



Fusion of evolvable genome structure and multi-objective optimization for subspace clustering

Dipanjoyoti Paul^{1,*}, Sriparna Saha¹, Jimson Mathew

Department of Computer Science and Engineering Indian Institute of Technology, Patna, India

ARTICLE INFO

Article history:

Received 20 August 2018

Revised 11 April 2019

Accepted 22 May 2019

Available online 31 May 2019

Keywords:

Multi-objective optimization

Subspace clustering

Evolvable genome structure

Cluster validity indices

Biclustering

ABSTRACT

Subspace clustering techniques become paramount in pattern recognition for detecting local variations from high dimensional data. Several techniques exist in the recent literature for subspace clustering, majority of which optimize implicitly or explicitly a single cluster quality measure. Inspired by the success of multi-objective optimization in solving clustering problem, we developed a multi-objective based subspace clustering technique in this paper. The proposed technique simultaneously optimizes two subspace cluster quality measures, capable of capturing different cluster shapes/properties. Two existing cluster quality measures, XB-index and PBM-index, are modified to develop subspace cluster validity indices, and then those are used as optimization criteria. These cluster validity indices measure the appropriateness of generated subspace clusters in terms of intra-subspace cluster similarity and separation between subspace clusters. The proposed approach utilizes a new evolvable genome structure which stores the information about subspaces in its phenotype and genotype and evolves this genome structure with the help of different genetic operators. The developed algorithm is applied on ten standard real-life data sets and sixteen synthetic datasets for identifying different subspace clusters. The results obtained by this algorithm are compared against some state-of-the-art techniques with respect to different performance metrics. Experimentation reveals that the proposed algorithm is able to take advantages of its evolvable genomic structure and multi-objective based framework and it can be applied to any data set. In a part of the paper, the efficacy of the proposed technique is also shown for bi-clustering of gene-expression data sets.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The idea of clustering is to group a set of objects together that share some similar properties and are dissimilar to the objects belonging to other groups [1]. Clustering has a huge number of applications in machine learning such as bio-informatics, image segmentation and marketing to name a few [2]. In recent years, a huge volume of data with a large set of features, known as high-dimensional data, is generated by different application domains. Distance-based similarity measures tend to be less meaningful for high dimensional datasets leading to the problem of *curse of dimensionality* [3]. However, it has appeared that not all the features used to describe an object are relevant for the purpose of clustering. Therefore, many feature selection algorithms Gao et al. [4], Zheng and Wang [5] have been developed to select the appropriate

set of features which actually represent the data set. Recently, the concept of subspace clustering [6], an emerging research area which is beneficial when dealing with the high dimensional data has been widely in use. It is based on the principle that, different subsets of features are relevant for clustering different subsets of objects and thus it attempts to cluster both objects and features together [1]. Concepts of subspace clustering can be applied for solving many problems such as low rank representation problem [7], unsupervised feature selection [8] etc. Many machine learning problems have been solved using subspace clustering approach such as SAP [9], FG-K-means [10], AFG-K-means [11] to name a few.

A lot of works have been carried out in the area of clustering using top-down approaches such as PROCLUS [12], ORCLUS [13] and bottom-up approaches such as CLIQUE [14], ENCLUS [15], MAFIA [16] to name a few. Top-down approaches start developing clusters using the full set of features and continue to remove a feature incrementally until the cluster performance reaches to a plateau. Bottom-up approaches start with searching dense regions for each feature space. It adds a dimension in its subspace

* Corresponding author.

E-mail addresses: dipanjoyoti.pcs17@iitp.ac.in (D. Paul), sriparna@iitp.ac.in (S. Saha), jimson@iitp.ac.in (J. Mathew).

¹ Both authors contributed equally.

when the dense region after adding a feature satisfies a predefined threshold [3]. Subspace clustering problem has also been solved by using Sparse representation [17] and Low-Rank representation (LRR) [18] of the data vectors. However, the algorithms SSC [17] and LRR [18] require the information about the number of subspaces to be generated at the beginning. In recent years some evolutionary approaches such as ChameleoClust [19], Kymeroclust [20] etc. are developed to solve the subspace clustering problem. These algorithms take the advantages of dynamic evolution of genome structure which leads to variable genome length or a variable ratio of functional versus non-functional elements [21]. ChameleoClust defines a *course grained* genome which contains sample of objects in the form of tuples. These tuples are mapped at the phenotype level to define core point locations in different dimensions, which then lead in building a subspace cluster. Kymeroclust [20] is an abstract evolutionary algorithm where each individual is represented by a phenotype constituted by a set of core points in their subspaces.

Most of the previous state-of-the-art algorithms on subspace clustering implicitly or explicitly optimize a single cluster quality measure. Kymeroclust uses a distance-based measure to determine the similarity between objects. It assigns an object to a cluster where the distance between the center of that cluster and the object is the minimum. Thus, this algorithm only optimizes cluster-compactness as the objective function and is limited in capturing only those subspace clusters which follow some particular shapes or properties. Therefore, for a better subspace cluster generation, along with compactness of objects within the cluster, some other cluster quality measures, capturing separation between clusters, are required to be incorporated in the optimization process. In order to simultaneously optimize more than one objective functions while performing subspace clustering on a given data set, concepts of multi-objective optimization (MOO) can be utilized [22]. Use of MOO can help in removing the limitation of using a single objective function in the subspace clustering process and is able to generate good quality subspace clusters.

Multi-objective optimization [22] aims at finding one or more optimal solutions by optimizing a set of conflicting objective functions. This technique is very useful while dealing with the clustering of datasets. Various MOO techniques are used in the domain of clustering in a hope to generate better clusters [22,23]. In this paper, we incorporate multi-objective optimization techniques in determining subspace clusters. The appropriateness of the generated subspace clusters is measured in terms of intra-subspace cluster similarity and separation between subspace clusters.

An evolutionary based multi-objective subspace clustering algorithm inspired by Kymeroclust [20] is proposed in this paper. This algorithm defines an individual as a phenotype which consists of a set of core points called phenotypic trait. Each core point is associated with a set of objects defined in its own subspace. Again, a genome is defined as the number of repeated genes involved in each biological function of a phenotypic trait. An important phenomenon of evolutionary approach is that objects in the genome have to undergo some duplication or replacement operation for evolution.

The mutation operation carried out using duplication-divergence operator can lead to either creation of a new phenotypic trait by generating a new center in the phenotype or a new biological function contributing to a previous existing phenotypic trait. To measure the wellness of genes that are surrounded by some phenotypic trait, the definitions of two existing cluster validity indices, PBM-index [24] and XB-index [25] are modified to handle the concept of subspace clustering effectively. Thereafter the modified versions of PBM-index and XB-index are optimized simultaneously using the search capability of MOO. These cluster validity indices measure the appropriateness of generated sub-

space clusters in terms of intra-subspace cluster similarity and separation between subspace clusters. The objective is to minimize the XB-index and to maximize the PBM-index in order to obtain a good set of subspace clusters. For the ease of implementation, we calculated the inverse of PBM-index and therefore our task is to minimize both the objective functions. In order to select a good set of solutions, concepts of non-dominated sorting and crowding distance operator can be applied. The proposed multi-objective based subspace clustering approach is applied on twenty-three real-life and artificial data sets to show its efficacy. Obtained results are compared against some existing subspace clustering approaches with respect to different performance metrics. Our experimentation clearly reveals the superiority of utilizing MOO concept in solving subspace clustering problem. In a part of the experiments, the superiority of the proposed method is also tested using three big datasets and also by applying the developed algorithm in solving a real-life problem of gene expression bi-clustering. The novel aspects of the proposed method are listed below:

- This paper reports the first attempt in integrating MOO and genomic structure for solving the subspace clustering problem.
- Multi-objective optimization framework is utilized to simultaneously optimize several subspace cluster quality measures.
- The existing cluster quality measures are modified to handle subspace clustering problem.
- As a part of the experiments, the proposed algorithm is applied on the bi-clustering of gene expression data to show the efficacy of the existing technique in solving some real-life problems. Bi-clustering of gene expression data is a subspace clustering problem where a subset of rows and a subset of columns need to be selected.

2. MOOSubClust

The proposed algorithm follows the structure of evolvable genome to explore the search space of subspace clustering. The concept of MOO is integrated with evolvable genome structure, an evolutionary optimization process to identify good quality subspace clusters from a dataset automatically.

2.1. Data preprocessing

A dataset $D = \{d_1, d_2, \dots, d_S\}$ is a set of objects and each object is described by $F = \{f_1, f_2, \dots, f_F\}$ dimensional features. The size of the dataset $|S|$ is the number of objects present in the dataset and $|F|$ is the total number of features where a feature represents a particular property of the dataset. To avoid the varying range of values over the features, z-score of each feature value v is calculated and is replaced with $z = \frac{v - \lambda}{\sigma}$. λ is the mean of the dataset and σ is the standard deviation of the corresponding feature.

2.2. Overview of the algorithm

In this algorithm, we define a model represented by a phenotype and a genotype in two dimensional matrix form. The algorithm starts with *sol* such initially empty models and the mutation operation is applied on each initial model in order to get non-empty models. The objects are grouped around the core points of phenotype known as subspace clusters and the objective values using modified versions of XB-index and PBM index corresponding to each model are calculated. At each iteration, new models are generated by applying mutation operation and the objective values

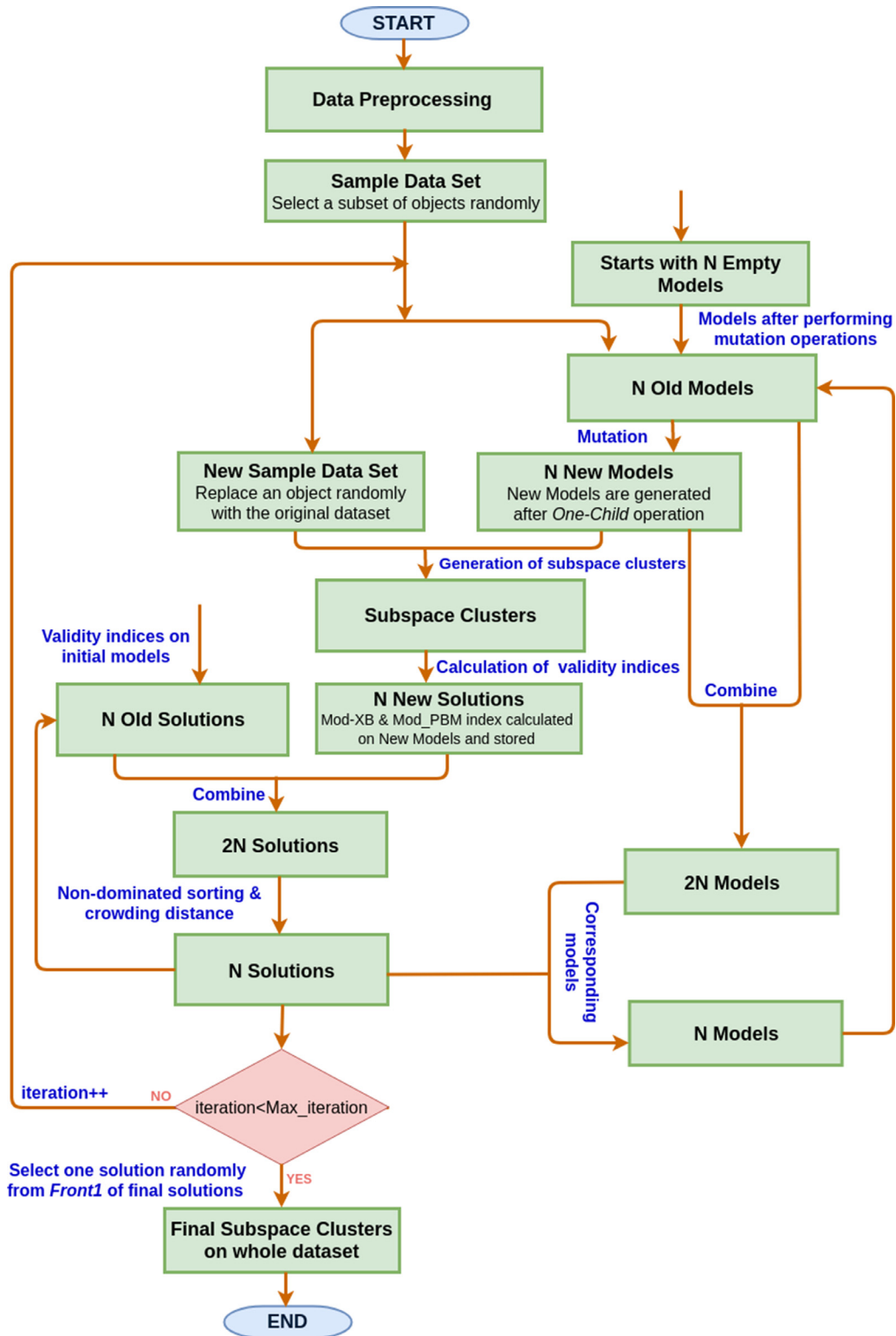


Fig. 1. Flowchart of the proposed algorithm (MOOSubClust).

of the new models are merged with the previous objective values to form a pool of size $2 \times sol$. Non-dominated sorting and crowding distance operator are used to select top sol solutions out of $2 \times sol$ solutions based on the objective functions. Finally, after all the iterations, the algorithms output the best set of clusters where each cluster is associated with a subset of features. The detailed flow of the algorithm is shown in Fig. 1.

2.3. Phenotype and genotype representation

The MOOSubClust algorithm uses a pair of matrices to describe the Phenotype-Genotype representation called model $M = \langle P, G \rangle$. Both the matrices have SC_{max} (maximum permissible subspace clusters) rows and F columns (features of the data set). The matrix P represents the values of the biological function defining

each phenotypic trait, i.e., the coordinates of the cluster centers along different dimensions. The matrix G defines a genotype, representing the number of genes associated with each biological function involved in each phenotypic trait, i.e., the associated weights of different centers along different features.

The algorithm starts with initializing both the phenotype and genotype with empty matrices. The matrices get updated to initial model by performing One-Child mutation operator until the number of genes in the genome equals to SC_{max} . The One-Child operator [20] [Section 2.4] modifies the phenotype representation by modifying the core point location and modifies the genotype representation by increasing the number of genes corresponding to the location where the phenotype is updated.

2.4. One-Child computation function

The One-Child function is used for mutation operation in order to generate new models from the previous models. This function is based on the duplication-divergence operator. The authors of [26,27] mentioned that in the evolutionary process, duplication of genes has an important role. However, once a gene is duplicated, it will not perform any useful task unless and until it is mutated and gives birth to new biological function. Again, the authors in [27] mentioned that, once a gene is duplicated, most of the times it is deleted just after being created. To avoid such cases, duplication and deletion must be chosen in a non-additive manner unlike the mutation operator in ChameleoClust [19]. Therefore, in One-Child function, deletion and duplication of genes are performed in a single operation.

This function initially calculates the weight of the genotype G and if the weight equals to the SC_{max} , it performs deletion operation by selecting a gene randomly from the genotype and deleting it. If the number of genes becomes zero after deletion, the corresponding phenotype value is also marked as zero [Lines 2 to 7 of Algorithm 1]. Next, the function performs duplication-divergence operator by drawing an object uniformly from sample data-space and a feature from the feature-space [Line 10 of Algorithm 1]. The co-ordinates of this sample data along the feature define the value of the biological function after divergence. Next, the function

Algorithm 1 : One-Child $((P, G), D', SC_{max})$

Input: P = Phenotype, G = Genotype, D' = Sample dataset, SC_{max} = Maximum permissible subspace clusters

```

1:  $(P', G') \leftarrow (P, G)$ 
2:  $weight \leftarrow \sum_{i=1}^{SC_{max}} \sum_{j=1}^F G'_{i,j}$ 
3: if  $weight = SC_{max}$  then
4:   Draw  $(i,j) \in \{1, \dots, SC_{max}\} * \{1, \dots, F\}$ 
5:    $G'_{i,j} \leftarrow G'_{i,j} - 1$ 
6:   if  $G'_{i,j} = 0$  then
7:      $P'_{i,j} \leftarrow 0$ 
8:   end if
9: end if
10: Draw  $d_s \in D'$  uniformly ; Draw  $f \in \{1, \dots, F\}$  uniformly
11:  $\alpha = \{i \in \{1, \dots, SC_{max}\} | \sum_{j=1}^F G'_{i,j} > 0\}$ 
12: Draw a random number  $r$  uniformly in  $[0,1]$ 
13: if  $r \leq \frac{1}{weight}$  then
14:   Draw  $c \in \{1, \dots, SC_{max}\} \setminus \alpha$  uniformly
15: else
16:   Draw  $c \in \alpha$  with probability  $\frac{\sum_{j=1}^F G'_{c,j}}{weight}$ 
17: end if
18:  $G'_{c,f} \leftarrow G'_{c,f} + 1$  ;  $P'_{c,f} \leftarrow$  Coordinates of  $d$  along feature  $f$ 
19: return  $(P', G')$ 
```

chooses a random center which either can lead to the creation of a new cluster center or can update an existing cluster center. When an existing cluster is chosen for duplication, it can either undergo a functional divergence by creating a new function or it can undergo an allelic divergence of the duplicated genes by simply modifying the biological function. The probability of choosing a new center depends on the inverse of the genome weight. If the inverse of the genome weight is less than some random probability, a new subspace cluster center will be created, otherwise, an existing center will be updated [Lines 13 to 16 of Algorithm 1].

2.5. Generation of subspace clusters

In each iteration, the algorithm generates temporal subspace clusters with respect to the sample data-space. Subspace clusters are generated by assigning each object to a particular cluster of current phenotype for which the distance between the cluster center and the object becomes minimum.

Let us consider a phenotype P contain many centers such that each center $p_c \in P$ is associated to a subspace f_i of F . For each object, the function calculates the distance of the object with respect to all the cluster centers [Lines 5 to 7 of Algorithm 2]. The distance

Algorithm 2 : GenerateSubspaceCluster (D, P)

Input: D = Dataset, P = Phenotype

```

1:  $dis \leftarrow$  a large number
2: for all  $i=1$  to  $[len(P)]$ ;  $j=1$  to  $[len(D)]$ ; do
3:    $membershipdegree[i][j]=0$ 
4: end for
5: for  $d_s$  in  $D$  do
6:   for  $p_c$  in  $P$  do
7:      $new\_distance = distance(d_s, p_c)$  {Refer Eqn. 1}
8:     if  $new\_distance < dis$  then
9:        $dis = new\_distance$ 
10:       $cluster = p_c$ 
11:    end if
12:   end for
13:    $membershipdegree[cluster][d_s]=1$ 
14: end for
15: return  $membershipdegree$ 
```

between an object(d_s) and a cluster center(p_c) can be computed as

$$distance(d_s, p_c) = \sum_{f \in f_i} |d_{s,f} - p_{c,f}| + \sum_{f \in F \setminus f_i} |d_{s,f} - \mu_f| \quad (1)$$

Where μ is the origin of the entire space, which contains the value zero after standarization of the dataset. The function then searches for the cluster for which the distance becomes minimum [Lines 8 to 9 of Algorithm 2] and assigns the object to that cluster [Line 13 of Algorithm 2]. If more than one cluster centers give the exact minimum value, one cluster center is selected in a non-deterministic way. The function finally returns the subspace clusters and the objects associated to the clusters [Line 15 of Algorithm 2].

2.6. Objective functions

The task of an objective function is to *minimize* or *maximize* some criteria. In case of clustering, the two objectives that need to be considered for evaluation and selection of optimal clustering are *compactness* and *separation*. The clusters obtained using a

clustering algorithm should have both high compactness and high separation. Further, the validity indices in [24] should be defined in such a way that they should focus on optimizing these two objectives to obtain good quality clusters.

The two objective functions that are used in the algorithm MOOSubClust are modified version of XB-index [25] and modified version of PBM-index [24]. Now onwards, in this paper, we will refer the modified version of XB-index and PBM index as simply XB-index and PBM-index respectively. Both these indices focus on minimizing the distance between a data object to its cluster center and maximizing the separation between different clusters. As subspace clusters involve only a subset of features, definitions of compactness and separation are modified in both PBM and XB indices.

2.6.1. Pakhira Bandyopadhyay Maulik (PBM) Index:

This index was proposed by Pakhira et al. [24]. The PBM-index is calculated using the distances between the points to their centers and the distances between the centers themselves. Manhattan distance is used to calculate various distances as it is robust to the concentration effect [28]. A subspace cluster contains many centers represented by the phenotype P such that each center $p_c \in P$ is associated to a subspace f_i of F . The index function is mathematically defined as:

$$PBM(D, P) = \left[\frac{1}{C} * \frac{H_\mu}{H_{dc}} * H_{cc} \right]^2$$

Where D is the dataset and C is the number of subspace clusters, i.e., the number of nonzero rows in the matrix P . The term H_μ is the sum of the distances of each pattern d_s to the origin of the entire space, μ . The term is independent of the number of clusters and is computed as:

$$H_\mu = \sum_{s=1}^S \sum_{f=1}^F |d_{s,f} - \mu_f|$$

Where $| \cdot |$ refers to the absolute value. The term H_{dc} is the sum of the distances between objects and the corresponding cluster centers where an object is assigned to a particular cluster for which the distance becomes minimum and is computed as follows:

$$H_{dc} = \sum_{s=1}^S \sum_{c=1}^C \min_c[dis(d_s, p_c)]$$

where $dis(d_s, p_c) = \sum_{f \in f_i} |d_{s,f} - p_{c,f}| + \sum_{f \in F \setminus f_i} |d_{s,f} - \mu_f|$. The factor H_{cc} measures the maximum separation between two clusters over all possible pairs of clusters and is computed as:

$$H_{cc} = \max_{1 \leq i \leq C, 1 \leq j \leq C} dis(p_i, p_j)$$

where $dis(p_i, p_j) = \sum_{f \in f_i \cap f_j} |p_{i,f} - p_{j,f}| + \sum_{f \in F \setminus (f_i \cap f_j)} |p_{i,f} - \mu_f|$. The more the value of PBM(D, P), the better is the quality of clusters. In this paper, the inverse of PBM-index is calculated so as to minimize the objective function.

2.6.2. Xie-Beni (XB) Index:

The Xie and Beni index is another objective function proposed by Xie and Beni in [25] which focuses on the compactness and separation between the clusters. The XB-index is computed by the mathematical function shown below.

$$XB(D, P) = \frac{\sum_{s=1}^S \sum_{c=1}^C \min_c[dis(d_s, p_c)]^2}{S * \min_{1 \leq i \leq C, 1 \leq j \leq C} dis(p_i, p_j)^2}$$

The best partitioning of the dataset is indicated by the minimum value of XB-index.

2.7. Non-dominated sorting and crowding distance operators

In a multi-objective optimization scenario, optimization of multiple objective functions may attain conflicting results. Therefore, non-dominated sorting [29] and crowding distance [30] operators are used in such situations to overcome the problem. Non-dominated sorting sorts all the solutions and returns in the form of fronts such as $Front_1$, $Front_2$ and so on. $Front_1$ contains all the solutions that are non-dominated with respect to any other solution. Again, the solutions that are in $Front_2$ are non-dominated with respect to any other solution except solutions in $Front_1$ and so on. However, the solutions that are in a particular front, $Front_i$, are non-dominated to each other.

In this current algorithm, we start with sol solutions and another sol number of solutions are generated in the next iteration. Non-dominating sorting sorts all the ($2 * sol$) solutions based on the values of objective functions and returns in the form of fronts such that all the fronts are arranged in decreasing order of their dominance and the solutions are selected in a sequence from $Front_1$, $Front_2$ and so on. In order to select the best sol solutions, we can consider the solutions of different fronts in an increasing order. Selecting in this way, a case may arise when we need to select p solution/s from q solutions ($p < q$) present in a front $Front_i$. Selection of a subset of solutions from the total solutions present in a front is handled by crowding distance as presented in [30]. Crowding distance selects the solutions which are spaced in a less dense region.

2.8. Steps of MOOSubclust algorithm:

The algorithm takes input a dataset D described by F features, maximum possible clusters, SC_{max} , size of the sample dataset, N , number of solutions, sol and the number of iterations, $Iter$. The algorithm starts by selecting a sample of dataset D' of size N drawn uniformly [Line 1 of Algorithm 3] and by initializing sol number of models with empty matrices, where a model is described by two matrices: a phenotype P (also called a solution) and a genotype G . Mutation operation One-Child function is applied on each of the empty models until the number of genes in the genotype equals to SC_{max} called as initial models. [Line 2 of Algorithm 3]. The steps of the algorithm are enumerated below.

- The objective functions of the initial models are computed with respect to the phenotype P_i and the values are stored in *Population* [Lines 6 to 8 of Algorithm 3].
- At each iteration, first, we replace a data from sample data [Lines 13 to 16 of Algorithm 3] and new models are generated by performing One-Child operation on each of the previous models. The corresponding objective values of the new models are again calculated and are stored in *Population_new* [Lines 20 to 22 of Algorithm 3].
- The ($2 * sol$) objective functions corresponding to ($2 * sol$) solutions are then passed as a parameter to non-dominating sorting and crowding distance [Lines 28 of Algorithm 3] in order to get the best sol number of solutions.
- The algorithm outputs sol number of phenotypes after completion of $Iter$ number of iterations, where each phenotype contains a set of subspace clusters with their corresponding subspaces. Then, we randomly choose 1 phenotype P , from $Front_1$ and *GenerateSubspaceClusters()* function [Refer to Section 2.5] is called [Line 30 of Algorithm 3] on the whole dataset which simply associates each data object to its closest cluster center and returns subspace clusters on the whole dataset.

Algorithm 3 : MOOSubClust ($D, SC_{max}, N, Iter, sol$)

Input: D =Dataset, SC_{max} =Maximum permissible clusters, N =sample dataset size $Iter$ =Number of iteration, sol =Number of solutions

```

1:  $D' \leftarrow \{d_1, d_2, \dots, d_N\}$  uniformly drawn from  $D$ 
2:  $P_0, \dots, P_{sol} \leftarrow$  initial phenotypes ;  $G_0, \dots, G_{sol} \leftarrow$  initial
   genotypes ;
3:  $iteration \leftarrow 0$ ;  $i \leftarrow 0$ 
4: while ( $i < sol$ ) do
5:   membershipdegree= GenerateSubspaceClusters( $D', P_i$ )
   {Refer Algorithm 2}
6:    $XB \leftarrow$  Calculate_XB( $D', P_i$ , membershipdegree)
7:    $PBM \leftarrow$  Calculate_PBM( $D', P_i$ , membershipdegree)
8:    $Population[i] = (XB, PBM)$ 
9:    $Phenotype\_dict[i] = P_i$  ;  $Genotype\_dict[i] = G_i$  ;
10:   $i = i + 1$ 
11: end while
12: while ( $iteration < Iter$ ) do
13:   Draw  $d_s \in D'$  uniformly
14:    $D' \leftarrow D' \setminus \{d_s\}$ 
15:   Draw  $d_s^* \in D$  uniformly
16:    $D' \leftarrow D' \cup \{d_s^*\}$ 
17:   while ( $j < sol$ ) do
18:      $P_j, G_j \leftarrow$  One-Child( $Phenotype\_dict[j], Genotype\_dict[j],$ 
        $D', SC_{max}$ )
19:     membershipdegree= GenerateSubspaceClusters( $D', P_j$ )
20:      $XB \leftarrow$  Calculate_XB( $D', P_j$ , membershipdegree)
21:      $PBM \leftarrow$  Calculate_PBM( $D', P_j$ , membershipdegree)
22:      $Population\_new[j] = (XB, PBM)$ 
23:      $Phenotype\_dict\_new[j] = P_j$  ;  $Genotype\_dict\_new[j] = G_j$  ;
24:   end while
25:    $Population\_Final \leftarrow Population \cup Population\_new$ 
26:    $Phenotype\_dict\_Final \leftarrow Phenotype \cup Phenotype\_new$ 
27:    $Genotype\_dict\_Final \leftarrow Genotype \cup Genotype\_new$ 
28:    $sol \text{ solutions} \leftarrow$  Crowding_Distance(Non_Dominating_
     solution( $Population\_Final$ ))
29: end while { until  $Iter$  iterations achieved }
30: membershipdegree= GenerateSubspaceClusters( $D, P$ )

```

3. Experimental setup

The various evaluation measures presented in [31] that are designed to quantify the quality of subspace clusters are used as reference frameworks in order to compare the proposed algorithm with many state-of-the-art algorithms.

3.1. Datasets

The performance of our algorithm, *MOOSubClust*, is studied using seven benchmark real-life datasets collected from [32] as reported in Table 1. These datasets (glass, breast, liver available in website,² pendigits,³ diabetes,⁴ vowel,⁵ shape) are also used in many state-of-the-art algorithms. The original class label information is used to measure the quality of solutions generated by different approaches. The proposed approach *MOOSubClust* is applied on these real-life data sets to generate different subspace clusters. The results are then compared with those obtained by different state-of-the-art algorithms, CLIQUE [14], PROCLUS [12], ChameleonClust

Table 1

Description of real-world datasets.

Dataset	Number of dimensions	Number of classes	Number of objects
liver	6	2	345
breast	33	2	198
shape	17	9	160
pendigits	16	10	7494
diabetes	8	2	768
vowel	10	11	990
glass	9	6	214

Table 2

Description of synthetic datasets.

Dataset	Number of dimensions	Number of classes	Number of objects
S1500	20	12	1595
S2500	20	12	2658
S3500	20	13	3722
S4500	20	12	4785
S5500	20	12	5848
D05	5	12	1595
D10	10	12	1595
D15	15	13	1598
D20	20	12	1595
D25	25	12	1595
D50	50	13	1596
D75	75	13	1596
N10	20	13	1611
N30	20	12	2071
N50	20	13	2900
N70	20	12	4833

[19], KymeroClust [20] to name a few as reported in [20] with respect to different performance metrics.

The proposed algorithm is also executed on 16 synthetic datasets as used in [32] as reported in Table 2. For these datasets, true clusters and their subspaces are known. Each dataset has 10 hidden subspace clusters lying in subspaces having 50%, 60% and 80% of the total dimension of the dataset. Out of these 16 datasets, seven synthetic datasets D05, D10, D15, D20, D25, D50, D75 were used to study the scalability of the dimension. Five synthetic datasets S1500, S2500, S3500, S4500, S5500 were used to study the behaviour of the algorithm with respect to the size of the dataset. Finally, four synthetic datasets N10, N30, N50, N75 were used to check the scalability when the noise is incorporated with 10%, 30%, 50% and 75% noisy objects in the dataset, respectively.

3.2. Parameter settings

The number of classes present in the real-life datasets may not be equal to the number of clusters obtained. A class may be split to generate many clusters and therefore it would be very common to get more number of clusters than classes. Based on this notion, parameters defined in [20] are also used in the current algorithm. The value of the expected number of clusters is set as three times of the number of classes, i.e., $ExpectedClusters = 3 * NumClasses$. The maximum size of the model, SC_{max} , is defined in a fashion that it should allow to build a model containing at least $ExpectedClusters$ number of clusters considering all the features F , i.e., $SC_{max} = ExpectedClusters * F$. To reduce the time complexity of the algorithm we have used few samples from the dataset, selected uniformly from the original dataset. The size of the sample, N , is an important parameter and its value should be set as to get an overall impact of the whole dataset. Here we set $N = 25 * ExpectedClusters$. Again, there is no hard and fast rule for setting the value of the parameter, number of iterations denoted

² <http://archive.ics.uci.edu/ml/datasets.html>.

³ <http://odds.cs.stonybrook.edu/pendigits-dataset/>.

⁴ <https://www.kaggle.com/uciml/pima-indians-diabetes-database/version/1>.

⁵ <http://homepages.cae.wisc.edu/~ece539/data/index.html>.

by *Iter*. User can monitor this value and can stop executing the process when the objective values do not improve further. The number of iterations can be defined as $Iter = 10 * ExpectedClusters * SC_{max}$.

For synthetic datasets, the number of expected clusters is considered to be 10, therefore the maximum size of the model SC_{max} was set to $SC_{max} = F * 10$, the sample size and the number of iterations were set to $N = 25 * 10$ and $Iter = 10 * 10 * SC_{max}$ respectively.

3.3. Evaluation measures

In order to evaluate and compare the quality of generated subspace clusters by different algorithms, standard evaluation measures presented in [31] are used. These evaluation measures are F_1 [31], $\overline{Entropy}$ (1-Entropy) [31], Clustering Error(CE) [33], Accuracy [31] and $\overline{RNIA}$ (1-RNIA) [33]. Along with these primary subspace clustering measures, we have also considered some other measures like coverage (objects which are part of a cluster), the number of subspace clusters, average features of the subspace clusters and execution time.

4. Experimental results

The algorithm *MOOSubClust* is applied on 23 datasets (7 real-life datasets and 16 synthetic datasets) and its performance is compared with 12 state-of-the-art algorithms. However, comparison shows that there is no ever winning paradigm or clustering method, each algorithm has its own benefits or interests. In this section, the advantages of our algorithm and how it could be considered as most superior than other algorithms is explained.

4.1. Results on real-life datasets

This section presents the results obtained on 7 real-life datasets utilized in [32]. The algorithm is executed 10 times on each real-life data set and the *minimum* and the *maximum* values of different performance metrics over several runs are presented in Tables 3–9. In these tables, the best results are highlighted with grey color and results similar to the best results are highlighted in bold. A rank-based comparison among algorithms using different aggregations is also presented in this section.

4.1.1. Analysis of performance with respect to primary cluster quality measure

To calculate the overall cluster quality measure, rankings of the 13 algorithms were computed using both the *maximum* and *minimum* scores with ranks ranging from 1 to 13, rank value 1 being the algorithm corresponding to highest quality. The overall average ranking of an algorithm was then simply the average of its rank values for different evaluation metrics, on each data set. The average ranks of each algorithm with respect to different cluster quality measures are reported in Fig. 2(a). The proposed algorithm ranks second in the overall cluster quality measure beating all the algorithms except MINECLUS.

4.1.2. Cluster quality with respect to individual primary evaluation metric

The cluster quality of each algorithm corresponding to individual evaluation metric is also checked. The same ranking procedure was used to compute the average ranks of algorithms with respect to different datasets. The ranks of the algorithm corresponding to evaluation metric F_1 , Accuracy, CE, RNIA and Entropy are shown in Fig. 2(b)–(f), respectively. With respect to Entropy and CE, the

Table 3
Results for dataset *liver*: 6 dimensions, 2 classes, 345 objects.

	F_1		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.67	0.64	0.68	0.58	0.11	0.07	0.51	0.35	0.18	0.11	0.99	0.90	45	13	3.0	1.9	625324	1625
CLIQUE	0.68	0.65	0.67	0.58	0.08	0.02	0.38	0.03	0.10	0.02	1.00	1.00	1922	19	4.1	1.7	38281	15
MINECLUS	0.73	0.63	0.65	0.58	0.09	0.09	0.68	0.48	0.33	0.16	0.99	0.92	64	32	4.0	3.7	49563	1954
SUBCLU	0.68	0.68	0.64	0.58	0.11	0.02	0.68	0.05	0.07	0.02	1.00	1.00	334	64	3.4	1.3	1422	47
SCHISM	0.69	0.69	0.68	0.59	0.04	0.03	0.45	0.26	0.10	0.08	0.99	0.99	90	68	2.7	2.1	31	0
FIRE	0.58	0.04	0.58	0.56	0.14	0.00	0.39	0.01	0.37	0.00	0.84	0.03	10	1	3.0	1.0	531	46
PROCLUS	0.53	0.39	0.63	0.63	0.26	0.11	0.66	0.25	0.05	0.05	0.83	0.46	6	2	5.0	3.0	78	31
INSCY	0.66	0.66	0.62	0.61	0.03	0.03	0.42	0.39	0.21	0.20	0.85	0.81	166	130	2.1	2.1	407	234
STATPC	0.69	0.57	0.65	0.58	0.23	0.01	0.58	0.37	0.63	0.05	0.77	0.71	159	4	6.0	3.3	1890	781
P3C	0.36	0.35	0.58	0.58	0.55	0.27	0.96	0.47	0.02	0.01	0.98	0.94	2	1	6.0	3.0	172	32
KYMEROCCLUS	0.65	0.56	0.67	0.60	0.21	0.09	0.56	0.43	0.12	0.04	1	1	17.0	11.0	2.82	2.0	2	2
CHAMELEOCCLUS	0.65	0.59	0.68	0.62	0.20	0.10	0.53	0.41	0.14	0.07	1	1	27	22	2.48	1.85	202	158
MOOSubClust	0.69	0.65	0.64	0.59	0.48	0.34	0.93	0.77	0.94	0.87	1	1	11.0	4.0	5.1	2.7	340	324

Table 4
Results for dataset *breast*: 33 dimensions, 2 classes, 198 objects.

	F_1		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.73	0.61	0.81	0.76	0.11	0.04	0.84	0.07	0.46	0.27	1.00	0.80	60	6	27.2	2.8	1E+06	37515
CLIQUE	0.67	0.67	0.71	0.71	0.02	0.02	0.40	0.40	0.26	0.26	1.00	1.00	107	107	1.7	1.7	453	453
MINECLUS	0.78	0.69	0.78	0.76	0.19	0.18	1.00	1.00	0.56	0.37	1.00	1.00	64	32	33.0	33.0	40359	29437
SUBCLU	0.68	0.51	0.77	0.67	0.02	0.01	0.54	0.04	0.27	0.24	1.00	0.82	357	5	2.0	1.0	5265	16
SCHISM	0.67	0.67	0.75	0.69	0.01	0.01	0.36	0.34	0.35	0.34	1.00	0.99	248	197	2.3	2.2	158749	114609
FIRE	0.49	0.03	0.76	0.76	0.03	0.00	0.05	0.00	1.00	0.01	0.76	0.04	11	1	2.5	1.0	250	31
PROCLUS	0.57	0.52	0.80	0.74	0.51	0.11	0.65	0.43	0.32	0.23	0.89	0.69	9	2	24.0	18.0	703	141
INSCY	0.74	0.55	0.77	0.76	0.02	0.00	0.24	0.11	0.60	0.39	0.97	0.74	2038	167	11.0	4.4	134373	63484
STATPC	0.41	0.41	0.78	0.78	0.16	0.16	0.33	0.33	0.29	0.29	0.43	0.43	5	5	33.0	33.0	5187	4906
P3C	0.63	0.63	0.77	0.77	0.04	0.04	0.19	0.19	0.36	0.36	0.85	0.85	28	28	6.9	6.9	6281	6281
CHAMELEOCCLUS	0.60	0.51	0.76	0.76	0.23	0.11	0.53	0.25	0.25	0.22	1	1	8	4	16.75	5.75	339	131
KYMEROCCLUS	0.66	0.57	0.79	0.76	0.18	0.14	0.56	0.51	0.31	0.25	1	1	19.0	13.0	12.36	9.26	8	7
MOOSubClust	0.68	0.63	0.56	0.50	0.27	0.19	0.42	0.34	0.97	0.76	1	1	9.0	5.0	7.4	4.6	1626	1318

Table 5Results for dataset *shape*: 17 dimensions, 9 classes, 160 objects, sample size = 120.

	F_1		Accuracy		CE		$RNIA$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.90	0.83	0.79	0.54	0.56	0.38	0.90	0.82	0.93	0.86	1.00	1.00	53	29	13.8	12.8	2E+06	86500
CLIQUE	0.31	0.31	0.76	0.76	0.01	0.01	0.07	0.07	0.66	0.66	1.00	1.00	486	486	3.3	3.3	235	235
MINECLUS	0.94	0.86	0.79	0.60	0.58	0.46	1.00	1.00	0.93	0.82	1.00	1.00	64	32	17.0	17.0	46703	3266
SUBCLU	0.36	0.29	0.70	0.64	0.00	0.00	0.05	0.04	0.89	0.88	1.00	1.00	3468	3337	4.5	4.1	4063	1891
SCHISM	0.51	0.30	0.74	0.49	0.10	0.00	0.26	0.01	0.85	0.55	1.00	0.92	8835	90	6.0	3.9	712964	9031
FIRES	0.36	0.36	0.51	0.44	0.20	0.13	0.25	0.20	0.88	0.82	0.45	0.39	10	5	7.6	5.3	63	47
PROCLUS	0.84	0.81	0.72	0.71	0.25	0.18	0.61	0.37	0.93	0.91	0.89	0.79	34	34	13.0	7.0	593	469
INSCY	0.84	0.59	0.76	0.48	0.18	0.16	0.37	0.24	0.94	0.87	0.88	0.82	185	48	9.8	9.5	22578	11531
STATPC	0.43	0.43	0.74	0.74	0.45	0.45	0.55	0.55	0.56	0.56	0.92	0.92	9	9	17.0	17.0	250	171
P3C	0.51	0.51	0.61	0.61	0.14	0.14	0.17	0.17	0.80	0.80	0.66	0.66	9	9	4.1	4.1	140	140
CHAMELEOCLUST	0.75	0.63	0.80	0.71	0.54	0.49	0.78	0.71	0.77	0.67	1	1	14	10	12.40	10.79	462	252
KYMEROCCLUS	0.82	0.72	0.86	0.79	0.57	0.53	0.86	0.80	0.83	0.77	1	1	19.0	16.0	13.5	12.56	101	91
MOOSubClust	0.15	0.11	0.93	0.85	0.20	0.18	0.37	0.28	0.97	0.89	1	1	25.0	12.0	6.0	3.9	2022	1981

Table 6Results for dataset *pendigits*: 16 dimensions, 10 classes, 7494 objects.

	F_1		Accuracy		CE		$RNIA$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
MINECLUS	0.87	0.87	0.86	0.86	0.48	0.48	0.89	0.89	0.82	0.82	1.00	1.00	64	64	12.1	12.1	780167	692651
DOC	0.52	0.52	0.54	0.54	0.18	0.18	0.35	0.35	0.53	0.53	0.91	0.91	15	15	5.5	5.5	178358	178358
CLIQUE	0.30	0.17	0.96	0.86	0.06	0.01	0.20	0.06	0.41	0.26	1.00	1.00	1890	36	3.1	1.5	67891	219
STATPC	0.91	0.32	0.92	0.10	0.09	0.00	0.67	0.11	1.00	0.53	0.99	0.84	4109	56	16.0	16.0	5E+06	3E+06
SCHISM	0.45	0.26	0.93	0.71	0.05	0.01	0.30	0.08	0.50	0.45	1.00	0.93	1092	290	10.1	3.4	5E+08	21266
CHAMELEOCLUST	0.71	0.51	0.74	0.59	0.51	0.30	0.78	0.49	0.68	0.58	1	1	14	10	12.40	7.21	4476	4226
KYMEROCCLUS	0.92	0.89	0.92	0.89	0.40	0.34	0.80	0.78	0.89	0.86	1	1	41.0	34.0	12.21	10.51	556	523
FIRES	0.45	0.45	0.73	0.73	0.09	0.09	0.33	0.33	0.31	0.31	0.94	0.94	27	27	2.5	2.5	169999	169999
PROCLUS	0.78	0.73	0.74	0.73	0.31	0.27	0.64	0.45	0.90	0.71	0.90	0.74	37	17	14.0	8.0	6045	4250
SUBCLU	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
P3C	0.74	0.74	0.72	0.72	0.28	0.28	0.58	0.58	0.76	0.76	0.90	0.90	31	31	9.0	9.0	2E+06	2E+06
INSCY	0.65	0.48	0.78	0.68	0.07	0.07	0.30	0.28	0.77	0.69	0.91	0.82	262	106	5.3	4.6	2E+06	1E+06
MOOSubClust	0.17	0.12	0.93	0.86	0.11	0.09	0.33	0.28	0.94	0.90	1	1	22.0	17.0	6.7	4.83	20969	19836

Table 7Results for dataset *diabetes*: 8 dimensions, 2 classes, 768 objects.

	F_1		Accuracy		CE		$RNIA$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.71	0.71	0.72	0.69	0.31	0.26	0.92	0.79	0.31	0.24	1.00	0.93	64	17	8.0	5.1	1E+06	51640
CLIQUE	0.70	0.39	0.72	0.69	0.03	0.01	0.14	0.01	0.23	0.13	1.00	1.00	349	202	4.2	2.4	11953	203
MINECLUS	0.72	0.66	0.71	0.69	0.63	0.13	0.89	0.58	0.29	0.17	0.99	0.96	39	3	6.0	5.2	3578	62
SUBCLU	0.74	0.45	0.71	0.68	0.01	0.01	0.01	0.01	0.14	0.11	1.00	1.00	1601	325	4.7	4.0	190122	58718
SCHISM	0.70	0.62	0.73	0.68	0.08	0.01	0.36	0.09	0.34	0.20	1.00	0.79	270	21	4.2	3.9	35468	250
FIRES	0.52	0.03	0.65	0.64	0.12	0.00	0.27	0.00	0.68	0.00	0.81	0.03	17	1	2.5	1.0	4234	360
PROCLUS	0.67	0.61	0.72	0.71	0.34	0.21	0.78	0.69	0.23	0.19	0.92	0.78	9	3	8.0	6.0	360	109
INSCY	0.65	0.39	0.70	0.65	0.37	0.11	0.45	0.42	0.44	0.15	0.83	0.73	132	3	6.7	5.7	112093	33531
STATPC	0.73	0.59	0.70	0.65	0.06	0.00	0.63	0.17	0.72	0.28	0.97	0.75	363	27	8.0	8.0	27749	4657
P3C	0.39	0.39	0.66	0.65	0.56	0.11	0.85	0.22	0.09	0.07	0.97	0.88	2	1	7.0	2.0	656	141
CHAMELEOCLUST	0.70	0.62	0.73	0.70	0.17	0.09	0.66	0.47	0.28	0.23	1	1	29	19	5.00	2.75	598	438
KYMEROCCLUS	0.69	0.64	0.73	0.70	0.21	0.08	0.55	0.40	0.25	0.21	1	1	16.0	13.0	3.69	2.87	3	3
MOOSubClust	0.76	0.72	0.66	0.51	0.37	0.28	0.67	0.58	0.91	0.80	1	1	9.0	4.0	5.6	3.7	1063	866

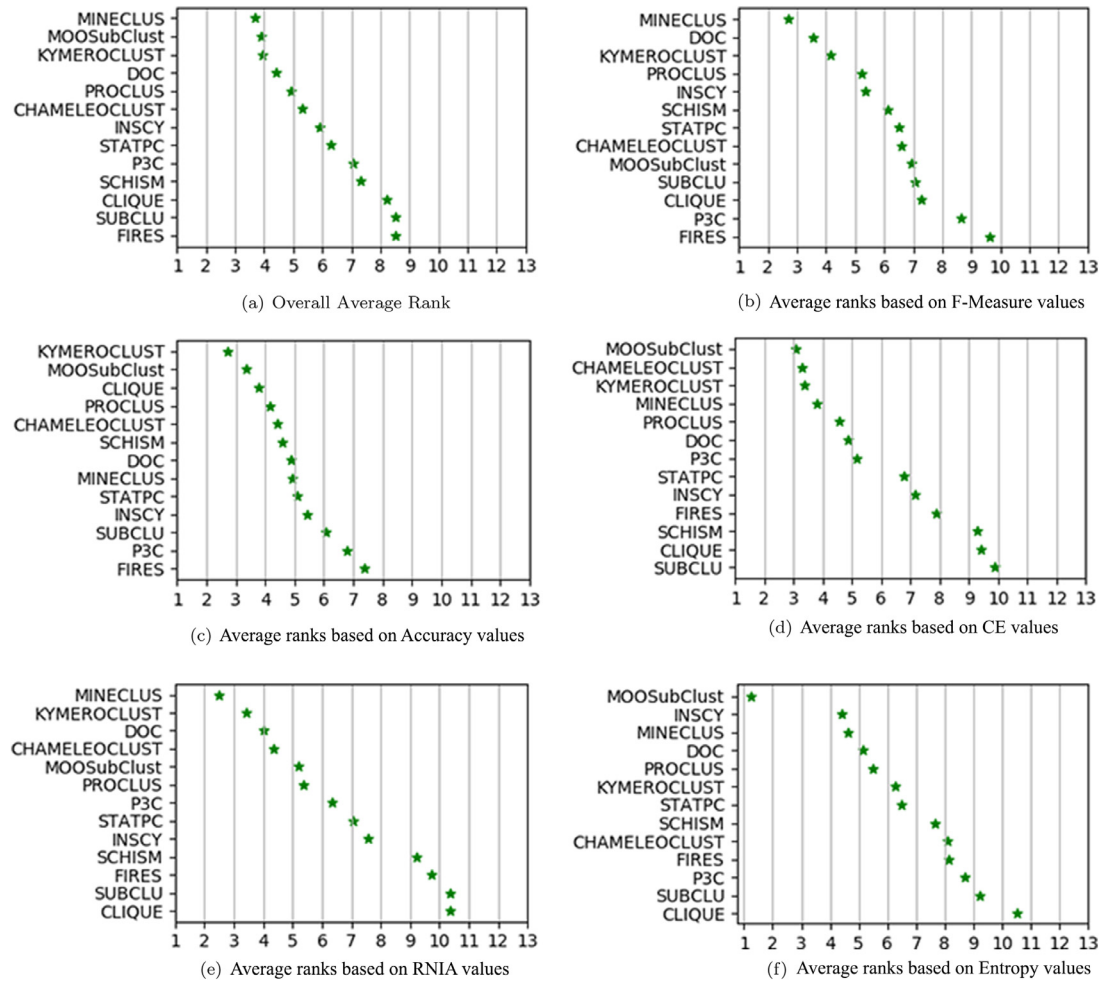
Table 8Results for dataset *glass*: 9 dimensions, 6 classes, 214 objects, sample size = 150.

	F_1		Accuracy		CE		$RNIA$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.74	0.50	0.63	0.50	0.23	0.13	0.93	0.33	0.72	0.50	0.93	0.91	64	11	9.0	3.3	23172	78
CLIQUE	0.51	0.31	0.67	0.50	0.02	0.00	0.06	0.00	0.39	0.24	1.00	1.00	6169	175	5.4	3.1	411195	1375
MINECLUS	0.76	0.40	0.52	0.50	0.24	0.19	0.78	0.45	0.72	0.46	1.00	0.87	64	6	7.0	4.3	907	15
SUBCLU	0.50	0.45	0.65	0.46	0.00	0.00	0.01	0.01	0.42	0.39	1.00	1.00	1648	831	4.9	4.3	14410	4250
SCHISM	0.46	0.39	0.63	0.47	0.11	0.04	0.33	0.20	0.44	0.38	1.00	0.79	158	30	3.9	2.1	313	31
FIRES	0.30	0.30	0.49	0.49	0.21	0.21	0.45	0.45	0.40	0.40	0.86	0.86	7	7	2.7	2.7	78	78
PROCLUS	0.60	0.56	0.60	0.57	0.13	0.05	0.51	0.17	0.76	0.68	0.79	0.57	29	26	8.0	2.0	375	250
INSCY	0.57	0.41	0.65	0.47	0.23	0.09	0.54	0.26	0.67	0.47	0.86	0.79	72	30	5.9	2.7	4703	578
STATPC	0.75	0.40	0.49	0.36	0.19	0.05	0.67	0.37	0.88	0.36	0.93	0.80	106	27	9.0	9.0	1265	390
P3C	0.28	0.23	0.47	0.39	0.14	0.13	0.30	0.27	0.43	0.38	0.89	0.81	3	2	3.0	3.0	32	31
CHAMELEOCLUST	0.43	0.28	0.57	0.50	0.43	0.26	0.88	0.55	0.46	0.36	1	1	8	4	7.50	4.75	195	95
KYMEROCCLUS	0.65	0.51	0.71	0.60	0.32	0.24	0.85	0.76	0.65	0.55	1	1	23.0	19.0	6.74	5.7	18	16
MOOSubClust	0.30	0.28	0.78	0.68	0.31	0.26	0.68	0.46	0.98	0.70	1	1	12.0	8.0	4.6	3.1	581	527

Table 9

Results for dataset vowel: 10 dimensions, 11 classes, 990 objects.

	F_1		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
DOC	0.49	0.49	0.44	0.44	0.14	0.14	0.85	0.85	0.58	0.58	0.86	0.86	64	64	10.0	10.0	120015	120015
CLIQUE	0.23	0.17	0.64	0.37	0.05	0.00	0.44	0.01	0.10	0.09	1.00	1.00	3062	267	4.9	1.9	523233	1953
MINECLUS	0.48	0.43	0.37	0.37	0.09	0.04	0.62	0.34	0.60	0.46	0.98	0.87	64	64	7.2	3.6	7734	5204
SUBCLU	0.24	0.18	0.58	0.38	0.04	0.01	0.39	0.04	0.30	0.13	1.00	1.00	10881	709	3.6	2.0	26047	2250
SCHISM	0.37	0.23	0.62	0.52	0.05	0.01	0.43	0.11	0.29	0.21	1.00	0.93	494	121	4.3	2.8	23031	391
FIRES	0.16	0.14	0.13	0.11	0.02	0.02	0.14	0.13	0.16	0.13	0.50	0.45	32	24	2.1	1.9	563	250
PROCLUS	0.49	0.49	0.44	0.44	0.11	0.11	0.53	0.53	0.65	0.65	0.67	0.67	64	64	8.0	8.0	766	766
INSCY	0.82	0.33	0.61	0.15	0.09	0.07	0.75	0.26	0.94	0.21	0.90	0.81	163	74	9.5	4.3	75706	39390
STATPC	0.22	0.22	0.56	0.56	0.06	0.06	0.12	0.12	0.14	0.14	1.00	1.00	39	39	10.0	10.0	18485	16671
P3C	0.08	0.05	0.17	0.16	0.12	0.08	0.69	0.43	0.13	0.12	0.98	0.95	3	2	7.0	4.7	1610	625
CHAMELEOCLUST	0.41	0.37	0.42	0.38	0.17	0.13	0.65	0.54	0.45	0.40	1	1	33	24	6.00	4.57	995	787
KYMEROCCLUS	0.53	0.48	0.53	0.47	0.16	0.14	0.75	0.70	0.56	0.52	1	1	50.0	45.0	6.82	6.3	364	339
MOOSubClust	0.24	0.22	0.86	0.79	0.17	0.14	0.48	0.43	0.95	0.89	1	1	24.0	10.0	4.4	3.7	7653	6307

**Fig. 2.** Average rankings of different clustering algorithms over all datasets with respect to combined primary evaluation metrics, F_1 , Accuracy, CE, RNIA and Entropy.

proposed algorithm *MOOSubClust* performs best. It also performs well with respect to Accuracy as it ranks 2nd on the list.

4.1.3. Runtimes

The same procedure was used to compare the execution times of the algorithms, rank 1 being the fastest. In *MOOSubClust* all the experiments (the Python implementations) were performed on 3.60 GHz CPU, Intel i7 processor, Ubuntu 16.04 LTS. The execution time is strongly linked to the parameter *sol* (Number of Solutions). The results presented in Tables 3–9 were computed with *sol* = 10 incurring a penalty of execution time increased by a factor of 10.

If the algorithm had implemented considering *sol* = 1 (only 1 solution), the execution time would have been reduced by 10 times. Fig. 3(c) shows the rank of the proposed algorithm corresponding to the execution time of a single solution.

4.1.4. Coverage

The same ranking method was used to rank the algorithms according to their coverage values. Coverage refers to the fraction of the objects that are part of any cluster. A rank value of 1 was associated to the highest coverage. *MOOSubClust* ranks 1st along with

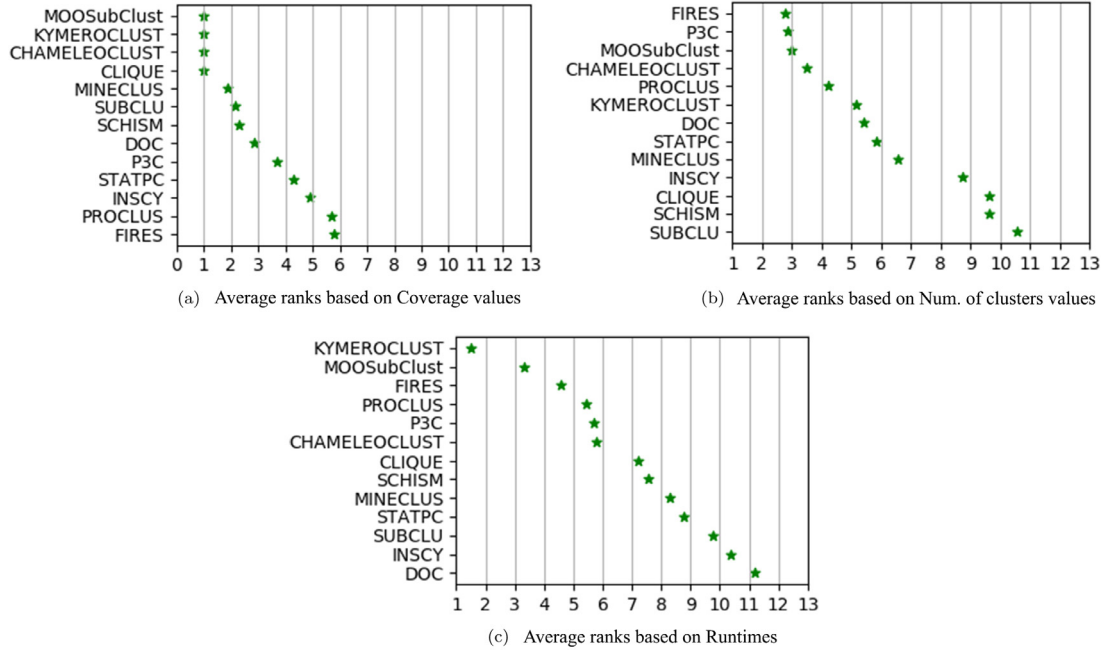


Fig. 3. Average rankings of different clustering algorithms over all datasets with respect to the Coverage, Number of clusters and Runtimes.

KymeroClust, ChameleoClust and Clique according to coverage as shown in Fig. 3(a).

4.1.5. Number of clusters

The rankings were computed in a similar way shown in [19] by computing the absolute value of the difference between the actual number of classes present in the dataset and the number of clusters produced by the algorithm. The smallest absolute value is given a rank of 1. The proposed algorithm ranks third with respect to the number of clusters as shown in Fig. 3(b).

4.1.6. Overall performance of proposed approach

The proposed algorithm ranks second with respect to the primary evaluation metrics. Though the algorithm MINECLUS tops in the list but it is time inefficient. Execution time is an important factor in any evaluation process and the proposed algorithm performs much better than MINECLUS with respect to runtime.

The cluster quality also highly depends on the generation of the number of clusters. Fig. 3(b) shows the superiority of the proposed algorithm in comparison with MINECLUS. Again, with respect to coverage, the objects in the dataset that belong to any of the output clusters are 97% for MINECLUS whereas 100% for MOOSubClust.

Furthermore, considering the individual primary evaluation metric, it is observed that, in comparison with MINECLUS, the algorithm MOOSubClust performs better with respect to Accuracy, CE, and Entropy. The proposed algorithm produced a balanced result with respect to both the primary as well as secondary evaluation metrics and thus its superiority is justified.

4.2. Results on synthetic datasets

This section presents the results obtained on 16 synthetic datasets used in [32]. The algorithm MOOSubClust was executed 10 times on each of the synthetic datasets. For each of the datasets, maximum and minimum values of various cluster quality evaluation metrics for the proposed algorithm are shown in Table 10. Again, for each evaluation metric, we plotted the best measured value obtained with respect to the average number of clusters (rounded to nearest integer) for 10 runs for each dataset shown

in Fig. 4. It is observed from the results that, for all the synthetic datasets, the average number of clusters identified by the proposed algorithm is very close to the actual number of clusters present in the dataset (i.e., 10) and for many datasets, the predicted average number of clusters equals the actual number of clusters. The algorithm, MOOSubClust, always found the average number of clusters between 7 and 13. Among the other algorithms except for ChameleoClust, P3C, and KymereoClust, rest identified the number of clusters between 5 and 50 and in few cases much more clusters were found as reported in [31]. ChameleoClust, P3C, and KymereoClust determine the number of clusters in the range of 9 to 16, 6 to 16 and 9 to 14 clusters, respectively, for these 16 synthetic datasets which are close to the actual number of clusters.

4.3. Advantages of multi-objective optimization

To elaborate in more detail we consider the data set vowel and plotted the accuracy values of all the solutions present in the final population identified by our proposed technique MOOSubClust. Different solutions encode different possible number of subspace clusters. The accuracy values attained by different solutions against the number of clusters present are plotted in Fig. 5. This figure clearly proves that the use of MOO indeed produces a large number of alternate solutions.

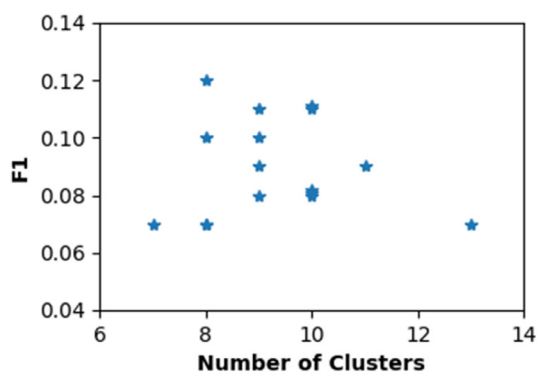
4.4. Performance of proposed algorithm on big data sets

In order to illustrate the advantages of using multiple objectives in the domain of subspace clustering, the proposed multi-objective optimization method is tested with a few big data sets collected from UCI [32]. The datasets selected are HTRU2 (17898 instances and 8 dimensions), Crowdsourced Mapping (10546 instances and 29 dimensions), MAGIC Gamma Telescope (19020 instances and 10 dimensions). In the proposed method the objectives functions that we have used are functions of compactness and separation in the form of XB-index and PBM-index. In order to illustrate the efficacy of optimizing multiple objective functions, results of a clustering algorithm optimizing a single cluster quality measure, intra-cluster compactness, on these big datasets are also

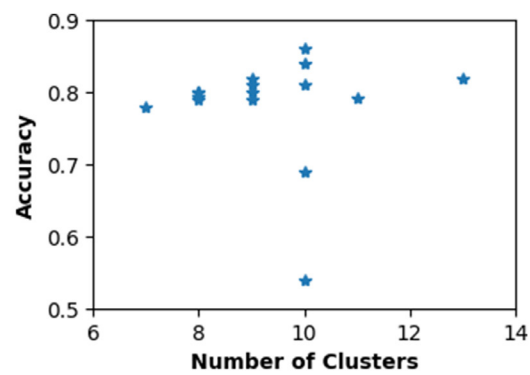
Table 10

Results of the proposed algorithm for the sixteen synthetic datasets.

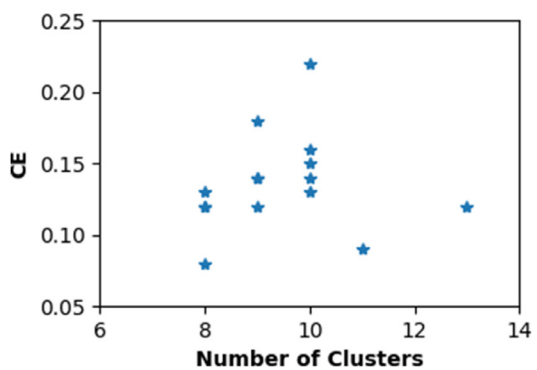
Dataset	F_1		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
S1500	0.11	0.08	0.81	0.77	0.13	0.04	0.31	0.15	0.97	0.92	12	8	4.9	3.0	7406	5463
S2500	0.12	0.05	0.79	0.70	0.12	0.05	0.32	0.18	0.98	0.71	13	7	5.6	3.6	8080	6636
S3500	0.10	0.06	0.80	0.70	0.12	0.06	0.33	0.22	0.96	0.91	9	6	4.9	4.2	8664	6668
S4500	0.10	0.05	0.79	0.75	0.12	0.06	0.30	0.19	0.93	0.85	9	5	5.6	3.7	12680	11735
S5500	0.11	0.05	0.81	0.72	0.14	0.07	0.29	0.23	0.96	0.68	11	8	4.7	3.5	14918	14132
D05	0.08	0.05	0.86	0.78	0.22	0.14	0.79	0.45	0.94	0.89	14	7	4.2	2.6	1709	1682
D10	0.08	0.06	0.82	0.65	0.18	0.09	0.76	0.37	0.91	0.84	15	6	7.0	3.8	3280	3152
D15	0.07	0.03	0.80	0.66	0.13	0.08	0.43	0.24	0.92	0.66	12	7	5.4	3.9	5376	5134
D20	0.08	0.05	0.84	0.76	0.14	0.07	0.32	0.20	0.96	0.76	13	7	5.9	3.3	5919	5801
D25	0.09	0.04	0.79	0.73	0.09	0.05	0.29	0.17	0.94	0.74	17	7	6.4	4.3	8283	7882
D50	0.07	0.03	0.80	0.63	0.08	0.04	0.27	0.18	0.94	0.63	12	7	10.25	5.6	31837	327840
D75	0.07	0.02	0.78	0.62	0.04	0.03	0.19	0.09	0.93	0.81	11	7	11.4	7.8	56361	53904
N10	0.07	0.04	0.82	0.73	0.12	0.07	0.30	0.14	0.96	0.71	23	6	5.7	3.0	15843	13054
N30	0.09	0.05	0.79	0.73	0.14	0.09	0.37	0.21	0.95	0.91	16	6	7.0	5.2	14796	14295
N50	0.08	0.06	0.69	0.62	0.15	0.08	0.26	0.16	0.97	0.86	19	6	7.8	5.3	9298	8653
N70	0.11	0.07	0.54	0.48	0.16	0.13	0.38	0.23	0.96	0.91	18	6	8.9	6.0	9980	9447



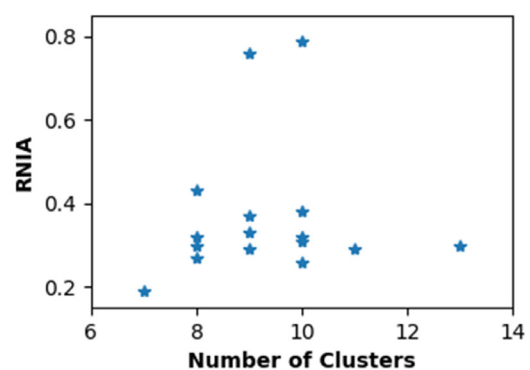
(a) Number of Clusters vs F1



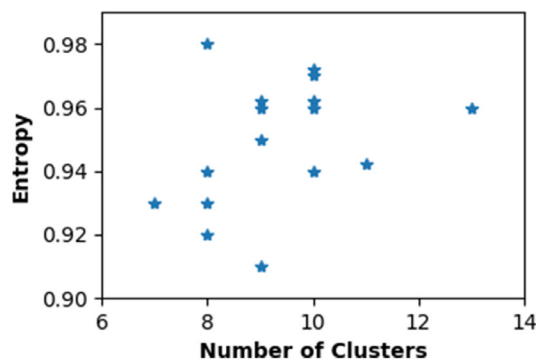
(b) Number of Clusters vs Accuracy



(c) Number of Clusters vs CE



(d) Number of Clusters vs RNIA



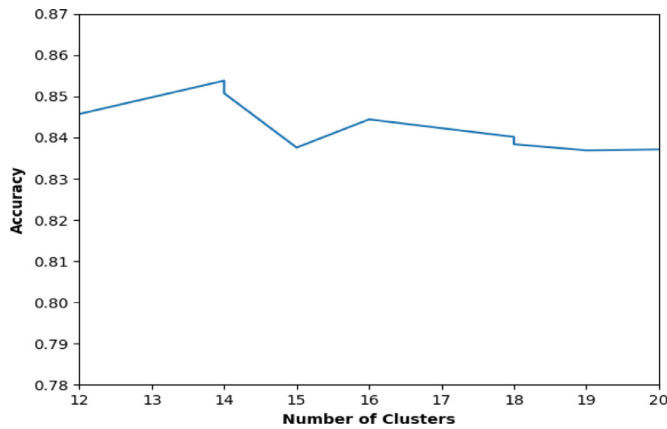
(e) Number of Clusters vs Entropy

Fig. 4. Values of primary evaluation metrics vs average number of clusters for the 16 synthetic datasets.

Table 11

Comparison of the proposed method using multiple objectives and single objective on big data.

Dataset	F-measure		Accuracy		CE		RNIA		Entropy	
	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO
HTRU	0.51	0.66	0.91	0.91	0.20	0.23	0.44	0.89	0.97	0.99
Crowd Source	0.09	0.13	0.69	0.70	0.07	0.20	0.24	0.42	0.99	0.99
Magic	0.63	0.63	0.67	0.69	0.20	0.32	0.39	0.51	0.76	0.97

**Fig. 5.** Accuracy values of subspace clusters with respect to the number of clusters on the final population attained by MOOSubClust algorithm for vowel.

reported. Both the algorithms, one optimizing multiple cluster validity indices (MOO) and the other optimizing single cluster quality measure (SOO) are executed 10 times and the best values corresponding to different evaluation metrics and datasets are stored for comparison as shown in Table 11. From Table 11, the benefits of optimizing multiple objectives over a single objective are clearly observed. For almost all the cases, the proposed multi-objective optimization methods show the superiority (values are shown in bold) in comparison with single objective version.

4.5. Comparison with low rank and sparse subspace clustering representation

Sparse representation subspace clustering (SSC) [17] and Low-Rank representation subspace clustering (LRR) [18] techniques segment the data drawn from a union of multiple linear or affine subspaces. In SSC, each data vector is sparsely represented individually with respect to a dictionary formed by all other data points. Whereas LRR finds the lowest rank representation among all the candidates that represent all data vectors as the linear combination of the bases in a dictionary. In the case of SSC and LRR, apriori knowledge of the number of subspaces is required. However, the proposed method, MOOSubClust doesn't require any apriori information about the number of subspaces and also it is flexible, unlike SSC and LRR where the number of subspaces is kept fixed. In LRR and SSC, the quality of the subspaces produced is measured by the classification accuracy. Table 12 compares the classification accuracies of the proposed method with respect to SSC and LRR.

From Table 12, it is clearly visible that the proposed algorithm performs better with respect to classification accuracy for all the 7 real-life datasets.

4.6. Real-life application of the proposed algorithm

The data analysis of gene expression through micro-array technology helps us to solve many problems like medical diagnosis, gene expression profiling etc. A major step of this analysis is the automatic identification of a group of genes that follow similar expression patterns. In order to form groups of such similar patterns, clustering techniques can be used in the field of micro-array data analysis. However, the conventional clustering techniques did not perform sufficiently well when the number of attributes increases [34] as those may miss the patterns that present over a locality of attributes. Subspace clustering techniques can be used as remedy to such situation, in which a subset of genes which show similar patterns over a subset of attributes are grouped together unlike conventional clustering where grouping is done considering all the attributes.

The applicability of the proposed algorithm is tested with the gene expression level datasets Human Large B Cell Lymphoma [34] and Yeast [34]. The subspace clusters produced by the proposed algorithm are evaluated using Mean squared residue (MSR) [34], Row variance (RV) [34] and BI-index [35]. Higher the value of RV whereas lower the values of MSR and BI indicate better performance. The developed algorithm is compared with various state-of-the-art algorithms such as MOPSOB [36], MOGAB [35], RWB [37], SGAB [38], BiVisu [39], Bi-max [40], OPSM [41], Cheng and Church (CC) [34], ISA [42]. The comparisons with the state-of-the-art algorithms are shown in Tables 13 and 14 for Human and Yeast dataset, respectively. The results shown in Table 13 and Table 14 clearly illustrate the benefits of the proposed algorithm.

5. Discussion

The proposed algorithm is compared with a set of state-of-the-art algorithms, among them our algorithm, MOOSubClust ranks second after combining the average ranks of different primary evaluation metrics. The only algorithm that performs better than the proposed algorithm by a very small margin is MINECLUS. Also, there is no ever winning algorithm or paradigm, each having its own advantages or interests. Although, MINECLUS ranks better than MOOSubClust but out of 8 evaluation metrics (combining both primary and secondary metrics), the proposed algorithm defeats the algorithm MINECLUS in 6 cases (Figs. 2(c), (d), (f) and 3(a)–(c)). The algorithm MINECLUS performs better than the proposed

Table 12

Comparison of the proposed method with LRSC and SSC for real-life datasets with respect to Accuracy.

Datset	Breast	Diabetes	Glass	Liver	Pendigits	Shape	Vowel
Proposed	0.56	0.66	0.78	0.64	0.93	0.93	0.86
SSC	0.53	0.51	0.29	0.51	0.12	0.68	0.12
LRSC	0.50	0.59	0.35	0.57	0.71	0.14	0.48

Table 13

Results of subspace clustering obtained by different algorithms on Human Large B Cell Lymphoma dataset.

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best(Max)	Average	Best(min)
MOOSubClust	2233.09	1281.92	10877.92	17273.32	0.31	0.07
MOPSOB	927.47	745.25	4348.2	8745.65	0.213	0.09
SGAB	855.36	572.54	2222.19	5301.83	0.5208	0.3564
MOGAB	801.37	569.23	2378.25	5377.51	0.4710	0.2283
RWB	1185.69	992.76	1698.99	3575.40	0.7227	0.3386
OPSM	1249.73	43.07	6374.64	11854.63	0.2520	0.1024
BiVisu	1680.23	1553.43	1913.24	2468.63	0.8814	0.6537
Bimax	387.71	96.98	670.83	3204.35	0.7120	0.1402
CC	1078.38	779.71	2144.14	5295.42	0.5662	0.2298
ISA	2006.83	245.28	4780.65	14682.47	0.6300	0.0853

Table 14

Results of subspace clustering obtained by different algorithms on Yeast dataset.

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best(Max)	Average	Best(min)
MOOSubClust	438.76	204.65	1395.34	4285.24	0.18	0.06
MOPSOB	218.54	200.58	789.85	1254.56	0.28	0.16
SGAB	198.88	138.75	638.23	3605.93	0.4026	0.2714
MOGAB	185.85	116.34	932.04	3823.46	0.3329	0.2123
RWB	295.81	231.28	528.97	1044.37	0.5869	0.2788
OPSM	320.39	118.53	1083.24	3804.56	0.3962	0.0012
BiVisu	290.59	240.96	390.73	775.41	0.7770	0.3940
Bimax	32.16	5.73	39.53	80.42	0.4600	0.2104
CC	204.29	163.94	801.27	3726.22	0.3822	0.0756
ISA	281.59	125.76	409.29	1252.34	0.7812	0.1235

algorithm with respect to F-measure and RNIA (Fig. 2(b) and (e)). RNIA and CE are two evaluation metrics specially developed for subspace clustering evaluation. In case of real-world datasets, since we do not know the actual subspaces corresponding to a cluster, instead, we use the original feature set for comparison purpose. Therefore, although the RNIA value produced by our approach is lower than that of MINECLUS but it appears in the list of top few. Moreover CE value attained by the proposed approach ranks top in the list, but in general CE values of different algorithms are low. The proposed algorithm ranks below in the list for the metric F-measure as it doesn't perform well for a few datasets such as pendigits, shape, glass and vowel. The possible reason could be the absence of clustering structure in these datasets. However, if we calculate the average ranks of the algorithms after considering combined primary metrics except F-measure, the proposed algorithm tops the list. The algorithm performs well corresponding to Accuracy and Entropy. The metric entropy generally gives a better result when the number of clusters is more and it gives the best result when each object forms a cluster. However, the proposed algorithm attains better result and ranks top with respect to entropy with a limited range of clusters. Again, in case of synthetic dataset, we compare with the number of clusters produced by different algorithms. The range of the number of clusters produced by the proposed algorithm is 7 – 13. Therefore, the maximum difference between the actual number of clusters and the produced cluster range is 3 whereas it is 4, 7 and 10 for KymereoClust, ChameleoClust and P3C, respectively.

6. Conclusions and future works

In this paper, we present a multi-objective optimization based subspace clustering algorithm, namely *MOOSubClust*. The algorithm builds upon the search capability of evolvable genome structure and multi-objective optimization. Two existing cluster validity indices, namely PBM-index and XB-index, are modified to handle the perception of subspace clustering. The values of these two in-

dices are simultaneously optimized using the search capability of any multi-objective optimization technique. The efficacy of the proposed technique is shown for generating subspace clustering from ten real-world data sets and sixteen synthetic data sets. It was observed that the proposed algorithm combines the advantages of evolutionary based techniques and multi-objective optimization. In a part of the paper, the efficacy of the proposed technique is also shown for bi-clustering of gene expression data.

The algorithm developed here is inspired by the steps of KymereoClust algorithm. In future, we would like to enhance the genetic operations of the proposed framework to develop a new multi-objective based evolutionary subspace clustering algorithm aiming to get even better results. We also plan to develop some online subspace clustering approaches which can handle data streams.

Acknowledgements

Dr. Sriparna Saha gratefully acknowledges the support from SERB Women in Excellence Award 2018 of Science and Engineering Research Board (SERB) of Department of Science & Technology, Govt of India for conducting this research.

References

- [1] K. Kailing, H.-P. Kriegel, P. Kröger, Density-connected subspace clustering for high-dimensional data, in: Proceedings of the 2004 SIAM International Conference on Data Mining, SIAM, 2004, pp. 246–256.
- [2] A. Adolfsson, M. Ackerman, N.C. Brownstein, To cluster, or not to cluster: an analysis of clusterability methods, Pattern Recognit. 88 (2019) 13–26.
- [3] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM Sigkdd Explor. Newslett. 6 (1) (2004) 90–105.
- [4] W. Gao, L. Hu, P. Zhang, Class-specific mutual information variation for feature selection, Pattern Recognit. 79 (2018) 328–339.
- [5] K. Zheng, X. Wang, Feature selection method with joint maximal information entropy between features and class, Pattern Recognit. 77 (2018) 20–29.
- [6] Y. Zhu, K.M. Ting, M.J. Carman, Grouping points by shared subspaces for effective subspace clustering, Pattern Recognit. (2018) 230–244.
- [7] Y. Liu, L. Jiao, F. Shang, An efficient matrix factorization based low-rank representation for subspace clustering, Pattern Recognit. 46 (1) (2013) 284–292.

- [8] P. Zhu, W. Zhu, Q. Hu, C. Zhang, W. Zuo, Subspace clustering guided unsupervised feature selection, *Pattern Recognit.* 66 (2017) 364–374.
- [9] G. Gan, M.K.-P. Ng, Subspace clustering using affinity propagation, *Pattern Recognit.* 48 (4) (2015) 1455–1464.
- [10] X. Chen, Y. Ye, X. Xu, J.Z. Huang, A feature group weighting method for subspace clustering of high-dimensional data, *Pattern Recognit.* 45 (1) (2012) 434–446.
- [11] G. Gan, M.K.-P. Ng, Subspace clustering with automatic feature grouping, *Pattern Recognit.* 48 (11) (2015) 3703–3713.
- [12] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, J.S. Park, Fast algorithms for projected clustering, in: *ACM SIGMOD Record*, 28, ACM, 1999, pp. 61–72.
- [13] C.C. Aggarwal, P.S. Yu, Finding generalized projected clusters in high dimensional spaces, 29, ACM, 2000.
- [14] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, 27, ACM, 1998.
- [15] C.-H. Cheng, A.W. Fu, Y. Zhang, Entropy-based subspace clustering for mining numerical data, in: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 1999, pp. 84–93.
- [16] S. Goil, H. Nagesh, A. Choudhary, Mafia: Efficient and scalable subspace clustering for very large data sets, in: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 1999, pp. 443–452.
- [17] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 2790–2797.
- [18] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 663–670.
- [19] S. Peignier, C. Rigotti, G. Beslon, Subspace clustering using evolvable genome structure, in: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ACM, 2015, pp. 575–582.
- [20] S. Peignier, Subspace clustering on static datasets and dynamic data streams using bio-inspired algorithms, INSA, Lyon, 2017 Ph.D thesis, Université de Lyon.
- [21] C. Knibbe, A. Coulon, O. Mazet, J.-M. Fayard, G. Beslon, A long-term evolutionary pressure on the amount of noncoding dna, *Mol. Biol. Evol.* 24 (10) (2007) 2344–2353.
- [22] S. Saha, S. Bandyopadhyay, A generalized automatic clustering algorithm in a multiobjective framework, *Appl. Soft Comput.* 13 (1) (2013) 89–108.
- [23] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, *IEEE Trans. Evol. Comput.* 12 (3) (2008) 269–283.
- [24] M.K. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, *Pattern Recognit.* 37 (3) (2004) 487–501.
- [25] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (8) (1991) 841–847.
- [26] S. Ohno, *Evolution by Gene Duplication*, Springer Science & Business Media, 2013.
- [27] G.C. Conant, K.H. Wolfe, Turning a hobby into a job: how duplicated genes find new functions, *Nat. Rev. Genet.* 9 (12) (2008) 938.
- [28] C.C. Aggarwal, A. Hinneburg, D.A. Keim, On the surprising behavior of distance metrics in high dimensional space, in: *International conference on database theory*, Springer, 2001, pp. 420–434.
- [29] N. Srinivasan, K. Deb, Multi-objective function optimisation using non-dominated sorting genetic algorithm, *Evol. Comp.* 2 (3) (1994) 221–248.
- [30] F.-A. Fortin, M. Parizet, Revisiting the NSGA-II crowding-distance computation, in: *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, ACM, 2013, pp. 623–630.
- [31] E. Müller, S. Günnemann, I. Assent, T. Seidl, Evaluating clustering in subspace projections of high dimensional data, *Proc. VLDB Endowment* 2 (1) (2009) 1270–1281.
- [32] M. Lichman, *UCI Machine Learning Repository*, 2013.
- [33] A. Patrikainen, M. Meila, Comparing subspace clusterings, *IEEE Trans. Knowl. Data Eng.* 18 (7) (2006) 902–916.
- [34] Y. Cheng, G.M. Church, Bicustering of expression data., in: *Ismb*, 8, 2000, pp. 93–103.
- [35] U. Maulik, A. Mukhopadhyay, S. Bandyopadhyay, Finding multiple coherent bi-clusters in microarray data using variable string length multiobjective genetic algorithm, *IEEE Trans. Inf. Technol. Biomed.* 13 (6) (2009) 969–975.
- [36] J. Liu, Z. Li, F. Liu, Y. Chen, Multi-objective particle swarm optimization biclustering of microarray data, in: *Bioinformatics and Biomedicine*, 2008. BIBM'08. IEEE International Conference on, IEEE, 2008, pp. 363–366.
- [37] F. Angiulli, C. Pizzuti, Gene expression biclustering using random walk strategies, in: *International Conference on Data Warehousing and Knowledge Discovery*, Springer, 2005, pp. 509–519.
- [38] A. Tanay, R. Sharan, R. Shamir, Discovering statistically significant biclusters in gene expression data, *Bioinformatics* 18 (suppl_1) (2002) S136–S144.
- [39] K.-O. Cheng, N.-F. Law, W.-C. Siu, T. Lau, Bivisu: software tool for bicluster detection and visualization, *Bioinformatics* 23 (17) (2007) 2342–2344.
- [40] J.A. Hartigan, Direct clustering of a data matrix, *J. Am. Stat. Assoc.* 67 (337) (1972) 123–129.
- [41] A. Ben-Dor, B. Chor, R. Karp, Z. Yakhini, Discovering local structure in gene expression data: the order-preserving submatrix problem, *J. Comput. Biol.* 10 (3–4) (2003) 373–384.
- [42] J. Ihmels, S. Bergmann, N. Barkai, Defining transcription modules using large-scale gene expression data, *Bioinformatics* 20 (13) (2004) 1993–2003.

Dipanjoy Paul received the M.Tech degree in Information Technology from IIST, Shibpur (2016). He is working toward the Ph.D degree in the CSE Department, Indian Institute of Technology, Patna. His research interest includes Pattern Recognition, Bioinformatics, Multiobjective Optimization, Genetic Algorithms. He is a student member of the IEEE.

Sriparna Saha is currently a Faculty Member of the Department of Computer Science and Engineering, Indian Institute of Technology Patna, India. Her current research interests include pattern recognition, multiobjective optimization and biomedical information extraction. Her h-index is 19 and total citation count of her papers is more than 2000 (according to Google scholar). She is the recipient of the Lt Rashi Roy Memorial Gold Medal from ISI Kolkata, Google India Women in Engineering Award, 2008, Junior Humboldt Research Fellowship, NASI Young Scientist Platinum Jubilee Award 2017, BIRD Award 2017, IEI Young Engineers' Award 2017 etc.

Jimson Mathew received the master in computer engineering from Nanyang Technological University, Singapore, and the Ph.D degree in computer engineering from the University of Bristol, Bristol, United Kingdom. He is currently an associate professor in the Computer Science and Engineering Department, Indian Institute of Technology Patna, India. He has held positions with the Centre for Wireless Communications, National University of Singapore, Bell Laboratories Research Lucent Technologies North Ryde, Australia, Royal Institute of Technology KTH, Stockholm, Sweden and Department of Computer Science, University of Bristol, United Kingdom. His research interests include fault-tolerant computing, computer arithmetic, hardware security, VLSI design and automation, and design of nano-scale circuits and systems. He is a senior member of the IEEE.