then, $idf = 1 + log(20,000,001/2,001) = 3.99$. Thus, the tf-idf weight of the term *cat* is: $0.03 * 3.99 = 0.119$. Similarly, document vector can be generated corresponding to vocabulary <John,likes, watch, movies, Mary, football, games> as given below

Doc1: $< 0.12, 0.24, 0.12, 0.34, 0.17, 0, 0, >$
Doc2: $< 0.17, 0.17, 0.17, 0, 0, 0.24, 0.24 >$

Word2vec

Word2vec [39, 45, 61] is a model that is used to generate word embeddings. It effectively captures the semantic properties of the words in the corpus. Here, we have used gensim tool to generate word vectors of varying dimensions. To generate the article vector, word vectors of different words in the article are averaged.

Glove

Glove [48] provides vector-representations of words similar to word2vec. Glove learns by constructing a co-occurrence matrix (words X context) that basically counts how frequently a word appears in a context and then this matrix is reduced to lower dimension, where, each row represents a word vector. Different dimensions (50, 100, 200, and 300) of Glove word2vec are used for our experiment which are already available at https://github.com/stanfordnlp/GloVe. Note that for 50, 100, and 200 dimensions, pre-trained glove word vectors have 400K vocabulary, while for 300 dimension, size of vocabulary is $2.2M$. To generate the article vector, word vector averaging is used similar to word2vec representation.

Parameter Setting

MOCK [28] and SOGA [7] are executed with default parameters (codes provided by authors). Parameter settings of other algorithms are explained below.

1. SMODoc_clust and MODoc_clust: Different parameter values used in our proposed clustering technique are shown in Table 2. These parameters are selected after conducting a thorough sensitivity study. It is important to note that mutation (normal, deletion and insertion) probabilities used here are same as reported in the existing literature [7, 8, 10]. Same parameters are used in MODoc_clust approach (excluding SOM parameters).
2. VAMOSA: This algorithm is executed with Tmax =10, Tmin=0.01, SL=20, HL =10. Here, Tmax and Tmin denote the maximum and minimum values of temperature, respectively. SL and HL are two parameters associated with the size of the archive. They denote the soft-limit and hard-limit on the archive size, respectively. Initially archive of AMOSA is initialized with SL number of solutions. During the process, number of solutions in the archive can be reached upto SL. Once number of solutions crosses the threshold SL, clustering procedure is applied to reduce this to HL. At the end of the execution, an archive having HL number of solutions is provided to the user. Rest of the parameter values are kept similar as reported in [51].
3. NSGA-II-clust: Different parameters used in the NSGA-II based multi-objective clustering are: number of generations=50, population size=50, crossover probability=0.8, mutation strength=0.2, normal (μ_n), insertion (μ_i) and deletion (μ_d) mutation probabilities are taken as: $\mu_n < 0.7$, $0.7 < \mu_i \leq 0.85$ and $\mu_d \geq 0.85$, respectively.

Only for VAMOSA on WebKB dataset, we have varied the range of clusters, K, between 2 to 15.

Table 2 Parameter setting for our proposed approach

Parameters	Values
Maximum number of generations (g_{max})	50
Population size (P)	50
Initial learning rate (η_0)	0.1
Initial neighborhood size (σ_0)	2
Number of training iterations in SOM	|P|
Mating pool size (H)	5
DE control parameters (F_1 and CR)	0.8, 0.8
Normal mutation probability	[0,0.6[
Insertion mutation probability	[0.6 to 0.8[
Deletion mutation probability	[0.8,1[

Table 3 Results obtained after application of the proposed clustering algorithm on text documents in comparison to other clustering algorithms using Dunn index (DI)

Data set	#N	Rep.	#F	SMODoc_clust		MODoc_clust		VAMOSA		NSGA-II-Clust		SOGA		K-means		single-linkage	
				OC	DI	OC	DI	OC	DI	OC	DI	OC	DI	OC	DI	OC	DI
NIPS 2015	403	tf	183	4	*0.2247*	4	0.1082	5	0.1058	2	0.0714	5	0.0471	4	0.0811	4	0.0698
		tf-idf	183	5	*0.1844*	4	0.1623	7	0.1081	2	0.0738	2	0.0832	5	0.1388	5	0.1494
		word2vec	50	4	*0.0732*	5	0.0397	2	0.0366	2	0.0121	4	0.0258	4	0.0268	4	0.0401
			100	2	*0.6414*	6	0.0282	2	0.6121	2	0.0111	2	0.0069	2	0.0059	2	0.0116
			200	2	*0.5657*	8	0.0445	9	0.0292	2	0.0123	2	0.0039	2	0.0090	2	0.0106
			300	2	*0.5723*	8	0.0445	11	0.0252	2	0.1676	3	0.0048	2	0.0058	2	0.0085
		glove	50	5	0.3096	5	0.2953	7	0.2674	2	0.2660	10	0.2900	5	0.2601	5	*0.3124*
			100	5	*0.3884*	4	0.3714	4	0.3533	2	0.3187	8	0.3833	5	0.3103	5	0.3593
			200	4	*0.4104*	2	0.4099	3	0.4097	2	0.3829	8	0.4068	4	0.3753	4	0.3443
			300	4	*0.3778*	4	0.3598	7	0.3669	2	0.3539	4	0.3111	4	0.3647	4	0.3509
AAAI 2013	150	tf	673	4	*0.2948*	4	*0.2948*	4	*0.2948*	2	0.1860	4	0.1328	4	0.1961	4	0.2635
		tf-idf	673	3	*0.5352*	3	0.5286	2	0.5218	2	0.5218	3	0.1431	3	0.4204	3	0.3339
		word2vec	50	9	*0.1805*	11	0.1751	5	0.1665	2	0.1726	10	0.0521	9	0.0692	9	0.0738
			100	5	*0.1238*	4	0.0871	2	0.1290	2	0.0504	7	0.0612	5	0.1110	5	0.0940
			200	5	*0.1168*	4	0.0827	3	0.0401	2	0.0333	2	0.0457	5	0.1094	5	0.1094
			300	9	*0.1513*	11	0.1292	xx	xx	2	0.0334	3	0.0401	9	0.0638	9	0.0763
		glove	50	2	*0.3213*	4	0.3213	5	0.2330	2	0.2513	2	*0.3213*	2	*0.3213*	2	*0.3213*
			100	3	*0.4005*	3	*0.4005*	5	0.2329	2	0.2753	3	0.0	3	0.2433	3	0.2470
			200	3	0.3323	3	*0.3640*	2	0.2461	2	0.2848	2	0.3135	3	0.2588	3	0.2588
			300	4	*0.2346*	3	0.2233	4	0.1338	2	0.1429	2	0.2080	4	0.1578	4	0.2319
WebKB	2803	tf	7229	2	*3.6423*	3	3.1248	3	0.6710	2	0.0069	4	0.0038	2	*3.6423*	2	*3.6423*
		tf-idf	7229	3	*0.9174*	10	0.7450	3	0.5610	2	0.0059	4	0.0012	3	0.9174	3	0.9174
		word2vec	50	4	0.0452	4	0.0452	3	0.0424	2	0.0493	4	0.0308	4	0.0452	4	*0.0480*
			100	4	*0.0474*	4	*0.0474*	5	0.0469	2	0.0463	2	0.0424	4	*0.0474*	4	0.0426
			200	5	0.0464	5	0.0449	2	*0.0985*	2	0.0454	3	0.0	5	0.0461	5	0.0460
			300	2	*0.0646*	5	0.0421	6	0.0461	3	0.0419	3	0.0	2	0.0445	2	*0.0607*
		glove	50	4	*0.5871*	2	0.5637	3	0.0597	2	0.0601	2	0.5129	4	0.0430	4	0.0643
			100	4	*0.6909*	4	0.6189	6	0.0400	2	0.0462	2	0.5780	4	0.0468	4	0.0541
			200	3	0.6107	*3*	*0.6391*	3	0.1613	2	0.0530	2	0.0727	3	0.0640	3	0.0698
			300	4	*0.6325*	4	*0.6325*	6	0.0461	2	0.0621	2	0.0	4	0.0672	4	0.0764

Rep representation, *N* number of scientific articles, *F* vocabulary size, *OC* obtained number of clusters, *DI* Dunn Index; xx: all data points assigned to single cluster. Italic entries indicates the best performance using Dunn index

Analysis of Results Obtained

In order to measure the goodness of the obtained partitionings by MOO-based proposed approach, two internal cluster validity indices, namely *Dunn Index* [44] and *Davies-Bouldin (DB) Index* [17] are used. The number of clusters detected by the proposed algorithm for different datasets are reported in Tables 3 and 4. The maximum and minimum values of Dunn index and DB index, respectively, imply better clustering results. Detailed descriptions of Dunn and DB index are given in Table 1. The most relevant words of different clusters (obtained using Dunn index) corresponding to optimal partitionings identified by the proposed approach for *NIPS 2015* and *AAAI 2013* data sets are shown in Fig. 9a and b, respectively. These keywords are extracted using topic modeling tool named Latent Dirichlet allocation (LDA) [11].

Results on NIPS 2015 Articles

On NIPS 2015 data set, our proposed approach performs better than all other existing approaches with different representation schemas used. Results obtained are shown in Tables 3 and 4. The best result having $DI = 0.64$ was obtained using word2vec model with obtained cluster (OC)=2, where each word vector is of 100 dimensions.

Table 4 Results obtained after application of the proposed clustering algorithm on text documents in comparison to other clustering algorithms using DB index

Data set	#N	Rep.	#F	SMODoc_clust		MODoc_clust		VAMOSA		NSGA-II-Clust		SOGA		K-means		single-linkage	
				OC	DB	OC	DB	OC	DB	OC	DB	OC	DB	OC	DB	OC	DB
NIPS 2015	403	tf	183	3	*0.8171*	3	0.8192	8	1.3949	2	1.8226	3	3.8074	3	1.3051	3	1.5270
		tf-idf	183	4	*0.8909*	4	1.9023	7	1.5161	2	1.6235	2	2.8180	4	1.3454	4	1.4449
		word2vec	50	2	*0.1323*	3	0.1346	4	0.3336	2	1.6002	4	0.5123	2	0.6897	2	0.6898
			100	4	0.4830	4	0.4833	5	0.4965	2	1.9047	5	0.4406	4	0.6415	4	0.6400
			200	3	*0.4420*	3	0.4433	6	0.4937	2	2.0387	2	0.7869	3	0.6073	3	0.5974
			300	3	*0.4424*	3	0.4448	7	0.4625	2	1.8985	3	0.6533	3	0.5950	3	0.5914
		glove	50	3	*1.7339*	4	1.8308	11	2.1762	2	1.4428	3	2.4221	3	2.3080	3	2.6423
			100	4	*1.5774*	3	1.6388	2	2.1357	2	1.6063	3	2.4676	4	2.7221	4	2.5282
			200	4	*1.6561*	4	*1.6561*	3	2.7614	2	2.0814	3	2.1848	4	2.9711	4	2.6400
			300	4	*1.8533*	3	1.8692	2	2.5201	2	1.9119	4	5.6560	4	2.9511	4	2.8510
AAAI 2013	150	tf	673	4	1.4330	3	1.4385	4	*1.1605*	2	1.8727	4	1.8695	4	1.8786	4	1.9064
		tf-idf	673	4	*1.7145*	3	1.7788	2	1.8407	2	1.8929	4	1.8486	4	2.0155	4	1.8986
		word2vec	50	3	0.7356	3	0.9981	5	*0.6382*	2	1.7318	5	1.0032	3	1.0308	3	1.0242
			100	3	*0.7170*	2	0.8773	2	0.8161	1	1.9175	5	1.0271	3	1.0259	3	1.0353
			200	3	*0.7276*	3	0.7452	3	1.0674	2	1.7372	2	1.2772	3	1.0142	3	1.0294
			300	3	*0.6879*	3	0.7054	xx	xx	2	1.7372	3	0.9644	3	0.9885	3	1.0076
		glove	50	3	*1.2799*	4	1.3200	5	1.7573	2	1.3644	3	1.4252	3	1.3475	3	1.4138
			100	4	*1.1374*	3	1.1822	5	1.5257	2	1.3644	3	1.2513	4	1.7296	4	1.6525
			200	4	*1.1970*	4	*1.1970*	2	1.6171	2	2.0304	3	2.2181	4	1.5871	4	1.6124
			300	4	*1.2884*	4	1.4062	4	1.7796	2	1.7294	3	1.6864	4	1.6865	4	1.6291
WebKB	2803	tf	7229	3	0.0206	3	0.0206	3	0.0678	2	6.9621	3	2.6846	3	0.0646	3	0.0646
		tf-idf	7229	3	0.0834	4	0.0497	3	0.0623	2	23.757	4	2.0806	3	2.5467	3	0.0522
		word2vec	50	5	*1.1400*	5	1.1502	3	1.5417	2	2.4978	3	1.8074	5	1.3936	5	1.5454
			100	5	1.1457	5	*1.1448*	4	1.7018	4	2.5136	2	1.6088	5	1.3867	5	1.1367
			200	5	*1.1352*	3	1.1913	2	0.6134	2	2.5136	3	2.5172	5	1.3574	5	1.5183
			300	5	*1.2220*	5	1.2203	6	3.4282	3	2.7561	2	2.5237	5	1.3442	5	1.4609
		glove	50	3	*0.5523*	2	0.8155	3	2.6150	2	2.2373	3	1.9142	3	1.9468	3	2.2323
			100	3	1.4299	2	0.8687	6	3.3422	2	1.9867	2	1.1582	2	2.9522	2	*0.2911*
			200	2	*0.1932*	3	1.3411	6	1.2107	2	2.6978	2	1.4694	2	0.3008	2	0.3008
			300	3	*1.6632*	3	2.9034	6	3.4282	2	2.2490	2	2.0660	3	1.8072	3	2.1201

Rep. representation, *N* number of scientific articles, *F* vocabulary size, *OC* obtained number of clusters, *DB*, Davies-Bouldin Index; xx: all data points assigned to single cluster. Italic entries indicates the best performance using DB index

On the other hand, best value of DB index=0.1323 was obtained using word2vec representation with same number of clusters, i.e., 2, where each word vector is of 50 dimensions. Thus, it can be inferred that optimal value of number of clusters for NIPS datasets is 2. Extracted relevant words for different clusters corresponding to the best result obtained by our approach are shown in Fig. 9a. This clearly indicates that two clusters correspond to the topics of deep learning and computer vision, respectively. Major observations related to the obtained clusters at the fine-grained level are as follows: articles in cluster-2 correspond to *Deep Convolution Neural network* applied on image data. Articles in cluster-1 correspond to simple feed-forward network with stochastic optimization in which features are extracted by the user and those are fed to the network. Pareto optimal solutions obtained after application of our proposed framework are shown in Fig. 8a. Here, we can see that after completion of the maximum number of generations, Pareto optimal front converges to only three to four non-dominated solutions. Each point in the Pareto optimal front of Fig. 8a represents a non-dominated solution. Note that our proposed aproach, SMODoc_clust, attains the best results with word2vec based representation with dimension 100. MOCK is also executed with this configuration. Best result by MOCK corresponds to $DI = 0.0151$ and $DB = 0.6401$ with $OC = 4$. In most of the cases, MODoc_clust, VAMOSA, NSGA-II-Clust, SOGA, K-means, and single-linkage algorithms fail in achieving good scores for this

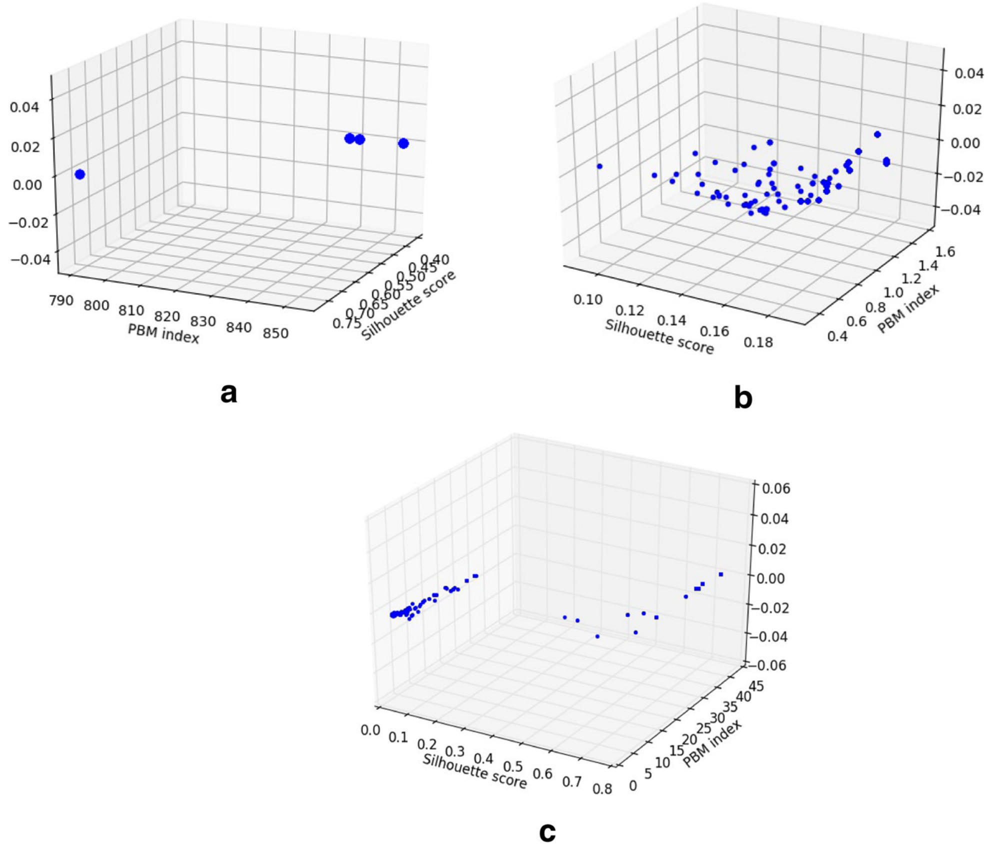
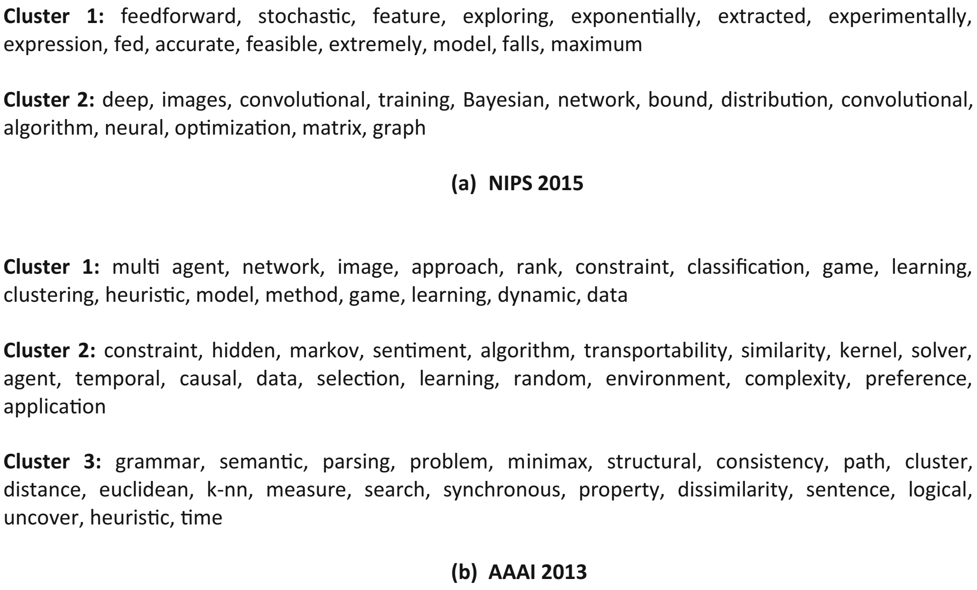


Fig. 8 Pareto optimal fronts obtained after application of the proposed clustering algorithm on scientific articles **a** *NIPS 2015*; **b** *AAAI 2013*; **c** *WebKB* datasets

data set; this clearly shows the utility of incorporating SOM based reproduction operators in the proposed clustering technique. Note that for NIPS 2015 articles, SOGA-based clustering does not converge after fifth generation while using tf and tf-idf based representation schemes. Therefore, for SOGA, the results obtained after the fifth generation are reported in Table 3.

Results on AAAI 2013 Articles

On AAAI 2013 data set, our proposed approach mostly performs better than all other existing approaches utilizing different representation schemes. The best result was obtained using tf-idf representation and the corresponding value of Dunn-index is 0.53 with OC=3. Only with "tf"



Cluster 1: feedforward, stochastic, feature, exploring, exponentially, extracted, experimentally, expression, fed, accurate, feasible, extremely, model, falls, maximum

Cluster 2: deep, images, convolutional, training, Bayesian, network, bound, distribution, convolutional, algorithm, neural, optimization, matrix, graph

(a) NIPS 2015

Cluster 1: multi agent, network, image, approach, rank, constraint, classification, game, learning, clustering, heuristic, model, method, game, learning, dynamic, data

Cluster 2: constraint, hidden, markov, sentiment, algorithm, transportability, similarity, kernel, solver, agent, temporal, causal, data, selection, learning, random, environment, complexity, preference, application

Cluster 3: grammar, semantic, parsing, problem, minimax, structural, consistency, path, cluster, distance, euclidean, k-nn, measure, search, synchronous, property, dissimilarity, sentence, logical, uncover, heuristic, time

(b) AAAI 2013

Fig. 9 Relevant cluster-keywords for **a** NIPS 2015; **b** AAAI 2013 data set corresponding to the best partitioning result obtained by the proposed approach

Table 5 Values of different components of the Dunn Index for tf, tf-idf and Glove representation with 100 dimension on WebKB dataset

Rep.	OC	DI=a/b	a	b
tf	2	3.6423	1010.2593	277.3699
tf-idf	3	0.9174	806.7541	879.386
glove (100)	4	0.6909	4.6481	6.727

Here, Rep. denotes representation, OC: obtained cluster, DI: Dunn Index, a: minimum distance between two points belonging to different clusters, b: maximum diameter amongst different clusters

based representation schema, MODoc_clust works similar to the proposed algorithm. MOCK is also executed with tf-idf based representation. Best solution obtained by MOCK corresponds to $DI = 0.2684$ and $DB = 12.1723$ with $OC = 3$. On the other hand, minimum DB value obtained by our proposed approach is 0.6879 with word2vec based representation scheme having 300 dimensions and the corresponding number of clusters is 3. Thus, we can say that optimal value of number of clusters for AAAI dataset is 3. Similar to NIPS 2015 data set, here also SOGA based clustering does not converge within fifth to eighth generations. Figure 9b clearly indicates the topics of different clusters. All clusters are related to machine learning. But at the lower level of abstraction, we can conclude that cluster-1 contains articles related to artificial intelligence as the words like multi-agent, game, heuristics method, etc. are pre-dominant in this cluster. Cluster-2 corresponds to the papers discussing about different applications of machine learning approaches, for example Hidden Markov Model to Sentiment Analysis and other domains. Cluster-3 precisely corresponds to the papers reporting applications of machine learning approaches like K-nearest neighbor classifiers etc. for solving different natural language processing tasks. These articles discuss about grammar, syntax and semantics, parsing etc. The Pareto optimal solutions obtained by the proposed clustering approach are shown in Fig. 8b. Each point in the Pareto optimal front of Fig. 8b represents a non-dominated solution. Again, MODoc_clust, VAMOSA, NSGA-II-clust, SOGA, K-means and single-linkage algorithms fail to achieve good scores for this data set in most of the cases. Note that our MOO based clustering approach and clustering (constraint based) approach discussed in the paper [46] are different in the sense that our goal is to cluster the scientific articles in an automated way without satisfying any constraint to extract broad areas of different articles. While the goal of the approach proposed in [46] is to extract the fine-grained keywords which can better represent the papers accepted in the conference. For this purpose all the words present in the abstract of the article are taken into consideration with some constraints. Non-dominated solutions on the final Pareto optimal front obtained by the proposed clustering approach are shown in Fig. 8c.

Results on WebKB Dataset

On WebKB data set, our proposed approach, in most of the cases, performs better than all other existing approaches utilizing different representation schemes. Out of different dimensions used in Word2vec based representation, maximum DI value of 0.0474 and minimum DB value of 1.1351 were obtained by our proposed approach using 200 dimensions with OC=5. On the other hand, using Glove representation varying the dimensions, maximum DI value of 0.6909 was obtained with OC=4 and 100 dimensions. Minimum DB value of 0.1932 was obtained using glove representation with 200 dimensions and OC=2. In Table 3, maximum DI value of 3.6423 was obtained with tf representation. After thorough investigation of this result we found that this solution corresponds to a partitioning where more than 80% of the total documents are assigned to a single cluster which in turn increases the compactness and separation of the clusters. This results into high value of Dunn index. This partitioning was generated because of the sparsity in document matrix (containing most of the components as zero in document vector) which is of size 2803 × 7229. Similar situation is happened with tf-idf based representation. The best value of Dunn index obtained is 0.6909 which corresponds to OC=4 with Glove representation having 100 dimensions, whereas the best value of obtained DB index is 0.1932 with OC=2. MOCK attains best DB index value of 7.2509 which is greater than minimum DB value obtained by our approach. In Table 5, values of numerator and denominator of Dunn index corresponding to *tf, tf-idf, glove with 100 dimensions* representations for this dataset are shown. Numerator measures the minimum distance between two points belonging to different clusters, while, denominator measures the maximum diameter amongst diameters of different clusters. It is clearly evident from Table 5 that for tf and tf-idf representations, both numerator and denominator values are too high as compared to Glove (100) representation. This is because generated clusters are not proper/compact; there is a big cluster (containing 80% of data points) and 1 or 2 small clusters. Because of the presence of large-cluster, denominator value is high and cluster separation (numerator) is also high. Thus Dunn-index value is also high. This in turn proves that DI is not a good measure of cluster quality. It prefers to have non-uniform sized clusters. Except the cases of Glove and Word2vec based representations with 100 dimensions, the proposed algorithm always beats other algorithms and attains best result.

Table 6 Results reporting DB index value obtained after application of the proposed clustering algorithm on WebKB documents using Doc2vec representation in comparison to other clustering algorithms

Data set	#N	Rep.	#F	SMODoc_clust		MODoc_clust		VAMOSA		NSGA-II-Clust		SOGA		K-means		single-linkage	
				OC	DB	OC	DB	OC	DB	OC	DB	OC	DB	OC	DB	OC	DB
WebKB	2803	Doc2vec	50	3	2.3204	3	3.0317	3	3.6981	2	3.9696	4	3.3678	3	3.6687	3	4.2620
			100	2	0.9723	2	0.9729	4	5.0457	2	3.8375	2	3.6676	2	3.7273	3	3.9529
			200	2	0.9549	2	1.0054	2	2.6654	2	3.1647	4	3.9685	2	4.0644	2	3.8797
			300	2	0.3217	3	0.8023	5	4.8537	2	2.9372	2	3.2979	2	4.3355	2	3.9873

Rep. representation, *N* number of scientific articles, *F* vocabulary size, *OC* obtained number of clusters, *DB* Davies-Bouldin Index

Generally, with the increase in dimension/size of Word2vec/glove vector representation, precision of capturing semantic information increases. With the increase in size parameters, more data is required to train the models and to represent the concepts.

However, in our work, due to the use of word2vec/glove averaging to represent the articles/documents, there is a loss of semantic information. Therefore, in Table 4, it can be seen that with the increase in the vector length using word2vec/glove, instead of decrease in the DB index values, there are fluctuations in the result. Some more robust representation is required to avoid loss of semantic information as this representation of document plays a key role in defining similarity/dissimilarity metric between documents which in turn can help in clustering documents in an automated way.

Therefore, we have tried Doc2vec[4] representation. Note that we have trained the Doc2vec on available WebKB documents, i.e., 4199 preprocessed documents, which make use of pre-trained glove [48] word embeddings having 2.2M as vocabulary size and 300 dimensional word vector. Results are also reported in Table 6. It can be inferred from the results obtained by SMODoc_clust, MODoc_clust, and NSGA-II-clust techniques (shown in Table 6) that *with the increase in the dimensionality of vector representation, qualities of clusters improve in terms of DB index value (lesser the value, more good is the cluster quality)*. However, in VAMOSA, this is not the case. From these statements, it can be inferred that the quality of clusters not only depends upon the algorithm but also on the type of objective functions (cluster validity indices in our case). In SMODoc_clust, MODoc_clust, and NSGA-II-clust, two objective functions namely, PBM and Silhouette indices are used. While, in VAMOSA, PBM and Xie-beni indices are used. Note that for doc2vec representation, we have not reported Dunn index as it is biased towards non-uniform sized clusters as mentioned in the end of first paragraph of current section.

Non-dominated solutions present on the final Pareto optimal set obtained by the proposed clustering approach are shown in Fig. 8c.

Theoretical Analysis

Possible theoretical reasons behind the success of the proposed clustering technique are analyzed below:

- In general existing multi-objective evolutionary algorithms (MOEAs) utilize the reproduction operators which are popular in single objective optimization (SOO).
- But topologies of optimal solutions are totally different in single (SOO) and multi-objective optimization problems (MOO). In case of SOO the topology of optimal solution is a point and the distribution of optimal solutions in case of MOO follows a regular manifold structure. This proves that reproduction operators which are well-suited for single objective optimization may not perform well for MOO. There is a need to design some new reproduction operators for MOO problems.
- In recent years, researchers have proved that use of simple reproduction operators of SOO in MOO framework leads to the poor performance of MOO for solving complex problems like tackling rotated and complicated MOPs [30, 67].
- Inspired by this, some specific reproduction operators for MOO algorithms are designed in recent years [65, 66]. Here, the topologies of Pareto optimal solutions of MOPs were utilized in designing new reproduction operators. It was shown in [65, 66] that these operators help in better convergence of the proposed MOO based approach.
- Inspired by the above observations, in the current study, topology-inspired reproduction operators are introduced in developing a MOO based clustering framework where several cluster quality measures are simultaneously optimized. The topology is measured with the help of self-organizing map [29, 34].

[4] https://github.com/jhlau/doc2vec

Table 7 *p* values obtained after conducting t-test comparing the performance of proposed SMODoc_clust algorithm with other existing clustering techniques with respect to Dunn index values reported in Table 3

Data Set	Representation	#F	MODoc_clust	VAMOSA	NSGA-II-Clust	SOGA	K-means	single-linkage
NIPS 2015	tf	183	3.01E-192	6.59E-190	7.89E-261	1.96E-307	2.28E-241	5.41E-264
	tf-idf	183	4.13E-011	7.44E-099	3.77E-172	1.09E-104	4.47E-041	1.77E-25
	word2vec	50	1.58E-023	1.24E-027	1.73E-68	5.21E-44	2.26E-042	2.99E-019
		100	0.0	0.0	0.0	0.0	0.0	0.0
		200	2.80E-021	0.0	0.0	0.0	0.0	.00
		300	0.0	0.0	0.0	0.0	0.0	0.0
	glove	50	2.62E-005	9.51E-036	6.59E-038	5.33E-009	1.59E-047	0.2513
		100	4.70E-007	1.31E-025	2.31E-085	0.182621	1.35E-102	3.25E-018
		200	0.911417	0.961362	1.93E-016	0.38863	1.31E-025	3.47E-078
		300	8.99E-008	0.001650	9.009E-13	2.26E-079	0.000127372	8.49E-016
AAAI 2013	tf	673	0.7885	0.788494	2.79E-168	2.82E-283	1.65E-146	1.13E-18
	tf-idf	673	0.0714026	8.69E-005	8.69E-05	0.0	3.72E-181	0.0
	word2vec	50	0.049742	3.49E-06	0.006069	1.46E-213	1.64E-196	1.95E-167
		100	3.69E-30	3.49E-06	0.00606986	2.17E-194	1.97E-05	4.79E-21
		200	1.49E-26	3.06E-103	1.43E-117	1.14E-91	0.009659	0.009659
		300	1.10E-012	xx	2.05E-191	4.19E-177	4.19E-126	1.05E-99
	glove	50	0.788494	0	1.99E-089	0.788494	0.788494	0.788494
		100	0.788494	6.93E-292	7.43E-207	0	7.30E-272	1.33E-264
		200	2.80E-021	2.52E-123	4.96E-047	9.69E-010	1.35E-096	1.35E-096
		300	0.000143	1.01E-154	4.10E-135	2.51E-17	1.89E-103	0.264497
WebKB	tf	7229	0	0	0	0.788494	0.788494	0.788494
	tf-idf	7229	0	0	0	0	0	0
	word2vec	50	0.788494	0.2513	0.308194	1.91E-006	0.788494	0.541214
		100	0.788494	0.670639	0.539444	0.0662238	0.78849	0.076022
		200	0.45977	5.26E-052	0.560392	3.48E-045	0.717001	0.693676
		300	4.55E-013	1.71E-009	2.91E-13	1.34E-078	7.48E-011	0.135651
	glove	50	5.94E-014	0.0	0.0	4.81E-098	0.0	0.0
		100	1.64E-093	0.0	0.0	9.56E-181	0.0	0.0
		200	1.99E-017	0.0	0.0	0.0	0.0	0.0
		300	0.788494	0.0	0.0	0.0	0.0	0.0

Here, xx: values are absent in Table 3

Statistical Significance

To further check the statistical significance of our approach, we have conducted some statistical hypotheses tests named as Welch's *t* test, guided by [62] at 5 %(0.05) significance level. It checks whether the improvements obtained by the proposed SMODoc_clust are statistically significant or happened by chance. Statistical *t* test provides some *p* value. Minimum *p* value implies that the proposed multi-objective clustering approach is better than others. In our experiment, *p* values are calculated considering two groups. Among these two groups, one group corresponds to the list of Dunn index values produced by our algorithm and another corresponds to the list of Dunn index values produced by some other algorithm. In this *t* test, two hypotheses are considered: the null hypothesis and the alternative hypothesis. The first hypothesis is that there is no significant difference between median values of two groups. On the other hand, alternative hypothesis shows that there is significant change between median values of two groups. The obtained *p* values are shown in Table 7 which evidently support the results of Table 3.

Complexity of Proposed Framework

Let N be the number of F-dimensional feature vectors, g be the maximum number of generations.

1) The population is initialized using K-means algorithm. The K-means algorithm takes $\mathcal{O}(tNFk)$ time [43]. Here, t is the number of iterations, K is the number of clusters. If there are P solutions, then for each solution

Table 8 Comparative complexity analysis of existing clustering algorithms

Algorithm	Time complexity
SMODoc_clust	$\mathcal{O}(gP(tNFK + MP))$
MODoc_clust	$\mathcal{O}(gP(tNFK + MP))$
MOCK	$\mathcal{O}(N^2 \log(N) F^3 k^2 P^2 MR)$
VAMOSA	$\mathcal{O}(KN \log(N) Total_{Iter})$
NSGA-II-clust	$\mathcal{O}(gP(tNFK + MP))$
SOGA	$\mathcal{O}(gtPNKF)$
K-means	$\mathcal{O}(tNKF)$
single-linkage	$\mathcal{O}(N^2 log(N))$

Here, R is the number of reference distributions [28]; K is the maximum number of clusters present in a data set which is $\sqrt{N}$; N is the number of data points; $Total_{Iter}$ is the number of iterations used and chosen in such a way that number of fitness evaluations of all the algorithms become equal

we have to calculate M objective functions; thus, total complexity to initialize population (including objective function calculation) will be $\mathcal{O}(P(tNFk + M))$.

2) Training complexity of SOM is $O(P^2)$ as mentioned in [50].
3) Extraction of neighborhood relationship for each solution takes $\mathcal{O}(P^2)$ time because of the calculation of the Euclidean distance of each neuron with respect to other neurons using associated weight vectors, which is a $P \times P$ matrix.
4) Crossover and mutation operations of differential evolution algorithm take constant time; these involve some addition, subtraction or multiplication operations. This implies, new solution generation using crossover and mutation takes $\mathcal{O}(P)$ time as new solution is required to be generated for each solution in the population.
5) K-means clustering steps are applied on each new solution and the objective functional values are calculated. This takes $\mathcal{O}(P(tNFk + M))$ time.
6) Non-dominated sorting takes $\mathcal{O}(MP^2)$ time as for each objective, comparison is required to be performed for each solution with respect to other solutions.

Thus total run time complexity = $\mathcal{O}(P(tNFK + M) + g(P^2 + P^2 + P + P(tNFK + M) + MP^2))$

Here, step-2 to step-3 will be repeated upto g number of generations.

$$\implies \mathcal{O}(P(tNFK+M)+g(2P^2+P+P(tNFK+M)+MP^2))$$

$$\implies \mathcal{O}(P(tNFK + M) + g(2P^2 + PtNFK + MP^2))$$

$$\implies \mathcal{O}(P(tNFK + M) + g(MP^2 + PtNFK))$$

$$\implies \mathcal{O}((1 + g)PtNFK + PM(1 + gP))$$

$$\implies \mathcal{O}(gPtNFK + gMP^2)$$

$$\implies \mathcal{O}(gP(tNFK + MP))$$

Thus, total complexity of our proposed system is $\mathcal{O}(gP(tNFK + MP))$.

Similarly, complexity of NSGA-II-clust can also be analyzed. The total run-time complexity of NSGA-II-clust is $\mathcal{O}(P(tNFK + M) + g(P(tNFK + M) + MP^2))$. Here, the first term is for population initialization and calculation of objective functional values; and the second term, $P(tNFK + M) + MP^2$ is for application of K-means clustering on new solution generated and then applications of non-dominated sorting and crowding distance mechanisms [20]. On solving, this boils down to $\mathcal{O}(gP(tNFK + MP))$.

Comparison of Complexity Analysis with other Algorithms

We have compared the time complexities of existing clustering algorithms and those are reported in Table 8. It is important to note that reported complexities of the existing algorithms are directly taken from the reference papers. It can be seen from Table 8 that the time complexities of our proposed multi-objective automatic document clustering algorithm with SOM (SMODoc_clust) and without SOM (MODoc_clust) based operators are almost same. MOCK algorithm is more expensive than ours. NSGA-II-clust runs with same complexity as of our proposed system. On comparing SOGA and K-means, it was found that SOGA takes little higher time as it is based on the search capability of genetic algorithm.

Conclusions and Future Works

In this paper, we have proposed a new automatic multi-objective document clustering approach utilizing the search capability of differential evolution. The current algorithm is a fusion of DE and SOM where the neighborhood information identified by SOM trained on the current population of solutions is utilized for generating the mating pool which can further take part in generating new solutions. The use of SOM during new solution generation helps the proposed clustering algorithm to better explore the search space of optimal partitioning. To generate more diverse solutions, concept of polynomial mutation is incorporated in DE which helps in convergence towards the global optimal solution. Two objective functions, both measuring the compactness and separation of clusters, are considered here and are optimized simultaneously to improve the cluster quality. The efficacy of the proposed multi-objective document clustering technique is shown in automatically partitioning two text document data sets containing some scientific articles and one web-document data set. Results are compared with various state-of-the-art techniques including single as well as multi-objective clustering algorithms and it was found that the proposed approach is able to reach the global optimal solution for all the data sets, while other algorithms got stuck at local optima. The results clearly show that proposed framework is well suited for partitioning the data sets in an automated manner. The proposed algorithm can be easily applied in the field of text-summarization and classification of Chinese text documents based on semantic information. Other applications of the proposed technique can be scope detection of journals/conferences, development of some automatic peer-review support systems, topic-modeling, etc.

Future work will include the applications of the proposed approach in solving some other real-life problems like text-summarization, automatic grading of essays etc. We would also like to investigate the effect of using more than two objectives and use of deep learning based representation of a text document in the developed clustering framework. Moreover making the mating pool size adaptive is another important future research work.

Acknowledgments Dr. Sriparna Saha would like to acknowledge the support from SERB Women in Excellence Award-SB/WEA/08/2017 for conducting this particular research.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Aggarwal CC, Zhai C. Mining text data. Berlin: Springer Science & Business Media; 2012.
2. Al-Radaideh QA, Bataineh DQ. A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms. Cognitive Computation, 1–19. 2018.
3. Arbelaitz O, Gurrutxaga I, Muguerza J, PéRez JM, Perona I. An extensive comparative study of cluster validity indices. Pattern Recogn. 2013;46(1):243–256.
4. Bandyopadhyay S, Maulik U. Nonparametric genetic clustering: comparison of validity indices. IEEE Trans Syst, Man, Cybern Part C (Applications and Reviews). 2001;31(1):120–125.
5. Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recogn. 2002;35(6):1197–1208.
6. Bandyopadhyay S, Saha S. Gaps: a clustering method using a new point symmetry-based distance measure. Pattern Recogn. 2007;40(12):3430–3451.
7. Bandyopadhyay S, Saha S. A new principal axis based line symmetry measurement and its application to clustering. In: International Conference on Neural Information Processing. Springer; 2008. p. 543–550.
8. Bandyopadhyay S, Saha S. A point symmetry-based clustering technique for automatic evolution of clusters. IEEE Trans Knowl Data Eng. 2008b;20(11):1441–1457.
9. Bandyopadhyay S, Maulik U, Mukhopadhyay A. Multiobjective genetic clustering for pixel classification in remote sensing imagery. IEEE Trans Geoscience Remote Sens. 2007;45(5):1506–1511.
10. Bandyopadhyay S, Saha S, Maulik U, Deb K. A simulated annealing-based multiobjective optimization algorithm: Amosa. IEEE Trans Evol Comput. 2008;12(3):269–283.
11. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. J Mach Learn Res. 2003;3:993–1022.
12. Buitelaar P, Eigner T. Topic extraction from scientific literature for competency management. In: The 7th International Semantic Web Conference; 2008. p. 25–66.
13. Cardoso-Cachopo A. Improving Methods for Single-label Text Categorization PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa. 2007.
14. Carpenter MP, Narin F. Clustering of scientific journals. J Assoc Inform Sci Technol. 1973;24(6):425–436.
15. Yw C, Zhou Q, Luo W, Du JX. Classification of chinese texts based on recognition of semantic topics. Cogn Comput. 2016;8(1):114–124. https://doi.org/10.1007/s12559-015-9346-8.
16. Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. IEEE Trans Syst, Man, Cybern-Part A: Syst Human. 2008;38(1):218–237.
17. Davies DL, Bouldin DW. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI. 1979;1(2):224–227. https://doi.org/10.1109/TPAMI.1979.4766909.
18. Deb K, Vol. 16. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.
19. Deb K, Tiwari S. Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization. Eur J Oper Res. 2008;185(3):1062–1087.
20. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput. 2002;6(2):182–197.
21. Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res. 2006;7(Jan):1–30.
22. Doerre J, Gerstl P, Goeser S, Mueller A, Seiffert R. Taxonomy generation for document collections. US Patent 6,446,061. 2002.
23. Dutta P, Saha S. Fusion of expression values and protein interaction information using multi-objective optimization for improving gene clustering. Comput Biol Med. 2017;89:31–43.
24. Fortuna B, Grobelnik M, Mladenic D. Visualization of text document corpus. Informatica. 2005;29:4.
25. Goldstein J, Mittal V, Carbonell J, Kantrowitz M. Multi-document summarization by sentence extraction. In: Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4, Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-ANLP-AutoSum '00; 2000. p. 40–48. https://doi.org/10.3115/1117575.1117580.
26. Gu F, Liu HL, Tan KC. A multiobjective evolutionary algorithm using dynamic weight design method. Int J Innovative Comput Inf Control. 2012;8:3677–3688.
27. Gupta V, Kaur N. A novel hybrid text summarization system for punjabi text. Cogn Comput. 2016;8(2):261–277.
28. Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE Trans Evol Comput. 2007;11(1):56–76.
29. Haykin SS, Vol. 3. Neural networks and learning machines. Upper Saddle River: Pearson; 2009.
30. Iorio A, Li X. Rotated problems and rotationally invariant crossover in evolutionary multi-objective optimization. Int J Comput Intell Appl. 2008;7(02):149–186.
31. Jain AK, Dubes RC. Algorithms for clustering data. Upper Saddle River: Prentice-Hall, Inc; 1988.

32. Kashef R, Kamel MS. Enhanced bisecting k-means clustering using intermediate cooperation. Pattern Recogn. 2009;42(11): 2557–2569.
33. Kennedy J. Particle swarm optimization. In: Encyclopedia of machine learning. Springer; 2011. p. 760–766.
34. Kohonen T. The self-organizing map. Neurocomputing. 1998;21(1):1–6.
35. Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: a tutorial. Reliability Eng Syst Safety. 2006;91(9):992–1007.
36. Korenius T, Laurikkala J, Järvelin K, Juhola M. Stemming and lemmatization in the clustering of finnish text documents. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM; 2004. p. 625–633.
37. Kovács F, Legány C, Babos A. Cluster validity measurement techniques. In: 6th International symposium of hungarian researchers on computational intelligence; 2005.
38. Lauren P, Qu G, Yang J, Watta P, Huang GB, Lendasse A. Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks. Cognitive Computation, 1–14. 2018.
39. Le Q, Mikolov T. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14); 2014. p. 1188–1196.
40. Li Y, Pan Q, Yang T, Wang S, Tang J, Cambria E. Learning word representations for sentiment analysis. Cogn Comput. 2017;9(6):843–851.
41. Lichman M. UCI machine learning repository. http://archive.ics.uci.edu/ml. 2013.
42. Loper E, Bird S. Nltk: the natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, ETMTNLP '02; 2002. p. 63–70. https://doi.org/10.3115/1118108.1118117.
43. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press; 2009.
44. Maulik U, Bandyopadhyay S. Performance evaluation of some clustering algorithms and validity indices. IEEE Trans Pattern Anal Mach Intell. 2002;24(12):1650–1654.
45. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 2013. arXiv:13013781.
46. Moran K, Wallace BC, Brodley CE. Discovering better aaai keywords via clustering with community-sourced constraints. In: AAAI; 2014. p. 1265–1271.
47. Pakhira MK, Bandyopadhyay S, Maulik U. Validity index for crisp and fuzzy clusters. Pattern Recogn. 2004;37(3):487–501.
48. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 1532–1543.
49. Price K, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Berlin: Springer Science & Business Media; 2006.
50. Roussinov DG, Chen H. A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation. 1998.
51. Saha S, Bandyopadhyay S. A symmetry based multiobjective clustering technique for automatic evolution of clusters. Pattern Recogn. 2010;43(3):738–751.
52. Saha S, Bandyopadhyay S. Some connectivity based cluster validity indices. Appl Soft Comput. 2012;12(5):1555–1565.
53. Saha S, Bandyopadhyay S. A generalized automatic clustering algorithm in a multiobjective framework. Appl Soft Comput. 2013;13(1):89–108.
54. Sahi M, Gupta V. A novel technique for detecting plagiarism in documents exploiting information sources. Cogn Comput. 2017;9(6):852–867.
55. Saini N, Chourasia S, Saha S, Bhattacharyya P. A self organizing map based multi-objective framework for automatic evolution of clusters. In: International Conference on Neural Information Processing. Springer; 2017. p. 672–682.
56. Saini N, Saha S, Bhattacharyya P. Cascaded Som: an improved technique for automatic email classification. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. p. 1-8.
57. Singh J, Gupta V. An efficient corpus-based stemmer. Cogn Comput. 2017;9(5):671–688.
58. Starczewski A. A new validity index for crisp clusters. Pattern Anal Applic. 2017;20(3):687–700.
59. Steinbach M, Karypis G, Kumar V, et al. A comparison of document clustering techniques. In: KDD Workshop on text mining, Boston; 2000. p. 525–526.
60. Suresh K, Kundu D, Ghosh S, Das S, Abraham A. Data clustering using multi-objective differential evolution algorithms. Fundamenta Informaticae. 2009;97(4):381–403.
61. Wang H. Introduction to word2vec and its application to find predominant word senses. 2014. http://complinghssntuedusg/courses/hg7017/pdf/word2vec and its application to wsd pdf.
62. Welch BL. The generalization of 'student's' problem when several different population variances are involved. Biometrika. 1947;34(1/2):28–35. http://www.jstor.org/stable/2332510.
63. Witten I, Bainbridge D, Paynter G, Boddie S. Importing documents and metadata into digital libraries: requirements analysis and an extensible architecture. Research and Advanced Technology for Digital Libraries, 219–229. 2002.
64. Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM; 2003. p. 267–273.
65. Zhang H, Zhang X, Gao XZ, Song S. Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble. Neurocomputing. 2016;173:1868–1884.
66. Zhang H, Zhou A, Song S, Zhang Q, Gao XZ, Zhang J. A self-organizing multiobjective evolutionary algorithm. IEEE Trans Evol Comput. 2016;20(5):792–806. https://doi.org/10.1109/TEVC.2016.2521868.
67. Zhou A, Qf Z, Zhang G. Multiobjective evolutionary algorithm based on mixture gaussian models. J Softw. 2014;25(5):913–928.