

Improved subspace clustering algorithm using multi-objective framework and subspace optimization

Dipanjoyoti Paul ^{a,*}, Sriparna Saha ^a, Jimson Mathew ^a

^a Indian Institute of Technology, Patna, Bihar, India

ARTICLE INFO

Article history:

Received 6 February 2020

Revised 18 April 2020

Accepted 26 April 2020

Available online 11 May 2020

Keywords:

Subspace clustering

Multi-objective Optimization (MOO)

Intra-Cluster Compactness (ICC)

Feature Non-Redundancy (FNR)

Feature Per Cluster (FPC)

ABSTRACT

Subspace clustering technique divides the data set into different groups or clusters where each cluster comprises of objects that share some similar properties. Again, the feature sets or the subspace features that are used to represent clusters are different for different clusters. Moreover, in subspace clustering, the grouping of similar objects and the subspace feature set representing that group are identified simultaneously. In evolutionary-based machine learning problems, two critical measures to determine the quality of the generated clusters are *compactness* within and *separation* between the clusters. However, the distance-based separation between two clusters may not be useful in the context of subspace clustering, as the clusters may belong to two different subspaces. Again, in the case of subspace clustering, the selection of relevant subspace features plays a primary role in generating good quality subspace clusters. Therefore, the proposed approach optimizes the subspace features by considering two new objective functions, *feature non-redundancy* (FNR) and *feature per cluster* (FPC) represented in the form of PSM-index. Another objective function, intra-cluster compactness (ICC-index), is modified and used to optimize the compactness among objects within the cluster. Finally, an evolutionary-based multi-objective subspace clustering technique is developed in this paper optimizing these validity indices. A new mutation operator, namely *duplication and deletion* along with the modified version of the *exogenous genetic material uptake*, are developed to explore the search space effectively. The developed algorithm is tested on sixteen synthetic data sets and seven standard real-life data sets for identifying different subspace clusters. Again, to show the effectiveness of using multiple objectives, the algorithm is also tested on three big data sets and a MNIST data set. Also, an application of the proposed method is shown in bi-clustering the gene expression data. The results obtained by the proposed algorithm are compared against some state-of-the-art methods. Experimentation reveals that the proposed algorithm can take advantage of its evolvable genomic structure and the newly defined objective functions on the multi-objective based framework.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Clustering refers to the concept of detecting patterns amongst objects and grouping them into different categories such that objects that follow similar patterns belong to the same cluster and the two objects that do not have a common pattern are assigned to different clusters (Kailing, Kriegel, & Kröger, 2004). Nowadays, a massive volume of data is being generated with a large set of features by different application domains. In case of high dimensional data sets, detecting patterns amongst objects based on some similarity measures is not effective which is known

as the *curse of dimensionality* (Parsons, Haque, & Liu, 2004) problem. However, it has appeared that all the features that describe an object are not relevant for clustering. Also, the presence of redundant or irrelevant features may degrade the cluster quality. Therefore, subspace clustering techniques, an emerging research area, become popular in handling such situations when a subset of features are relevant for a subset of data points (Zhu, Ting, & Carman, 2018). Subspace clustering approach is used to solve many machine learning problems such as SAP (Gan & Ng, 2015), FG-K-means (Chen, Ye, Xu, & Huang, 2012), AFG-K-means (Gan & Ng, 2015) to name a few. It is also useful in solving problems like low rank representation (Liu, Jiao, & Shang, 2013), unsupervised feature selection (Zhu, Zhu, Hu, Zhang, & Zuo, 2017) etc. The subset of features that are used to represent a subspace cluster is known as subspace feature. Fig. 1 shows the subspace cluster representa-

* 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 ORCID ID: 0000-0001-9079-7524.

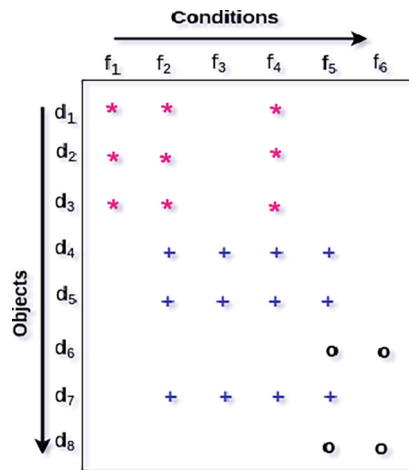


Fig. 1. Representation of subspace clustering.

tion consisting of three subspace clusters such that cluster* [Objects={d₁, d₂, d₃}, Subspace-feature={f₁, f₂ and f₄}], cluster+ [Objects={d₄, d₅, d₇}, Subspace-feature={f₂, f₃, f₄ and f₅}] and cluster_o [Objects={d₆, d₈}, Subspace-feature={f₅ and f₆}]

In the literature, lots of work in the area of subspace clustering have been carried out using bottom-up approaches and top-down approaches. Algorithms such as ORCLUS (Aggarwal & Yu, 2000), PROCLUS (Aggarwal, Wolf, Yu, Procopiuc, & Park, 1999) etc. are based on top-down approaches which initially starts developing clusters using the full feature set and keep on removing a feature incrementally until the clustering performance reaches to a plateau. ENCLUS (Cheng, Fu, & Zhang, 1999), MAFA (Goil, Nagesh, & Choudhary, 1999) etc. are few bottom-up approaches where each algorithm starts with searching dense regions for each feature space. It adds a dimension in its subspace when the dense region after adding a feature satisfies a predefined threshold (Parsons et al., 2004). Again, the algorithms, CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998), SUBCLU (Kailing et al., 2004), use density-based approach to generate clusters. SUBCLU finds out the dense regions using the DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) algorithm whereas, in CLIQUE, presence of more than a predefined number points within a region is called dense. The approaches (Passalis & Tefas, 2019; Huang et al., 2014; Zografos, Ellis, & Mester, 2013) etc. are the discriminative based clustering approaches where (Huang et al., 2014 & Zografos et al., 2013) are specially designed to produce subspace clusters. The algorithm (Zografos et al., 2013), generates subspace clusters by using quadratic classifier trained from an unlabeled data, i.e., clustering by classification whereas in (Huang et al., 2014), a 3-order tensor is used in the computation of compactness and separation and to determine the importance of a feature in a cluster. (Yamaguchi, Irie, Kawanishi, & Kashino, 2019), is another subspace clustering algorithm which is graph based and generates subspace clusters in two steps: graph construction and graph clustering. In the graph construction step, an affinity matrix is created which reports the similarity amongst objects belonging to the same subspace and in the graph clustering step, clusters are created from the affinity matrix. The subspace clustering approaches (Huang et al., 2014; Zografos et al., 2013; Yamaguchi et al., 2019) constructed the subspace clusters by optimizing a single objective function. Another disadvantage of these approaches is that, the number of clusters that the algorithms will generate has to be provided in the form of input. Again, the algorithms (Passalis & Tefas, 2019 & Bouveyron & Brunet, 2012) are conventional clustering based approaches. (Passalis & Tefas, 2019) is a regularized subspace learning based clustering approach. It uses similarity-based sub-

space learning to improve the clustering by learning a projection function that improves the intra-cluster compactness and the inter-cluster separability. The algorithm (Bouveyron & Brunet, 2012), adapts the traditional mixture model for modeling and classifying data in a latent discriminative subspace. A few deep-learning based approaches, (Xie, Girshick, & Farhadi, 2016; Hershey, Chen, Le Roux, & Watanabe, 2016; Tian, Zhou, & Guan, 2017) etc. are also developed to generate clusters. The methods (Xie et al., 2016 & Tian et al., 2017), simultaneously learn feature representations and perform the cluster assignment under the same framework and the method (Hershey et al., 2016) is specifically designed to address the problem “cocktail-party” source separation in a deep learning framework. These deep-learning based approaches are designed to perform conventional clustering and requires the number of clusters to be generated as an input. In recent years, Chameleoclust (Peignier, Rigotti, & Beslon, 2015) and KymeroClust (Peignier, 2017) are two evolutionary approaches developed to solve the subspace clustering problem. These algorithms take the advantage of the dynamic evolution of genome structure which leads to variable ratio of functional versus non-functional elements (Knibbe, Coulon, Mazet, Fayard, & Beslon, 2007) or variable genome length. Inspired by (Crombach & Hogeweg, 2007), in the algorithm Chameleoclust also, a sample of objects are represented in the form of tuples known as *course grained* genome. The genome containing the tuples are mapped at the phenotype level to define core point locations or phenotypic traits, in different dimensions, which then help in generating a subspace cluster. Each gene of the genome is not represented explicitly, but it only keeps track of the number of repeated genes involved in each biological function of a phenotypic trait. KymeroClust is also defined in the form of phenotype; consists of set of core points and a genotype; the number of biological functions associated with each element of the phenotype. The duplication-divergence operator is used to modify both genotype and phenotype representations.

Most of the previous state-of-the-art subspace clustering algorithms implicitly or explicitly optimize a single cluster quality measure. KymeroClust and ChameleoClust are two evolutionary-based techniques which use some distance-based measures to find out the similarity between objects. The objects are assigned to a particular cluster based on the minimum distance between an object and a cluster center. Thus, these algorithms optimize cluster-compactness as the only objective function and are limited in capturing only those subspace clusters which follow some particular shapes or properties. Again, in case of subspace clustering, selection of subspace feature for a cluster has a massive impact on the quality of clusters. Therefore, optimization of subspace features must also be considered. Concepts of multi-objective optimization (MOO) can be incorporated in order to avoid the limitation of optimizing a single objective function in the subspace clustering process and is able to generate good quality subspace clusters.

Simultaneously when one or more objective functions need to be optimized, the concept of MOO technique can be incorporated. Multi-objective optimization (Saha & Bandyopadhyay, 2013; Bandyopadhyay & Saha, 2012) aims at finding one or more optimal solutions by optimizing a set of conflicting objective functions. MOO technique is very popular and useful while dealing with many machine learning problems. However, there are many clustering problems, (Bandyopadhyay & Saha, 2012; Saha & Bandyopadhyay, 2010), where MOO technique is used and is found to be effective. In this paper, MOO technique is incorporated by considering intra-cluster compactness, non-redundant subspace features and features per cluster as the objective functions.

In this paper, we proposed a subspace feature optimization based multi-objective subspace clustering algorithm, using evolu-

tionary technique. Here, an individual or a solution is defined as a phenotype which consists of a set of core points called phenotypic traits. Each core point in the phenotype is associated with a set of objects defined in its own subspace. Again, a genotype is defined as the number of repeated genes involved in each biological function of each phenotypic trait. An important phenomenon of evolutionary approach is that objects in the genome have to undergo some duplication or replacement operation for evolution. Authors in (Conant & Wolfe, 2008; Ohno, 1999) investigated and suggested that duplication of genes has an important role in the area of evolution. However, once a gene is duplicated, it performs some useful tasks only after it is mutated. In order to effectively explore the search spaces, two different mutation operators are introduced. These are *duplication and deletion* and *exogenous genetic material uptake*. These mutation operators are developed in such a way that it can adapt to the new environment very easily i.e., the process to adapt to the new environment is very fast. Due to environment change, some bacteria may reach to *competent state*, uptaking external DNA that sometimes carries locally useful biological functions. *Exogenous genetic material uptake* is based on such concept where direct addition of environmental information takes place. The *duplication and deletion* mutation is carried out using the duplication-divergence operator. In order to make some useful changes in the phenotype, duplication and deletion are performed in a single operation where multiple genes are selected either to duplicate or to get deleted. To measure the wellness of genes that are surrounded by some phenotypic trait, a new cluster validity index, namely PSM-index, is defined in this paper. Another validity index, ICC-index based on intra-cluster compactness is also used to handle the subspace clusters effectively. PSM-index and ICC-index are optimized simultaneously using the search capability of MOO. ICC-index measures the appropriateness of generated subspace clusters in terms of intra-subspace cluster similarity and PSM-index measures the appropriateness of selected subspace features of the clusters. The objective is to minimize both ICC-index and PSM-index in order to obtain a good set of subspace clusters. Concepts of non-dominated sorting (Deb, Pratap, Agarwal, & Meyarivan, 2002) and crowding distance (Fortin & Parizéau, 2013) operator can be applied to select the best subset of solutions from the available solutions. The proposed multi-objective based subspace clustering algorithm is applied on twenty-three real-life and artificial data sets to show its efficacy. Also, the efficiency of using multiple objectives is verified using three big data sets and one categorical data set. A real-life application of the proposed method is shown in bi-clustering the gene expression data. When compared against some existing subspace clustering approaches with respect to different performance metrics, the superiority of utilizing MOO concept and optimization of subspace feature set in solving subspace clustering problem evidently revealed. The key-contributions of the proposed research work are as follows.

- This paper integrates the concept of multi-objective optimization in evolutionary based technique to solve the problem of subspace clustering.
- It reports the first attempt in simultaneously optimizing the subspace feature set and subspace compactness to identify good quality subspace clusters.
- A new validity index namely PSM-index specially designed for the selection of optimal subspace feature set is introduced.
- A new mutation operator is introduced and an existing mutation operator is modified and those are used to explore the search space effectively.
- A real-life application of the proposed subspace clustering method is shown in bi-clustering the gene expression data.

2. SOSubClust algorithm

The algorithm, subspace optimization based subspace clustering, *SOSubClust* is based on evolutionary based technique and multi-objective optimization framework. The proposed method, generates subspace clusters automatically while giving importance to the selection of appropriate subspace feature.

2.1. Problem formulation

The subspace clustering problem can be formulated as a multi-objective optimization problem.

- Given:
 - A dataset of size $|S|$ such as $D = \{d_1, d_2, \dots, d_s\}$ where
 - Each data is described by F features such that $F = \{f_1, f_2, \dots, f_F\}$
 - A set of m objective functions $VI_1, VI_2, \dots, VI_m$ where VI_i is the cluster validity index measured on the partitions obtained after clustering the data set. The validity indices are used to optimize the generated subspace clusters.
- Find:
 - Some alternative partitionings (Z) of the data set into N (N is flexible) subspace clusters.
 - * The set of data-points D , is grouped into N clusters, $\{Z_1, Z_2, \dots, Z_N\}$
 - * $Z_i = \{d_1^i, d_2^i, \dots, d_{n_i}^i\}$, n_i : number of data points in cluster i , d_j^i : j th element of cluster i , and
 - * The subspace feature set of each cluster, $\{F_1, F_2, \dots, F_N\}$
 - * $F_i = \{f_1^i, f_2^i, \dots, f_{k_i}^i\}$, k_i : number of features present in subspace cluster i and f_j^i : j th feature of subspace feature set i . The features that are not present in a subspace-feature set hold the value '0'.
 - * Produced non-overlapping clusters, i.e., $\bigcup_{i=1}^N Z_i = D$ and $Z_i \cap Z_j = \phi$ for all $i \neq j$
 - which simultaneously optimize the objective functions. The simultaneous optimization of these objectives provides a Pareto optimal front.

2.2. Model representation

In meta-heuristic based optimization strategies there are different ways of representing the clustering solution. The proposed method is a center-based approach and therefore, our primary task is to find the optimal set of cluster centers. A model consists of phenotype and genotype represented in the form of matrices, with SC_{max} rows (upper bound to the number of clusters) and F columns (total number of features) as shown by matrix U and V respectively. The rows of the phenotype represent the phenotypic traits (i.e., the coordinates of the cluster centers along different features) and the rows in genotype represent the number of genes or biological functions associated with each of the elements in the phenotypic trait i.e., the associated weights of different centers along different features.

$$U = \begin{matrix} & \begin{matrix} f_1 & & f_F \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_{SC_{max}} \end{matrix} & \begin{pmatrix} d_{11} & & d_{1F} \\ d_{21} & \dots & d_{2F} \\ d_{31} & & d_{3F} \\ \vdots & & \vdots \\ d_{SC_{max}1} & & d_{SC_{max}F} \end{pmatrix} \end{matrix} \quad V = \begin{matrix} & \begin{matrix} f_1 & & f_F \end{matrix} \\ \begin{matrix} a_{SC_{max}1} & & a_{SC_{max}F} \end{matrix} & \begin{pmatrix} a_{11} & & a_{1F} \\ a_{21} & \dots & a_{2F} \\ a_{31} & & a_{3F} \\ \vdots & & \vdots \end{pmatrix} \end{matrix}$$

In the above matrices, $d_{ij} \in \mathbb{R}$, represents the co-ordinate value of i th cluster along j th feature and $a_{ij} \in \mathbb{N}$ represents the number of genes associated with the co-ordinate d_{ij} ; where $i \in \{1, 2, \dots, SC_{max}\}$ and $j \in \{1, 2, \dots, F\}$. Again, when a feature is not relevant to a particular phenotypic trait or cluster center; it is represented with zero (i.e., $d_{ij}, a_{ij} = 0.0$). However, if all the features of a cluster center hold zero, then it doesn't represent a cluster.

Initialization: The model is initialized by selecting a cluster center uniformly and traversing all its features, so that either a co-ordinate value along with a feature get selected for updation or not, with a probability $\frac{1}{2}$. The co-ordinates that are selected are updated with a value (by uniformly selecting an object and a feature), and the rest of the co-ordinates are initialized with zero. The co-ordinates that are updated, their corresponding genotype value is also increased by one. This process is repeated $NumClasses$ times, i.e., until the number of genes in the initial genotype reaches to $NumClasses * \lceil \frac{F}{2} \rceil$. The model gets updated to a new model by applying mutation operators to the phenotype.

2.3. Mutation operator

The phenotype in the model consists of a set of cluster centers and in evolutionary terms, it can also be called an individual. However, in order to generate a new set of cluster centers and to explore the search space, mutation operators are applied to individuals to generate new individuals or off-springs. A new mutation operator, *duplication and deletion* and modified version of *exogenous genetic material uptake* (Peignier, 2017), are used in this approach. At each iteration, both the mutation operators *exogenous genetic material uptake* and *duplication and deletion* are applied to an individual only once and in the same order. Both the mutation operators initially calculate the weight of the genotype V , and if the weight is greater than $2 * SC_{max}$, it first selects a gene randomly from the genotype and deletes it. After deletion, if the number of genes in the genotype becomes zero, the corresponding value in the phenotype is also marked as zero.

2.3.1. Exogenous genetic material uptake

When the environment changes, some bacteria reaches the state of *competent*, uptaking external DNA that sometimes carries locally useful biological functions. This corresponds to the direct addition of environmental information to modify the location of an existing phenotypic trait in the phenotype or to create a new phenotypic trait (i.e., a new cluster center).

To perform this mutation operator, an object d' is selected uniformly from the sample data set. A subset of features f' , is selected from the total set of features. The number of features selected in f' , is an integer drawn uniformly between the range $[1; \frac{F}{2}]$, limiting the size to at most half of the total number of features. Only the co-ordinates of d' along the feature f' will be used to perform mutation. Each co-ordinate of d' is being injected as a biological function.

This mutation operator can either create a new phenotypic trait or update an existing phenotypic trait. A new trait is created based on the probability $\left(\frac{f'_{size}}{f'_{size} + size(V)}\right)$. $size(V)$ and f'_{size} are the total number genes present in the genotype V and size of f' , respectively. This ratio makes the creation probability higher when the genotype size is small. As the genotype size may increase with an increase in time and when the algorithm reaches near to optimal solution, the chances of creating a new cluster will be low and therefore limiting huge changes in the model. In case of creation, the features of $f \in f'$ are updated with the co-ordinate value of d' along the features, f . The corresponding genotype value is also increased by one.

If no new cluster is created, an existing cluster (c) is selected based on the probability $\frac{|f' \cap F_c|}{|F_c|}$. F_c is the subspace feature representing the cluster center c and $||$ represents the size of the set. Only the features of $f \in f'$ of cluster c are need to be updated. If the co-ordinates of $f \in f'$ is a zero, it is updated with the new co-ordinate; otherwise, it is updated with the mean value of the previous and the new co-ordinate. The gene count in the genotype for that co-ordinate is also increased by one.

2.3.2. Duplication and deletion

The *duplication and deletion* mutation operator is based on the duplication-divergence operator. The duplication of genes has an important role in the evolutionary process as mentioned by the authors in (Ohno, 2013; Conant & Wolfe, 2008). However, in order to perform some useful task by the duplicated genes, they have to undergo some mutation and give birth to new biological functions. Therefore, duplication and deletion have to be chosen in a non-additive manner. In this operator, deletion and duplication of genes are performed in a single operation.

This mutation operator is performed on the phenotype after applying exogenous genetic material uptake. Let μ_{dd} be a parameter representing the duplication-deletion rate, then for a phenotype, the number of duplication-deletion events is drawn from a binomial distribution $\beta(N, \mu_{dd})$, where N is the total number of clusters. For each event, a cluster center is selected uniformly and the operation is carried out as follows.

Let c be the cluster drawn uniformly for this event, then we select a subset of features f' from the subspace feature of cluster c . Let the size of f' be τ , drawn uniformly between the interval $[1; F_c^{size}]$. F_c^{size} is the size of F_c ; the subspace feature of cluster c . Next, we select a number δ drawn uniformly in $\{-\tau, \dots, -1, 1, \dots, \tau\}$ (excluding 0). If the value δ is negative, that indicates the deletion of δ elements and a positive value indicates the duplication of δ elements.

To perform a duplication operation, only the co-ordinates of c along features f' are used. Initially, an existing cluster center is selected uniformly for updation and η_1 (drawn uniformly from the interval $[1; \delta]$) number of co-ordinates along the features f' are updated. Next, another center is chosen uniformly and η_2 (selecting uniformly between $[1; \delta - \eta_1]$) co-ordinates are updated along remaining features of f' and so on. The process is repeated until all the δ elements are duplicated. However, the co-ordinate values of center c that are used for duplication are passed through uniform distribution $\mathbb{U}(d_{c_f}, 2)$, which slightly changes its value and the corresponding genotype value is decreased by 1. Again, the co-ordinates of uniformly selected centers that were selected for updation, if the value is zero, it is updated with the new co-ordinate, otherwise, it is updated with the mean of the previous and the new co-ordinate value. The corresponding genotype value is also increased by one. The deletion operation is performed in a similar fashion by selecting the cluster centers uniformly until δ elements are deleted. The co-ordinate value selected for deletion is replaced by zero and the corresponding genotype value is also marked as zero.

2.4. Objective functions

In the evolutionary-based machine learning algorithm, the top solutions from the set of available solutions are selected based on the objective values. The objective functions that have been used in this approach can be categorized into two parts. One is the optimization of the objects, i.e., finding the optimum membership of the objects to the cluster and the other is the optimization of the subspace feature. The objective function *Compactness*

(ICC-index) is used to find the optimum membership of the objects. Again, in case of subspace clustering problem, selection of the subspace feature set that are used to represent clusters play an important role in generating and optimizing the clusters. *Feature non-redundancy* and *features per cluster* are the two objective functions used in this approach in order to choose the relevant subspace features for clusters. These two objective functions are represented in the form of an index known as *Paul Saha Mathew-index* (PSM-index).

2.4.1. Intra-cluster compactness

Intra-cluster compactness measures how close the objects of a cluster are from its center. It uses the distance-based measurement; therefore, less the distance, more compact the cluster is and therefore better the degree of membership to the clusters are. Moreover, in evolutionary-based subspace clustering, an individual or a solution represented by the phenotype U contains many cluster centers (i.e., N of SC_{max}) such that each center C_i is associated with a subspace feature F_i of F . The Intra-cluster compactness (ICC-index) can be computed as

$$ICC(D, U) = \left[\frac{1}{N} * \sum_{i=1}^N \frac{1}{|n_i|} \sum_{s \in \mu_i} dis(d_s, C_i) \right]$$

Where, μ_i is the membership degree for the cluster i and $|n_i|$ is the size of the cluster i . Again, the distance between an object (d_s) and cluster center (C_i) is calculated based on Manhattan distance and is defined as $dis(d_s, C_i) = \sum_{f \in F_i} |d_{sf} - C_{if}| + \sum_{f \in F \setminus F_i} |d_{sf} - \mu_f|$. $| \cdot |$ measures the absolute value and μ_f is the average distance from data points to original cluster center.

2.4.2. Feature non-redundancy

The subspace feature that used to represent a cluster has a huge impact on the quality of the subspace clusters. Good quality subspace clusters emphasize the selection of distinct subspace feature sets (Hund et al., 2016). This is basically to avoid the subspace clusters represented by highly similar subset of features. Consider two subspace feature representing two clusters as $F_1 = \{f_1, f_2, f_3\}$ and $F_2 = \{f_1, f_2, f_3, f_4\}$. Many features of F_1 and F_2 are in common, (in this case $F_1 \subset F_2$), having such subspace features, where most of the features are in common, provide redundant information. Therefore, avoiding such a subspace feature improves the selection of the feature set and improves the quality of the subspace clusters. So, it would be efficient to choose the subspace features such that the intersection between any two subspace features are less. Therefore, the objective function *feature non-redundancy* (FNR) has to be minimized and is computed as

$$FNR(U) = \frac{1}{NC_2} \sum_{i=1}^{N-1} \sum_{j=i+1}^N |(F_i \cap F_j)|$$

2.4.3. Features per cluster

Subspace clustering algorithms are meant to reduce the number of features representing a cluster. However, if the number of features representing a cluster is almost equal to the actual number of features, i.e., say if $|F|$ is the total number of features and a cluster is represented by $\sim (|F| - 1) - (|F| - 3)$ features, then it is not suitable for subspace clustering problems (Hund et al., 2016). Also, if the number of features representing a cluster is very less i.e., only $\sim 1-3$ features when F is large, in that case they can not provide any valuable information regarding cluster and are of no use (Hund et al., 2016). The ideal value for the number of features representing a subspace cluster can neither be too low nor too high and it should be a value near to the half of the total number of features. Therefore, we consider the optimum number of features that

can be present in the subspace feature as the median of the maximum and the minimum number of features that a subspace feature can have i.e., 1 and F , respectively and the median value will be $\frac{F+1}{2}$. Therefore, the objective function *features per cluster* (FPC) should be minimized and it is defined as

$$FPC(U) = \left[\frac{\sum_{i=1}^N |F_i|}{N} - \frac{F+1}{2} \right]$$

where $| \cdot |$ represent the size of the set. The objective functions that are described in this section are represented in the form of validity indices such as ICC-index and PSM-index. The objective function, *intra-cluster compactness*, is represented as ICC-index, which optimizes the compactness of the objects. The other two objective functions, *feature non-redundancy* and *feature per cluster*, are to be minimized and therefore their sum should also be minimized. The scaled sum of these two objective values are then represented in the form of PSM-index which optimizes the selection of subspace features.

$$PSM - index = \left[\frac{2 * FPC(U) + FNR(U)}{F} \right]$$

To illustrate with an example, let us consider, a sample data set D' of size 8 described by 4 features $F = \{f_1, f_2, f_3, f_4\}$ such as $d_1 = \{2.5, 0.6, 5.4, 2.8\}$, $d_2 = \{3.4, 2.7, 4.5, 1.9\}$, $d_3 = \{4.5, 5.6, 2.1, 2.2\}$, $d_4 = \{5.1, 4.6, 2.8, 2.4\}$, $d_5 = \{4.4, 4.8, 3.8, 2.3\}$, $d_6 = \{3.3, 3.6, 2.9, 2.0\}$, $d_7 = \{5.1, 3.7, 1.4, 3.1\}$, $d_8 = \{2.4, 0.5, 3.5, 2.8\}$. Again, assume that cluster centers at any time instance 't' are $m_1 = \{3.0, 0.0, 5.3, 0.0\}$, $m_2 = \{0.0, 1.7, 3.3, 0.0\}$, $m_3 = \{2.0, 2.3, 4.5, 0.0\}$; therefore, the corresponding subspace features of the clusters are $F_1 = \{f_1, f_3\}$, $F_2 = \{f_2, f_3\}$, $F_3 = \{f_1, f_2, f_3\}$. The subspace feature containing the value zero are not present in the feature-space. The objects are assigned to a particular cluster based on the minimum Manhattan distance between an object and the cluster center. Therefore, the corresponding membership degrees of the clusters m_1, m_2 and m_3 are $\mu_1 = [1, 0, 0, 0, 0, 0, 1]$, $\mu_2 = [0, 0, 1, 0, 0, 0, 1, 0]$, $\mu_3 = [0, 1, 0, 1, 1, 1, 0, 0]$ respectively.

The normalized compactness within a cluster is computed as the ratio of the summation of the distances between all the points to their cluster centers to the number of points in that cluster. In the above case the normalized compactness of clusters m_1, m_2 and m_3 are 4.85, 8.05, 6.825, respectively. Therefore the average normalized compactness or ICC-value is $\frac{(4.85+8.05+6.825)}{3} = 6.575$. Again, the feature non-redundancy can be calculated as $|F_1 \cap F_2| + |F_1 \cap F_3| + |F_2 \cap F_3|$. Therefore, the normalized value of feature redundancy is $\frac{(1+2+2)}{3} = 1.67$ and the value of features per cluster is $\left(\frac{\sum_{i=1}^N |F_i|}{N} - \frac{F}{2} \right) = \left(\frac{2+2+3}{3} - \frac{4}{2} \right) = 0.33$. Therefore, the value of the indices at time 't' are (6.575, 0.583).

2.5. Subspace cluster generation

In each iteration of the algorithm, when a phenotype is updated, the subspace clusters corresponding to the data set are created by assigning each object to a particular cluster center of the phenotype. We calculate the membership degree by assigning each object of the data set to a cluster with respect to which the distance between the object and a cluster center is minimum [Lines 5–12 of Algorithm 1]. In order to calculate the distance, we consider only those features that are relevant to the cluster center, i.e., the subspace feature of that center. The objects that are present in a subspace cluster are the members of the cluster and the feature set that is used to represent the cluster is the subspace feature. From the developed subspace clusters, we calculate the validity indices which help us in determining the qualities of the subspace clusters.

Algorithm 1:Generate Subspace Clusters (D, U)

```

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

```

We use Manhattan distance, as the clusters are represented by a subset of features only. The distance between an object (d_s) and a cluster center (C_i) in subspace clustering can be calculated using the Eq. 1.

$$dis(d_s, C_i) = \sum_{f \in F_i} |d_{sf} - C_{if}| + \sum_{f \in F \setminus F_i} |d_{sf} - \mu_f| \quad (1)$$

where F_i is the subspace feature of C_i and μ_f is the mean distance of all the objects to the origin.

2.6. Non-dominating Sorting and Crowding Distance

In case of clustering, in a multi-objective scenario, while dealing with multiple objective values, a case may arise where few objectives may favor the solution whereas others may oppose the solution. Non-dominating sorting (Deb et al., 2002) and crowding distance (Fortin & Parizéau, 2013) operators can be used to overcome such situation. Non-dominating sorting sorts the solutions based on their ranks by comparing each solution with all other solutions. Solutions that are not dominated by any other solution are assigned to front $Front_1$. Next, all the remaining solutions i.e., the solutions except the solutions that are in $Front_1$, are compared with each others. All non-dominated solutions are then assigned to front $Front_2$. This process continues until all the solutions become the part of a front, as shown in Fig. 2.

In the proposed approach, *SOSubClust*, we start with 10 solutions. After performing mutation operators, we receive another 10 solutions. Now, to choose the best 10 solutions out of 20 solutions, non-dominated sorting and crowding distance operators are applied. The non-dominated sorting procedure returns the fronts containing solutions in the decreasing order of their dominance and the solutions are selected in a sequence from $Front_1$, $Front_2$ and so on. Continuing this, a situation may arise when a subset of solutions from a particular front $Front_i$, which are non dominated to each other, need to be chosen. Crowding-distance (Fortin & Parizéau, 2013), as shown in Fig. 2, helps us to resolve such a problem and selects solutions that are spaced in a less dense region.

2.7. Overview of the proposed algorithm

Different steps of the proposed subspace clustering technique are shown in Fig. 3 and enumerated below.

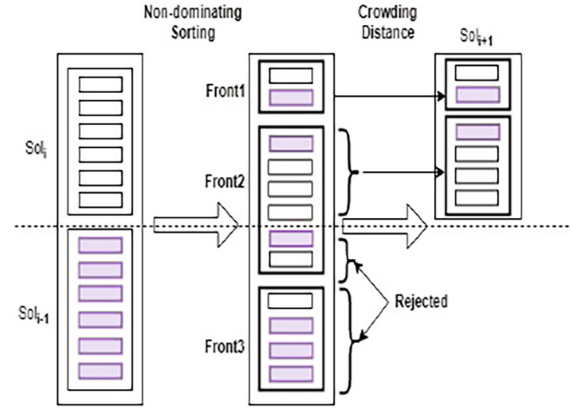


Fig. 2. Non-dominated sorting & crowding distance.

- The algorithm takes inputs as, data set D , maximum possible subspace clusters, SC_{max} , initial number of solutions, sol and the number of iterations, $Iter$. The algorithm also selects a sample data set D' of size X , chosen uniformly from the whole data set.
- The algorithm initializes sol number of models in a fashion described in [Section 2.2]. Initial subspace clusters are being generated with the initial set of models. The corresponding objective values of the models (also called a solution) are also computed.
- In order to get the effect of the whole data set, at each iteration, an object from sample data is replaced uniformly and then the mutation operators are applied on each individual in a sequential manner [Refer to Section 2.3] to get a new set of models from the previous models.
- Next, the subspace clusters are generated [Section 2.5] on the current sample data set. The corresponding objective values (also called as a solution) of each of the new models are calculated with respect to validity indices [Refer Section 2.4].
- The new solutions are then combined with the previous solutions to get $(2 * sol)$ number of solutions. These solutions are then passed through non-dominated sorting and crowding distance operations [Section 2.6].
- Non-dominated sorting, sorts the solutions according to their non-dominance and selects the best sol number of solutions, the corresponding models are also retrieved.
- This process is repeated $Iter$ number of iterations, then the algorithm returns the final best sol number of solutions, where each phenotype contains a set of subspace cluster centers with their corresponding subspaces. Then, for all the phenotypes, subspace clusters are generated on the whole data set to generate the final subspace clusters. Finally, the algorithm reports the solution with the best results.

3. Experimental setup

The developed algorithm is tested with standard real and synthetic data sets collected from (Lichman, 2013). The various evaluation measures presented in (Müller, Günnemann, Assent, & Seidl, 2009), are also used as reference frameworks in order to compare the proposed algorithm, *SOSubClust*, with state-of-the-art algorithms.

3.1. Data sets & parameter settings

Seven benchmark real-life data sets collected from (Lichman, 2013) are used to evaluate the performance of the proposed

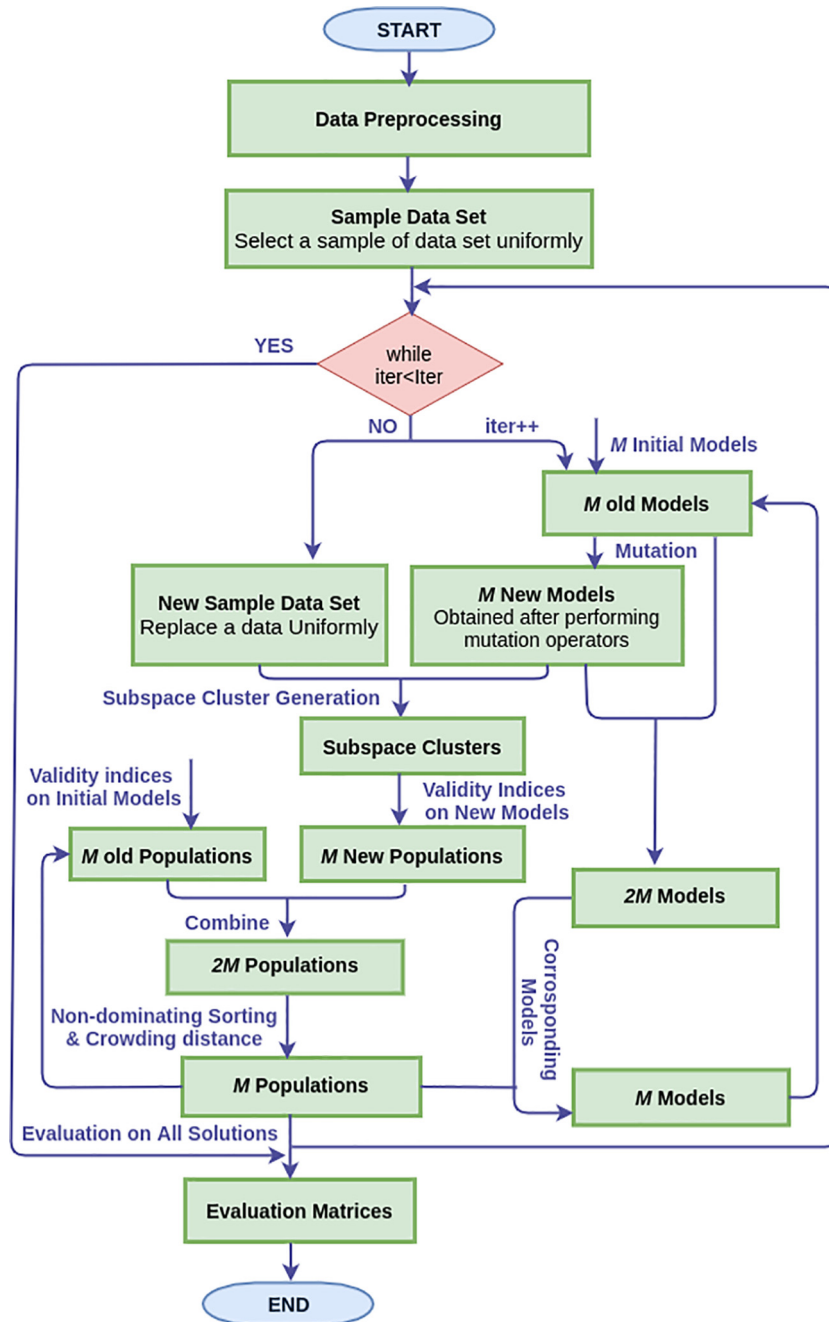


Fig. 3. Overview of the proposed algorithm SOSubClust.

algorithm. The data sets are *liver*, *breast*, *shape*, *glass*, *vowel*, *pendigits*, *diabetes* described in Table 1. The proposed algorithm, *SOSubClust*, is also executed on 16 synthetic data sets (Lichman, 2013) as described in Table 2. For the synthetic data sets, true clusters and their subspaces are known.

In the real-life data sets, only the information about the class label is available and the proposed algorithm provides the cluster label information. Again, in case of clustering, a class may be split into many clusters. Based on this notion, a similar approach as developed in (Peignier, 2017) is followed in order to define the parameters. The expected number of clusters is defined as, three times the number of classes, i.e., $ExpectedClusters = 3 * NumClasses$. To reduce the complexity of the algorithm, instead of considering whole data set in the optimization process, a sample of data set, D' of size X is considered. The sample size is an important parameter and it must be chosen in such a way that, the selected sample

Table 1
Characteristics of the real-life data sets.

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

data of size X , should have the impact of the overall data set. Therefore, the sample size is defined as $X = 25 * ExpectedClusters$. A different criterion is used in this algorithm to find the parameter, upper bound of the clusters, SC_{max} , which is defined in a fashion

Table 2
Description of synthetic data sets.

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

such that it should allow each class to traverse all the features at least once so that either a feature will be present or absent in the initial subspace features with probability $\frac{1}{2}$. Therefore, the maximum permissible subspace cluster SC_{max} is computed as $Num_Classes * \lceil \frac{F}{2} \rceil$. Again, there is no primary criteria for the selection of the value of the parameter, number of iterations denoted by *Iter*. In this algorithm we considered the number of iterations as 2000 as it is sufficient for clustering all the data sets. The duplication-deletion rate, μ_{dd} is considered as 0.005.

For synthetic data sets, the number of expected clusters is considered to be 10, therefore in a similar fashion, the sample size was set to $X=25 * 10$ and the maximum size of the model SC_{max} was set to $SC_{max}=3 * 10$.

3.2. Evaluation measures

The quality of the generated subspace clusters by the proposed method is compared with the many state-of-the-art algorithms using different evaluation measures. The state-of-the-art approaches that are considered for comparison are SUBCLU (Kailing et al., 2004), STATPC (Sembiring, Zain, & Embong, 2010), SCHISM (Sequeira & Zaki, 2005), PROCLUS (Aggarwal et al., 1999), P3C (Sembiring et al., 2010), MINECLUS (Yiu & Mamoulis, 2003), INSCY (Assent, Krieger, Müller, & Seidl, 2008), FIRES (Kriegel, Kroger, Renz, & Wurst, 2005), DOC (Procopiu, Jones, Agarwal, & Murali, 2002), CLIQUE (Agrawal et al., 1998), KYMERO-CLUST (Peignier, 2017) and CHAMELEOCLUST (Peignier et al., 2015). Again, the standard evaluation measures presented in (Müller et al., 2009) are considered in this paper. These evaluation measures are F-Measure (Müller et al., 2009), Accuracy (Müller et al., 2009), Clustering Error (CE) (Patrikainen & Meila, 2006), RNIA (1-RNIA) (Patrikainen & Meila, 2006) and Entropy (1-Entropy) (Müller et al., 2009). Along with these primary evaluation metrics, the results are also compared with respect to some secondary evaluation metrics such as coverage, average features of the subspace clusters, number of subspace clusters, and execution time.

4. Experimental results

The effectiveness of the developed algorithm is tested with 16 synthetic data sets and 7 real-life data sets and in order to compare its performance, many state-of-the-art algorithms are considered. However, from the comparison it is observed that, for some evaluation metrics, some algorithms are dominating, whereas for other evaluation metrics, other algorithms are dominating. In this sec-

tion, the advantages of the proposed algorithm and how it could be beneficial compared to the state-of-the-art algorithms are explained.

4.1. Results on real-life data sets

The proposed algorithm is executed 10 times on each of the real-life data sets. Every time, the algorithm outputs 10 solutions and the best solution is considered for further evaluation. From the 10 best solutions, from 10 executions, we report the maximum and minimum scores with respect to each of the evaluation metrics as presented in Table 3 & Table 4. However, in order to compare with different state-of-the-art algorithms, a rank-based comparison is made using different aggregations.

In order to compute the rank, we consider both the *maximum* and *minimum* scores of each algorithm with respect to different evaluation metrics. The rank of each algorithm is calculated with respect to individual primary metrics, such as *F-measure*, *Accuracy*, *CE*, *RNIA*, *Entropy* and combining all the primary evaluation metrics together. For each measure, two rank values of each algorithm are calculated, one for *maximum score* and another for *minimum score*. The overall ranking of an algorithm was then simply the average of these two rank values. The rank of the algorithms ranges from 1 to 13, where rank 1 being the algorithm with the highest quality. Fig. 4 shows the rank of all the primary evaluation metrics.

4.1.1. Analysis of performance with respect to primary evaluation metrics

Comparing the developed algorithm with respect to the state-of-the-art algorithms, it is observed that, the proposed algorithm, *SOSubClust*, ranks *first* with respect to the combined primary evaluation metrics, shown in Fig. 4(a). However, while comparing with respect to the individual primary evaluation metrics, proposed algorithm ranks *second* with respect to *F-Measure* (Fig. 4(b)), *Accuracy* (Fig. 4(c)) and *RNIA* (Fig. 4(e)) whereas it ranks *first* with respect to *CE* (Fig. 4(d)). It can be observed that, in all the cases, the proposed approach either ranks first or second, except *Entropy*, where it ranks *fifth*. This is because, a suitable range of number of clusters is produced by the proposed algorithm, and we know that entropy performs well with an increase in the number of clusters. The number of clusters produced by the algorithm *INSCY* is huge, and therefore it ranks *first* with respect to the metric *entropy*. However, although the algorithm *MINECLUS* ranks first with respect to two individual primary metrics, but it is not that impressive with respect to other metrics.

4.1.2. Analysis of performance with respect to secondary evaluation metrics

We compute the rank of the proposed algorithm combining all the secondary evaluation metrics such as *Coverage*, *Runtime* and *Number of Clusters*. The proposed algorithm again ranks *first* in the list (Fig. 5(a)). Moreover, with respect to the individual secondary metrics, *coverage* and *number of cluster*, the proposed algorithm ranks *first* and with respect to *runtime*, it ranks *second*. However, the details of each metrics are described below and shown in Fig. 5.

4.1.2.1. Runtime. The execution time of the proposed approach has a huge impact on the parameter *sol* (number of solutions). More the value of *sol*, the more time the algorithm will take to complete its execution. However, the execution of all the solutions could have also been carried out parallelly. The results shown in Table 3 & 4 were computed with *sol* = 10 incurring a penalty of execution time increased by a factor of 10. Therefore, considering a parallel computing scenario, the rank of the algorithm with respect to runtime [Shown in Fig. 5(d)] is calculated after reducing the execution

Table 3Results for the data sets *Breast*, *Diabetes*, *Glass*, *Liver* with respect to primary and secondary metrics.

Dataset Breast	F-Measure		Accuracy		CE		RNIA		Entropy		Coverage		NumCluster		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.75	0.71	0.82	0.79	0.33	0.27	0.67	0.56	0.38	0.33	1	1	7	5	16.27	13.17	783	637
SUBCLU	0.68	0.51	0.77	0.67	0.02	0.01	0.54	0.04	0.27	0.24	1.0	0.82	357	5	2	1	5265	16
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	33	5187	4906
SCHISM	0.67	0.67	0.75	0.69	0.01	0.01	0.36	0.34	0.35	0.34	1.0	0.99	248	197	2.3	2.2	158749	114609
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	18	703	141
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
MINECLUS	0.78	0.69	0.78	0.76	0.19	0.18	1.0	1.0	0.56	0.37	1.0	1.0	64	32	33	33	40359	29437
KYMEROCCLUS	0.66	0.57	0.79	0.76	0.18	0.14	0.56	0.51	0.31	0.25	1.0	1.0	19	13	12.36	9.26	8	7
INSCY	0.74	0.55	0.77	0.76	0.02	0.0	0.24	0.11	0.60	0.39	0.97	0.74	2038	167	11	4.4	134373	63484
FIRES	0.49	0.03	0.76	0.76	0.03	0.0	0.05	0.0	1.0	0.01	0.76	0.04	11	1	2.5	1	250	31
DOC	0.73	0.61	0.81	0.76	0.11	0.04	0.84	0.07	0.46	0.27	1.0	0.8	60	6	27.2	2.8	1.00E + 06	37515
CLIQUE	0.67	0.67	0.71	0.71	0.02	0.02	0.40	0.40	0.26	0.26	1.0	1.0	107	107	1.7	1.7	453	453
CHAMELEOCLUS	0.60	0.51	0.76	0.76	0.23	0.11	0.53	0.25	0.25	0.22	1.0	1.0	8	4	16.75	5.75	339	131
Dataset Diabetes	F-measure		Accuracy		CE		RNIA		Entropy		Coverage		NumCluster		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.73	0.68	0.75	0.72	0.29	0.27	0.68	0.59	0.3	0.26	1	1	8	6	4	2.92	583	483
SUBCLU	0.74	0.45	0.71	0.68	0.01	0.01	0.01	0.01	0.14	0.11	1.0	1.0	1601	325	4.7	4	190122	58718
STATPC	0.73	0.59	0.70	0.65	0.06	0.0	0.63	0.17	0.72	0.28	0.97	0.75	363	27	8	8	27749	4657
SCHISM	0.7	0.62	0.73	0.68	0.08	0.01	0.36	0.09	0.34	0.2	1.0	0.79	270	21	4.2	3.9	35468	250
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	6	360	109
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	2	656	141
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	5.2	3578	62
KYMEROCCLUS	0.69	0.64	0.73	0.70	0.21	0.08	0.55	0.40	0.25	0.21	1.0	1.0	16	13	3.69	2.87	3	3
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
FIRES	0.52	0.03	0.65	0.64	0.12	0.0	0.27	0.0	0.68	0.0	0.81	0.03	17	1	2.5	1	4234	360
DOC	0.71	0.71	0.72	0.69	0.31	0.26	0.92	0.79	0.31	0.24	1.0	0.93	67	17	8	5.1	1.00E + 06	51640
CLIQUE	0.70	0.39	0.72	0.69	0.03	0.01	0.14	0.01	0.23	0.13	1.0	1.0	349	202	4.2	2.4	11953	203
CHAMELEOCLUS	0.70	0.62	0.73	0.70	0.17	0.09	0.66	0.47	0.28	0.23	1.0	1.0	29	19	5	2.75	598	438
Dataset Glass	F-Measure		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.61	0.53	0.68	0.59	0.3	0.28	0.69	0.62	0.53	0.52	1	1	11	8	4	3.17	1117	1077
SUBCLU	0.50	0.45	0.65	0.46	0.0	0.0	0.01	0.01	0.42	0.39	1.0	1.0	1648	831	4.9	4.3	14410	4250
STATPC	0.75	0.40	0.49	0.36	0.19	0.05	0.67	0.37	0.88	0.36	0.93	0.8	106	27	9	9	1265	390
SCHISM	0.46	0.39	0.63	0.47	0.11	0.04	0.33	0.20	0.44	0.38	1.0	0.79	158	30	3.9	2.1	313	31
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	2	375	250
P3C	0.28	0.23	0.47	0.39	0.14	0.13	0.3	0.27	0.43	0.38	0.89	0.81	3	2	3	3	32	31
MINECLUS	0.76	0.40	0.52	0.50	0.24	0.19	0.78	0.45	0.72	0.46	1.0	0.87	64	6	7	4.3	907	15
KYMEROCCLUS	0.65	0.51	0.71	0.60	0.32	0.24	0.85	0.76	0.65	0.55	1.0	1.0	23	19	6.74	5.7	18	16
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
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
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	3.3	23172	78
CLIQUE	0.51	0.31	0.67	0.50	0.02	0.0	0.06	0.0	0.39	0.24	1.0	1.0	6169	175	5.4	3.1	411195	1375
CHAMELEOCLUS	0.43	0.28	0.57	0.50	0.43	0.26	0.88	0.55	0.46	0.36	1.0	1.0	8	4	7.5	4.75	195	95
Dataset Liver	F-Measure		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.7	0.65	0.69	0.64	0.25	0.24	0.6	0.53	0.19	0.16	1	1	6	4	3	2.5	673	670
SUBCLU	0.68	0.68	0.64	0.58	0.11	0.02	0.68	0.05	0.07	0.02	1.0	1.0	334	64	3.4	1.3	1422	47
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	3.3	1890	781
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
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	3	78	31
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	3	172	32
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	3.7	49563	1954
KYMEROCCLUS	0.65	0.56	0.67	0.60	0.21	0.09	0.56	0.43	0.12	0.04	1.0	1.0	17	11	2.82	2	2	2
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
FIRES	0.58	0.04	0.58	0.56	0.14	0.0	0.39	0.01	0.37	0.0	0.84	0.03	10	1	3	1	531	46
DOC	0.67	0.64	0.68	0.58	0.11	0.07	0.51	0.35	0.18	0.11	0.99	0.9	45	13	3	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	1.0	1.0	1922	19	4.1	1.7	38281	15
CHAMELEOCLUS	0.65	0.59	0.68	0.62	0.20	0.10	0.53	0.41	0.14	0.07	1.0	1.0	27	22	2.48	1.85	202	158

time by 10 times. The comparison shows that the execution time of *SOSubClust* is in a suitable range as it ranks second in the list.

4.1.2.2. Number of clusters. For the real-life data set, information on the actual number of clusters is not available, rather information about the number of classes is available. Therefore, we say an algorithm generates appropriate number of clusters, if the absolute difference between the number of clusters produced by the algorithm to the number of classes is minimum. *SOSubClust* generates suitable number of clusters and it ranks *first* as shown in Fig. 5(c).

4.1.2.3. Coverage. The coverage denotes the fraction of the objects in the data set that are part of any cluster. This measure takes the value of 1 if all the objects are associated with one of the output clusters; otherwise, it will be less than 1. *SOSubClust* ranks 1st in this category along with ChameleoClust, KymeroClust and Clique as shown in Fig. 5(b).

4.2. Results on synthetic data sets

Like real-life data sets, in case of synthetic data sets also, we executed the experiments 10 times and in each execution, we

Table 4
Results for the data sets *Shape*, *Vowel*, *Pendigits* with respect to primary and secondary metrics.

Dataset Shape	F-Measure		Accuracy		CE		$\overline{RNIA}$		$\overline{Entropy}$		Coverage		NumCluster		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Proposed	0.76	0.72	0.82	0.77	0.36	0.33	0.63	0.56	0.77	0.68	1	1	15	13	7.67	6.25	1729	1650
SUBCLU	0.36	0.29	0.70	0.64	0.0	0.0	0.05	0.04	0.89	0.88	1.0	1.0	3468	3337	4.5	4.1	4063	1891
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	17	250	171
SCHISM	0.51	0.3	0.74	0.49	0.10	0.0	0.26	0.01	0.85	0.55	1.0	0.92	8835	90	6	3.9	712964	9031
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	7	593	469
P3C	0.51	0.51	0.61	0.61	0.14	0.14	0.17	0.17	0.8	0.8	0.66	0.66	9	9	4.1	4.1	140	140
MINECLUS	0.94	0.86	0.79	0.60	0.58	0.46	1.0	1.0	0.93	0.82	1.0	1.0	64	32	17	17	46703	3266
KYMEROCCLUS	0.82	0.72	0.86	0.79	0.57	0.53	0.86	0.80	0.83	0.77	1.0	1.0	19	16	13.5	12.56	101	91
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
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
DOC	0.90	0.83	0.79	0.54	0.56	0.38	0.90	0.82	0.93	0.86	1.0	1.0	52	29	13.8	12.8	2.00E + 06	86500
CLIQUE	0.31	0.31	0.76	0.76	0.01	0.01	0.07	0.07	0.66	0.66	1.0	1.0	486	486	3.3	3.3	235	235
CHAMELEOCLUS	0.75	0.63	0.80	0.71	0.54	0.49	0.78	0.71	0.77	0.67	1.0	1.0	14	10	12.4	10.79	462	252
Dataset Vowel	F-Measure		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
Proposed	0.38	0.33	0.48	0.43	0.25	0.16	0.66	0.57	0.33	0.31	1	1	15	11	5	4.5	5169	5097
SUBCLU	0.24	0.18	0.58	0.38	0.04	0.01	0.39	0.04	0.30	0.13	1.0	1.0	10881	709	3.6	2	26047	2250
STATPC	0.22	0.22	0.56	0.56	0.06	0.06	0.12	0.12	0.14	0.14	1.0	1.0	39	39	10	10	18485	16671
SCHISM	0.27	0.23	0.62	0.52	0.05	0.01	0.43	0.11	0.29	0.21	1.0	0.93	494	121	4.3	2.8	23031	391
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	8	766	766
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	4.7	1610	625
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
KYMEROCCLUS	0.53	0.48	0.53	0.47	0.16	0.14	0.75	0.70	0.56	0.52	1.0	1.0	50	45	6.82	6.3	364	339
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
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
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	10	120015	120015
CLIQUE	0.23	0.17	0.64	0.37	0.05	0.0	0.44	0.01	0.10	0.09	1.0	1.0	3062	267	4.9	1.9	523233	1953
CHAMELEOCLUS	0.41	0.37	0.42	0.38	0.17	0.13	0.65	0.54	0.45	0.40	1.0	1.0	33	24	6	4.57	995	787
Dataset Pendigits	F-Measure		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
Proposed	0.6	0.54	0.73	0.68	0.33	0.29	0.68	0.61	0.59	0.54	1	1	23	18	6.7	5056	5819	5736
STATPC	0.91	0.32	0.92	0.10	0.09	0.0	0.67	0.11	1.0	0.53	0.99	0.84	4109	56	16	16	5.00E + 06	3.00E + 06
SUBCLU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCHISM	0.45	0.26	0.93	0.71	0.05	0.01	0.30	0.08	0.5	0.45	1.0	0.93	1092	290	10.1	3.4	5.00E + 08	21266
PROCLUS	0.78	0.73	0.74	0.73	0.31	0.27	0.64	0.45	0.90	0.71	0.9	0.74	37	17	14	8	6045	4250
P3C	0.74	0.74	0.72	0.72	0.28	0.28	0.58	0.58	0.76	0.76	0.9	0.9	31	31	9	9	2.00E + 06	2.00E + 06
MINECLUS	0.87	0.87	0.86	0.86	0.48	0.48	0.89	0.89	0.82	0.82	1.0	1.0	64	64	12.1	12.1	780167	692651
KYMEROCCLUS	0.92	0.89	0.92	0.89	0.40	0.34	0.80	0.78	0.89	0.86	1.0	1.0	41	34	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
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	2.00E + 06	1.00E + 06
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	1.0	1.0	1890	36	3.1	1.5	67891	219
CHAMELEOCLUS	0.71	0.51	0.74	0.59	0.51	0.30	0.78	0.49	0.68	0.58	1.0	1.0	14	10	12.4	7.21	4476	4226

select the solution with the best results. The *maximum* and *minimum* scores with respect to different evaluation metrics attained by the proposed algorithm, *SOSubClust*, are reported in Table 5. The parameter *Iter* for synthetic data sets is taken as 4000. Again, for each evaluation metric, we select the best scores produced for all the 16 synthetic data sets. Then for each evaluation measure, we plot the best measure values obtained versus the best number of clusters generated, for 16 synthetic data sets [shown in Fig. 6]. Again, with respect to the generation of *number of clusters*, it is observed that for all the synthetic data sets, the number of clusters found is very close to the actual number of clusters (i.e., 10). The number of clusters generated by the proposed algorithm, *SOSubClust*, ranges from 10 to 14. Among the state-of-the-art algorithms, the number of clusters produced by P3C, ChameleoClust and KymereoClust, are in the range of 6 to 16, 9 to 16 and 9 to 14, respectively. The rest of the algorithms produce huge number of clusters in the range of 5 and 50 and in a few cases, many more as reported in (Müller et al., 2009). Therefore, considering all the synthetic data sets, the proposed algorithm provides best result with respect to the number of clusters generated.

4.3. Results on categorical and big data sets

In subspace clustering problem, in order to illustrate that the usage of multi-objective optimization provides better results than those obtained by single objective optimization, we have considered one categorical data set and three big data sets collected from UCI (Lichman, 2013). The big data sets selected are *HTRU2* (17898 instances and 8 dimensions), *MAGIC Gamma Telescope* (19020 instances and 10 dimensions) and *Crowdsourced Mapping* (10546 instances and 29 dimensions). In the proposed method, ICC-index and PSM-index are used as the optimization functions. In order to show the effectiveness of the use of multi-objective based search (MOO), these data sets are also evaluated using a subspace clustering approach (SOO) optimizing only a single objective function (ICC-index only). Both the algorithms, MOO (using multi-objective) and SOO (using single-objective), are executed 10 times. We have reported the best values obtained corresponding to different evaluation metrics as shown in Table 6. However, the categorical data set MNIST² is also considered for comparison

² <https://www.kaggle.com/oddrational/mnist-in-csv>.

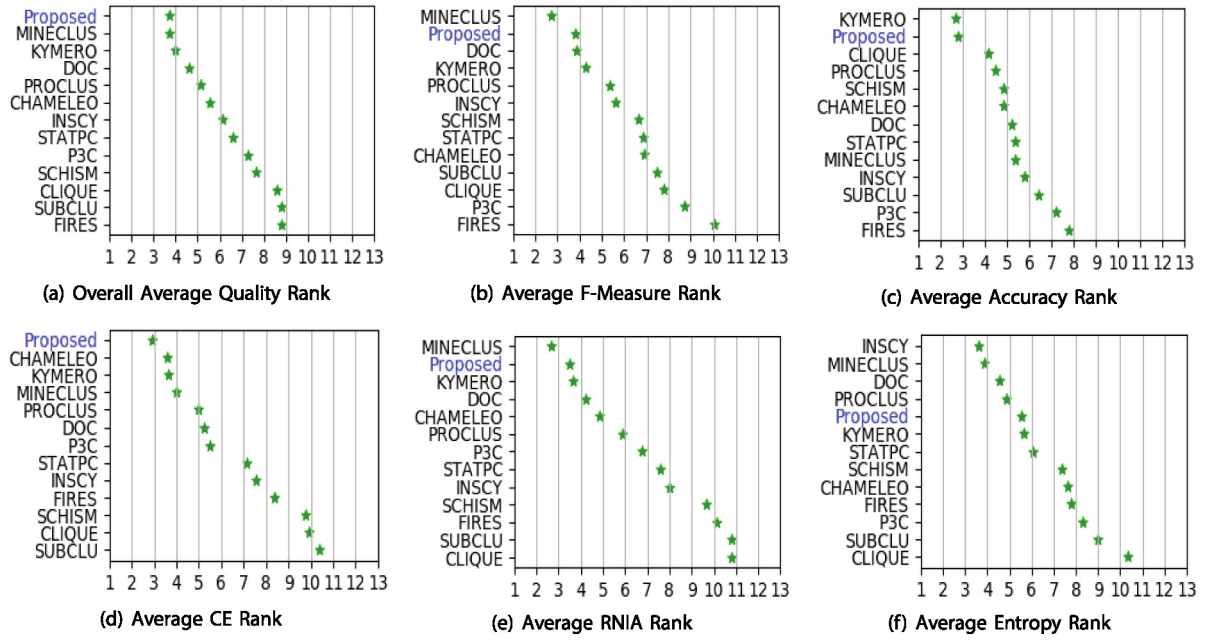


Fig. 4. Average rankings over all real-life data sets with respect to F1, Accuracy, CE, RNIA, Entropy and combining all the primary evaluation metrics together.

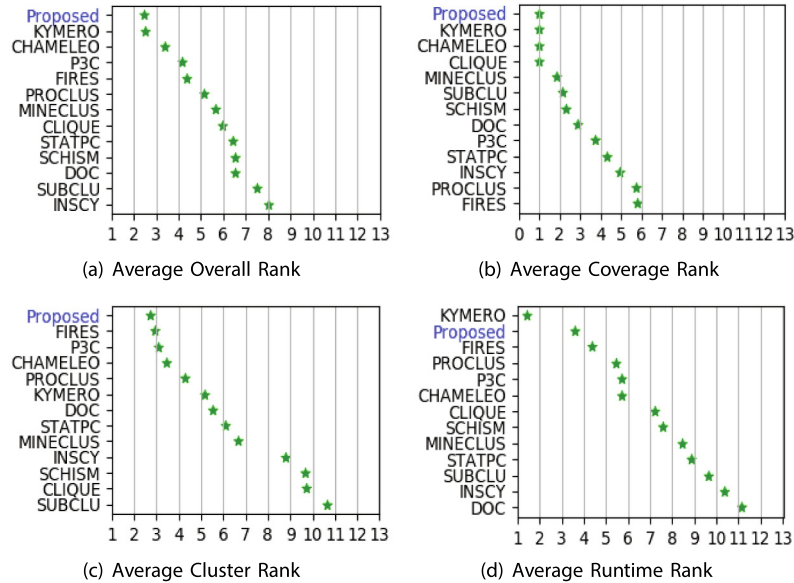


Fig. 5. Average rankings over all data sets with respect to combined secondary metrics, coverage, number of clusters and runtime.

between the multi-objective and single-objective versions of subspace clustering approach. Similarity between two categorical data points can be measured by checking similarity of their attribute values. From the results shown in Table 6, the efficacy of optimizing multiple objectives over a single objective is clearly visible. In almost all the cases, the proposed MOO based technique outperforms (values are shown in bold) the single objective version.

4.4. Subspace clustering vs. conventional clustering

In order to show the utility of subspace clustering, results of the proposed subspace clustering technique are also compared with those obtained by clustering algorithms applied on datasets with full feature set. To accommodate the full feature set, we have modified the proposed method. In the modified version (named as *pro-*

p_orig.), we have only optimized ICC-index as the only objective; PSM-index which is used to optimize the subspace feature is not required as we are considering the full feature set for clustering. Moreover, the mutation operators that have been defined in this approach are also modified accordingly. Furthermore, we have also considered the K-means algorithm with full feature set (named as *kmeans_orig.*) for the purpose of comparison. Finally, we have compared the proposed method, i.e., method with subspace feature set (named as *prop_sub.*) with the modified proposed method (*prop_orig.*), and K-means (*kmeans_orig.*) utilizing the full-feature set. For the purpose of comparison, we have considered a few real-life data sets *breast*, *liver*, *diabetes* and *glass*, described in Table 1, and F-measure, Accuracy and Entropy as evaluation metrics. The obtained results are shown in Table 7. It is observed that, the proposed method with subspace feature set performs well (values are

Table 5
Results for the 16 synthetic data sets with respect to both primary and secondary metrics.

Dataset	F-Measure		Accuracy		CE		RNIA		Entropy		NumClusters		AvgDim		RunTime	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
D05	0.61	0.44	0.68	0.55	0.54	0.29	0.81	0.46	0.73	0.59	13	10	1.91	1.67	1963	1807
D10	0.75	0.64	0.74	0.62	0.4	0.28	0.57	0.47	0.8	0.67	13	10	5	4	2789	2515
D15	0.67	0.43	0.75	0.67	0.39	0.28	0.53	0.38	0.79	0.69	14	10	7	4.67	2332	2122
D20	0.78	0.66	0.75	0.72	0.36	0.33	0.5	0.44	0.81	0.78	17	14	7.94	7.35	1929	1883
D25	0.77	0.63	0.78	0.74	0.4	0.35	0.58	0.53	0.82	0.79	17	13	10.29	9.4	2012	1917
D50	0.76	0.67	0.79	0.6	0.41	0.34	0.57	0.42	0.82	0.6	19	14	22.08	21.88	4234	4101
D75	0.62	0.52	0.67	0.59	0.35	0.33	0.57	0.52	0.74	0.66	17	14	33.94	33.13	2767	2613
S1500	0.79	0.68	0.78	0.74	0.35	0.3	0.47	0.45	0.82	0.78	18	12	9.06	7.28	3914	3792
S2500	0.68	0.55	0.74	0.64	0.39	0.35	0.52	0.47	0.8	0.72	14	11	9.55	8.5	3910	3731
S3500	0.67	0.61	0.77	0.71	0.4	0.36	0.5	0.47	0.79	0.72	15	13	8.31	7.4	3766	3578
S4500	0.83	0.5	0.81	0.58	0.38	0.26	0.51	0.43	0.86	0.6	13	11	10	8.62	3895	3720
S5500	0.78	0.7	0.81	0.78	0.38	0.31	0.52	0.43	0.83	0.74	15	13	9.92	8.6	3873	3717
N10	0.82	0.62	0.84	0.71	0.41	0.32	0.51	0.44	0.86	0.72	17	13	9.43	7.39	3567	3452
N30	0.71	0.6	0.74	0.64	0.25	0.22	0.4	0.34	0.73	0.65	16	12	10	9.71	3988	3878
N50	0.76	0.66	0.8	0.73	0.3	0.24	0.37	0.33	0.83	0.69	15	13	10	9.73	3918	3791
N70	0.77	0.59	0.78	0.64	0.19	0.13	0.27	0.23	0.8	0.67	18	14	10	8.36	3811	2309

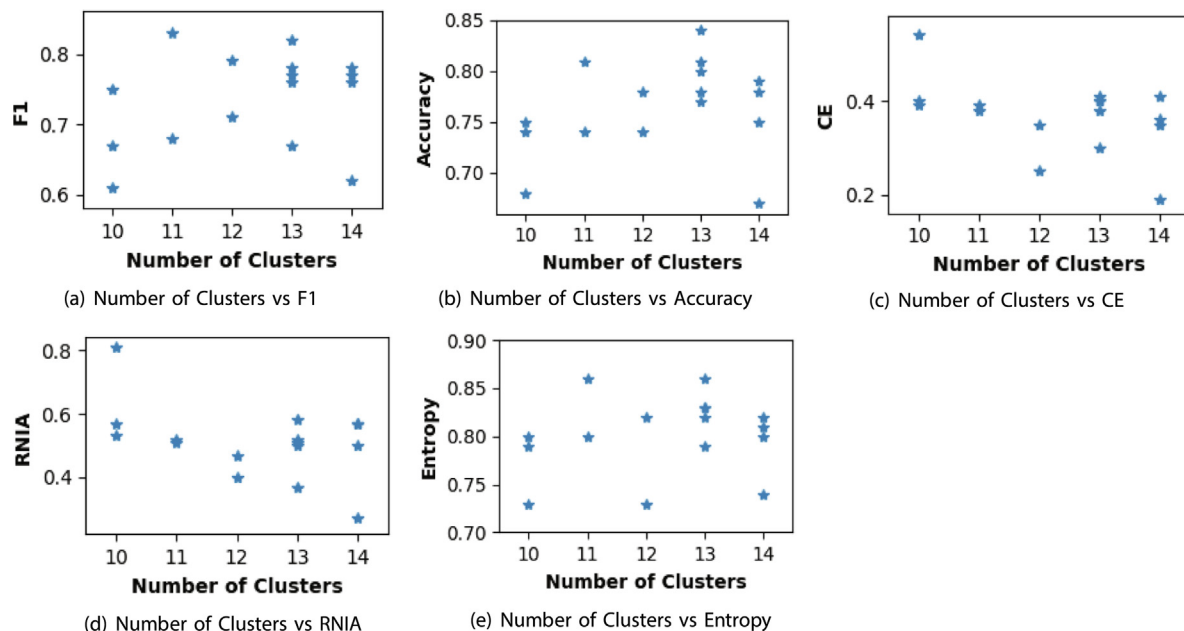


Fig. 6. Primary evaluation metrics vs average number of clusters for the best results among 10 runs for the 16 synthetic datasets.

shown in bold). This indicates that the quality of clusters will be improved if the clusters are represented only by the relevant sub-space feature sets.

The proposed method is also tested with some more real-life data sets such as *iris*³ (150 instances, 4 dimensions), *ecoli*⁴ (336 instances, 8 dimensions) and *glass* and the obtained results are compared with those by some state-of-the-art clustering approaches such as DSKmeans (Huang et al., 2014), ESSC (Deng, Choi, Chung, & Wang, 2010) and AWA (Chan, Ching, Ng, & Huang, 2004). The proposed method is compared with these previous methods with respect to the evaluation metrics, *F-measure* and *Accuracy* as shown in Table 8. The comparison shown in Table 8, clearly illustrates the superiority of the proposed algorithm (SOSubClust).

4.5. Real-life application of the proposed method

The analysis of gene expression data in micro-array technology is useful in solving problems like gene expression profiling, medical diagnosis etc. These problems require the proper identification of similar genes, i.e., genes having similar expression patterns and automatically grouping them together. Many traditional clustering techniques have been applied in grouping these genes and it is found that the conventional clustering method is not suitable when the genes are represented by large number of features. However, the behaviours of these genes are such that, some genes are relevant to a particular subset of features where as some other genes are relevant to some other subset of features. Therefore, sub-space clustering technique can be useful in dealing with such situation where different groups of clusters can be represented by different subspace feature sets unlike traditional clustering methods where each cluster is represented by same lower dimensional space.

³ <http://archive.ics.uci.edu/ml>.

⁴ <http://archive.ics.uci.edu/ml>.

Table 6

Results on one categorical and three big data sets in terms of primary evaluation metrics.

Dataset	F-Measure		Accuracy		CE		RNIA		Entropy	
	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO	SOO	MOO
Crowd Source	0.70	0.63	0.69	0.72	0.17	0.18	0.49	0.54	0.19	0.16
HTRU	0.87	0.89	0.96	0.97	0.28	0.45	0.62	0.74	0.84	0.82
Magic	0.32	0.36	0.76	0.84	0.13	0.32	0.40	0.77	0.58	0.71
MNIST	0.61	0.68	0.73	0.85	-	-	-	-	0.65	0.73

Table 7Comparison of the proposed subspace clustering method (*prop_sub.*) with respect to K-means (*kmeans_orig.*) and the modified proposed method (*prop_orig.*) which are applied on the original feature sets.

Data set	F-Measure			Accuracy			Entropy		
	<i>prop_sub.</i>	<i>prop_orig.</i>	<i>kmeans_orig.</i>	<i>prop_sub.</i>	<i>prop_orig.</i>	<i>kmeans_orig.</i>	<i>prop_sub.</i>	<i>prop_orig.</i>	<i>kmeans_orig.</i>
liver	0.70	0.65	0.56	0.69	0.65	0.58	0.19	0.09	0.04
breast	0.75	0.51	0.54	0.82	0.77	0.74	0.38	0.29	0.22
diabetes	0.73	0.69	0.64	0.75	0.75	0.70	0.30	0.24	0.21
glass	0.61	0.49	0.46	0.68	0.60	0.55	0.53	0.48	0.46

Table 8Comparison of the proposed method with some previous subspace clustering methods considering the data sets, *iris*, *glass* and *ecoli*.

Data Set	F-Measure				Accuracy			
	Proposed	DSKmeans	AWA	ESSC	Proposed	DSKmeans	AWA	ESSC
glass	0.61	0.52	0.46	0.48	0.68	0.47	0.43	0.43
iris	0.95	0.92	0.91	0.86	0.95	0.91	0.89	0.84
ecoli	0.59	0.66	0.56	0.56	0.82	0.60	0.46	0.46

Table 9

Bi-clustering results obtained by the proposed method and other state-of-the-art techniques on Human Large B Cell Lymphoma data set.

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best(Max)	Average	Best(min)
Proposed	348.83	133.12	21685.97	23232.02	0.196	0.145
SGAB	855.36	572.54	2222.19	5301.83	0.5208	0.3564
MOPSOB	927.47	745.25	4348.2	8745.65	0.213	0.09
MOGAB	801.37	569.23	2378.25	5377.51	0.4710	0.2283
OPSM	1249.73	43.07	6374.64	11854.63	0.2520	0.1024
RWB	1185.69	992.76	1698.99	3575.40	0.7227	0.3386
BiVisu	1680.23	1553.43	1913.24	2468.63	0.8814	0.6537
ISA	2006.83	245.28	4780.65	14682.47	0.6300	0.0853
Bimax	387.71	96.98	670.83	3204.35	0.7120	0.1402

To show the applicability of the developed method, we have considered two gene-expression data sets such as *Human Large B Cell Lymphoma* (Cheng & Church, 2000) and *Yeast* (Cheng & Church, 2000) and tested with the proposed method. The generated subspace clusters are computed using the evaluation metrics, Mean Squared Residue (MSR) (Cheng & Church, 2000), BI-index (Maulik, Mukhopadhyay, & Bandyopadhyay, 2009) and Row Variance (RV) (Cheng & Church, 2000). The results obtained by the proposed method are compared with those obtained by several state-of-the-art methods such as MOPSOB (Liu, Li, Liu, & Chen, 2008), BiVisu (Cheng, Law, Siu, & Lau, 2007), MOGAB (Maulik et al., 2009), RWB (Angiulli & Pizzuti, 2005), OPSM (Ben-Dor, Chor, Karp, & Yakhini, 2003), SGAB (Tanay, Sharan, & Shamir, 2002), Bimax (Hartigan, 1972) and ISA (Ihmels, Bergmann, & Barkai, 2004). The comparisons are shown in Table 9 (human data set) & Table 10 (yeast data set) and it can be seen that the proposed

approach performs well in majority of the cases (values are shown in bold).

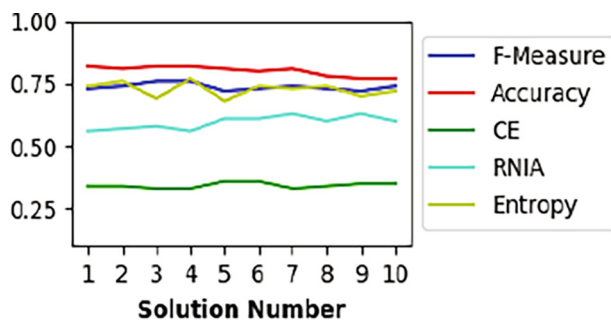
4.6. Advantages of multi-objective optimization

The proposed algorithm utilizes the advantages of optimizing multiple objectives; at any point in time, the algorithm produces multiple solutions. For simplicity, we have considered only the data set *shape* and plotted the results present in the final population with respect to all the evaluation metrics. In a multi-objective framework, different solutions encode different values of evaluation measures. The results attained by different solutions are plotted [Shown in Fig. 7]. It is clearly visible from the figure that, the use of MOO indeed produces a large number of alternative solutions. Depending on the need, the user can select a single one.

Table 10

Bi-clustering results of the proposed method and other state-of-the-art techniques on Yeast data set.

Algorithm	MSR		RV		BI-Index	
	Average	Best(min)	Average	Best(Max)	Average	Best(min)
Proposed	450.58	10.91	1643.23	4971.32	0.2039	0.029
SGAB	198.88	138.75	638.23	3605.93	0.4026	0.2714
MOPSOB	218.54	200.58	789.85	1254.56	0.28	0.16
MOGAB	185.85	116.34	932.04	3823.46	0.3329	0.2123
OPSM	320.39	118.53	1083.24	3804.56	0.3962	0.0012
RWB	295.81	231.28	528.97	1044.37	0.5869	0.2788
BiVisu	290.59	240.96	390.73	775.41	0.7770	0.3940
ISA	281.59	125.76	409.29	1252.34	0.7812	0.1235
Bimax	32.16	5.73	39.53	80.42	0.4600	0.2104

**Fig. 7.** Evaluation metric score on *shape* data set attained by the different solutions of the final population.

5. Conclusions and future works

In the proposed work we have developed a subspace clustering approach named *SOSubClust* which has the following unique properties:

- The proposed algorithm is developed in a multi-objective based framework using an evolutionary optimization technique.
- At each iteration, the qualities of the subspace clusters are improved continuously by using the objective functions, ICC-index and PSM-index.
- PSM-index deals with the selection of subspace features for representation of subspace clusters, as the selection of optimum subspace feature, is an important criterion to generate the good quality subspace clusters. This index is introduced in this paper.
- A new mutation operator, *duplication and deletion* and the modified version of *exogenous genetic material uptake* help the proposed approach to adapt to the new environment quickly.
- The efficacy of the proposed algorithm is shown for generating subspace clustering from seven real-world data sets, sixteen synthetic data sets, three real-life big data sets and many other data sets.
- The qualities of the obtained subspace clusters are compared with those obtained by the state-of-the-art algorithms and observed that the proposed algorithm can take advantages of evolutionary-based techniques and multi-objective framework.

In the algorithm, *SOSubClust*, each object is assigned to only one cluster based on minimum Manhattan distance. However, in the case of subspace clustering, as the subspace features may overlap, so an object can also be assigned to more than one cluster. Therefore, in future, we plan to develop a subspace clustering algorithm, where an object can belong to more than one cluster.

CRedit authorship contribution statement

All the authors conceived the idea of this research article. Dipanjyoti Paul conducted the experiment(s). All authors analysed the results. Dipanjyoti Paul wrote the manuscript with valuable inputs from Sriparna Saha and Jimson Mathew. All authors reviewed the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Dr. Sriparna Saha gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.

References

- Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMOD Record* (vol. 28, pp. 61–72). ACM.
- Aggarwal, C. C., & Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. *ACM*, 29.
- Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *ACM*, 27.
- Angiulli, F., & Pizzuti, C. (2005). Gene expression biclustering using random walk strategies. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 509–519). Springer.
- Assent, I., Krieger, R., Müller, E., & Seidl, T. (2008). Inscy: Indexing subspace clusters with in-process-removal of redundancy. *2008 Eighth IEEE International Conference on Data Mining* (pp. 719–724). IEEE.
- Bandyopadhyay, S., & Saha, S. (2012). A generalized automatic clustering algorithm in a multiobjective framework.
- Ben-Dor, A., Chor, B., Karp, R., & Yakhini, Z. (2003). Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology*, 10, 373–384.
- Bouveyron, C., & Brunet, C. (2012). Simultaneous model-based clustering and visualization in the fisher discriminative subspace. *Statistics and Computing*, 22, 301–324.
- Chan, E. Y., Ching, W. K., Ng, M. K., & Huang, J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, 37, 943–952.
- Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45, 434–446.
- Cheng, C.-H., Fu, A. W., & Zhang, Y. (1999). Entropy-based subspace clustering for mining numerical data. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 84–93). ACM.
- Cheng, K.-O., Law, N.-F., Siu, W.-C., & Lau, T. (2007). Bivisu: software tool for bicluster detection and visualization. *Bioinformatics*, 23, 2342–2344.
- Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. *Ismb* (vol. 8, pp. 93–103).
- Conant, G. C., & Wolfe, K. H. (2008). Turning a hobby into a job: How duplicated genes find new functions. *Nature Reviews Genetics*, 9, 938.

- Crombach, A., & Hogeweg, P. (2007). Chromosome rearrangements and the evolution of genome structuring and adaptability. *Molecular Biology and Evolution*, 24, 1130–1139.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
- Deng, Z., Choi, K.-S., Chung, F.-L., & Wang, S. (2010). Enhanced soft subspace clustering integrating within-cluster and between-cluster information. *Pattern Recognition*, 43, 767–781.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* (vol. 96, pp. 226–231).
- Fortin, F.-A., & Parizeau, M. (2013). Revisiting the nsga-ii crowding-distance computation. *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (pp. 623–630). ACM.
- Gan, G., & Ng, M. K.-P. (2015). Subspace clustering using affinity propagation. *Pattern Recognition*, 48, 1455–1464.
- Gan, G., & Ng, M. K.-P. (2015). Subspace clustering with automatic feature grouping. *Pattern Recognition*, 48, 3703–3713.
- Goil, S., Nagesh, H., & Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 443–452). ACM.
- Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67, 123–129.
- Hershey, J. R., Chen, Z., Le Roux, J., & Watanabe, S. (2016). Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 31–35). IEEE.
- Huang, X., Ye, Y., Guo, H., Cai, Y., Zhang, H., & Li, Y. (2014). Dskmeans: a new kmeans-type approach to discriminative subspace clustering. *Knowledge-Based Systems*, 70, 293–300.
- Hund, M., Färber, I., Behrisch, M., Tatu, A., Schreck, T., Keim, D. A., & Seidl, T. (2016). Visual quality assessment of subspace clusterings. *Workshop on Interactive Data Exploration and Analytics (IDEA'16)*.
- Ihmels, J., Bergmann, S., & Barkai, N. (2004). Defining transcription modules using large-scale gene expression data. *Bioinformatics*, 20, 1993–2003.
- Kailing, K., Kriegel, H.-P., & Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 2004 SIAM International Conference on Data Mining* (pp. 246–256). SIAM.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., & Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding dna. *Molecular Biology and Evolution*, 24, 2344–2353.
- Kriegel, H.-P., Kroger, P., Renz, M., & Wurst, S. (2005). A generic framework for efficient subspace clustering of high-dimensional data. *Fifth IEEE international conference on data mining (ICDM'05)* (pp. 8). IEEE.
- Lichman, M. (2013). UCI machine learning repository. URL: <http://archive.ics.uci.edu/ml>.
- Liu, J., Li, Z., Liu, F., & Chen, Y. (2008). Multi-objective particle swarm optimization biclustering of microarray data. *Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE International Conference on* (pp. 363–366). IEEE.
- Liu, Y., Jiao, L., & Shang, F. (2013). An efficient matrix factorization based low-rank representation for subspace clustering. *Pattern Recognition*, 46, 284–292.
- Maulik, U., Mukhopadhyay, A., & Bandyopadhyay, S. (2009). Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm. *IEEE Transactions on Information Technology in Biomedicine*, 13, 969–975.
- Müller, E., Günnemann, S., Assent, I., & Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2, 1270–1281.
- Ohno, S. (1999). Gene duplication and the uniqueness of vertebrate genomes circa 1970–1999. *Seminars in cell & developmental biology* (vol. 10). Elsevier.
- Ohno, S. (2013). *Evolution by gene duplication*. Springer science & business media.
- Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *Acm Sigkdd Explorations Newsletter*, 6, 90–105.
- Passalis, N., & Tefas, A. (2019). Discriminative clustering using regularized subspace learning. *Pattern Recognition*, 96 106982.
- Patrikainen, A., & Meila, M. (2006). Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering*, 18, 902–916.
- Peignier, S. (2017). *Subspace clustering on static datasets and dynamic data streams using bio-inspired algorithms* phd thesis. france: INSA, Lyon.
- Peignier, S., Rigotti, C., & Beslon, G. (2015). Subspace clustering using evolvable genome structure. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 575–582). ACM.
- Procopiu, C. M., Jones, M., Agarwal, P. K., & Murali, T. (2002). A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (pp. 418–427).
- Saha, S., & Bandyopadhyay, S. (2010). A symmetry based multiobjective clustering technique for automatic evolution of clusters. *Pattern Recognition*, 43, 738–751.
- Saha, S., & Bandyopadhyay, S. (2013). A generalized automatic clustering algorithm in a multiobjective framework. *Applied Soft Computing*, 13, 89–108.
- Sembarino, R.W., Zain, J.M., & Embong, A. (2010). Clustering high dimensional data using subspace and projected clustering algorithms. *arXiv preprint arXiv:1009.0384*.
- Sequeira, K., & Zaki, M. (2005). Schism: a new approach to interesting subspace mining. *International Journal of Business Intelligence and Data Mining*, 1, 137–160.
- Tanay, A., Sharan, R., & Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18, S136–S144.
- Tian, K., Zhou, S., & Guan, J. (2017). Deepcluster: a general clustering framework based on deep learning. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 809–825). Springer.
- Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478–487).
- Yamaguchi, M., Irie, G., Kawanishi, T., & Kashino, K. (2019). Subspace structure-aware spectral clustering for robust subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 9875–9884).
- Yiu, M. L., & Mamoulis, N. (2003). Frequent-pattern based iterative projected clustering. *Third IEEE International Conference on Data Mining* (pp. 689–692). IEEE.
- Zhu, P., Zhu, W., Hu, Q., Zhang, C., & Zuo, W. (2017). Subspace clustering guided unsupervised feature selection. *Pattern Recognition*, 66, 364–374.
- Zhu, Y., Ting, K. M., & Carman, M. J. (2018). Grouping points by shared subspaces for effective subspace clustering. *Pattern Recognition*.
- Zografos, V., Ellis, L., & Mester, R. (2013). Discriminative subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2107–2114).