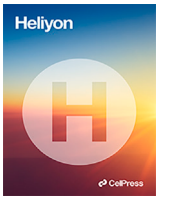
Heliyon 10 (2024) e34736

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A parameter free relative density based biclustering method for identifying non-linear feature relations

Namita Jain [a,*], Susmita Ghosh [a], Ashish Ghosh [b]

[a] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India
[b] International Institute of Information Technology, Bhubaneswar 751003, India

ARTICLE INFO

Dataset link: https://github.com/namitaML/RelDenClu-Non-linear-feature-relation-based-biclustering

ABSTRACT

The existing biclustering algorithms often depend on assumptions like monotonicity or linearity of feature relations for finding biclusters. Though a few algorithms overcome this problem using density-based methods, they tend to miss out many biclusters because they use global criteria for identifying dense regions. The proposed method, PF-RelDenBi, uses local variations in marginal and joint densities for each pair of features to find the subset of observations, forming the basis of the relation between them. It then finds the set of features connected by a common set of observations using a non-linear feature relation index, resulting in a bicluster. This approach allows us to find biclusters based on feature relations, even if the relations are non-linear or non-monotonous. Additionally, the proposed method does not require the user to provide any parameters, allowing its application to datasets from different domains.

To study the behaviour of PF-RelDenBi on datasets with different properties, experiments were carried out on sixteen simulated datasets and the performance has been compared with eleven state-of-the-art algorithms. The proposed method is seen to produce better results for most of the simulated datasets. Experiments were conducted with five benchmark datasets and biclusters were detected using PF-RelDenBi. For the first two datasets, the detected biclusters were used to generate additional features that improved classification performance. For the other three datasets, the performance of PF-RelDenBi was compared with the eleven state-of-the-art methods in terms of accuracy, NMI and ARI. The proposed method is seen to detect biclusters with greater accuracy. The proposed technique has also been applied to the COVID-19 dataset to identify some demographic features that are likely to affect the spread of COVID-19.

1. Introduction

Biclustering finds hidden patterns in the data by searching for paired subsets of features and observations. Each of these subsets is based on spatial proximity in space defined by selected features or similarity between features for selected observations. Most of the existing algorithms generally focus on the former, i.e., finding biclusters based on closeness in the selected subspace [1]. On the other hand, relation-based biclustering methods mostly search for biclusters based on linear relations between the features [2]. A few algorithms [3,4] exist for finding non-linear relation-based biclusters, where they search for biclusters with some constraints. For example, UniBic finds monotonous relations and CBSC [5] does not adjust for variations in marginal distributions, limiting their

* Corresponding author.
E-mail address: namita.saket@gmail.com (N. Jain).

https://doi.org/10.1016/j.heliyon.2024.e34736
Received 16 September 2023; Received in revised form 9 July 2024; Accepted 16 July 2024

Available online 20 July 2024

ability to find various types of relation-based biclusters. As seen, UniBic will not be able to find a relation based on the sine wave. CBSC finds more general relations, but if marginal densities have sparse and dense regions, the actual biclusters in the data may not be obtained due to fragmentation. In this article, we have tried to overcome these problems.

The proposed algorithm finds biclusters based on non-linear continuous feature relations. By scrutinizing joint and marginal densities for each feature pair, we discern observations forming relationships, subsequently expanding these associations to identify dense regions of related features. Leveraging distances between features based on selected observations, the proposed method PF-RelDenBi, efficiently clusters features, resulting in the identification of biclusters possessing desirable properties such as invariance to permutations, scaling, and translation. Notably, this invariance ensures the consistency of the proposed algorithm, regardless of changes in data representation. Additionally, it has been crafted to be entirely parameter-free, alleviating the user's burden by requiring the user to only provide the dataset for analysis. Since the thresholds used in the algorithm are calculated automatically based on the dataset itself, the proposed method is not dataset-dependent. This makes the proposed method versatile enough to be applied to datasets across different domains.

Beyond its parameter-free nature, PF-RelDenBi distinguishes itself by incorporating a non-linear feature dependence index named MIDI (Mutual Information based Dependence Index [6]) for clustering features that share common dense regions. This combination of parameter-free operation and the use of MIDI not only enhances the efficiency of the method but also underlines its adaptability and robust performance in diverse datasets. PF-RelDenBi, therefore, stands as a pioneering advancement in biclustering methodologies, providing a powerful, accessible, and efficient tool for uncovering meaningful patterns in complex datasets.

The rest of the article is organized as follows. We have discussed some of the existing biclustering methods in Section 2. Section 3 discusses the objective of the proposed method. Section 4 presents the proposed method and the algorithm. PF-RelDenBi is applied to simulated datasets with different properties and their results are presented in Section 5. The results on real-world datasets are reported in Section 6. We conclude with Section 7.

2. Related work and contributions

As mentioned, the present article aims to develop a non-linear feature relation-based biclustering technique using joint and marginal densities of features. In this section, we are going to discuss the related state-of-the-art techniques.

2.1. Discussion on density-based and relationship-based biclustering methods

Hartigan [7] pioneered a biclustering method based on a constant value of elements. Subsequently, many biclustering algorithms [8] were proposed with varying objectives and approaches. These include linear algebra-based ones like non-negative matrix factorization [9] and multibody factorization. Some algorithms find biclusters where the values of elements in a column are constant. This approach has been utilized for feature ranking and selection for classification [10]. BM-FM [11] finds biclusters with constant columns and utilizes them to forecast the trading points in the stock market. The method BIC-K-NN [12] also uses biclustering to study patterns in the stock market. Methods like spectral clustering by Luxburg [13] use linear algebra to find biclusters in transformed space. Several methods make use of evolutionary computing for biclustering. BP-EBA [14] finds biclusters using evolutionary computing in two phases. MVMC [15] is a method for multiview clustering where each bicluster uses a distinct dissimilarity matrix. Evolutionary computing based biclustering methods have been applied to genome analysis [16]. They have also been used to identify genes related to Cancer [17]. Other approaches used for biclustering are iterative methods [18] and graph-based methods [19]. Deep learning-based methods [20], aim to find a subspace in which observations lie close together. Detailed surveys on biclustering and subspace clustering have been done by Kriegel et al. [21], Prelić et al. [22], Parsons et al. [23], Madeira and Oliveira [8] and Vidal [1]. Density-based biclustering algorithms such as SubClu [24] and CLIQUE [25] find high-density regions in corresponding subspaces. These methods start by finding dense regions in one-dimensional space and grow biclusters using the apriori approach. Mean-shift methods [26] perform kernel density estimation, and iteratively locate the local maxima of the kernel mixture. The method CURLER [27] finds non-linear clusters by combining a density-based approach with the principal component method. CURLER finds high-density Gaussian regions using Expectation Maximization and combines them using the relation between directional information of resultant biclusters. The density-based methods mentioned above, find biclusters based on the proximity of observations in a subspace.

Existing methods for finding biclusters based on feature relations include the algorithm proposed by Cheng and Church [28] and the multiplicative algorithm FABIA [29]. Here the values of a feature can be obtained by multiplying the values of another feature by a constant. These methods assume that the relation between features is linear or multiplicative.

More generalized biclustering methods based on an arbitrary relationship between features are available in existing literature. These include algorithms like UniBic [4], based on the longest common subsequence. UniBic can find biclusters based on monotonous relationships between features. OPSM [3] finds order-preserving submatrices.

A more recent algorithm named CBSC [5] identifies non-linear relationships between pairs of dimensions using a density-based approach. However, CBSC tends to produce fragmented biclusters when marginal densities are highly variable. Another method, ARBic [30] measures trend-preserving patterns between rows as the paths in directed acyclic graphs. It then combines them to find biclusters. ARBic has been compared to several algorithms like FABIA, OPSM, QUBIC [31] and QUBIC2 [32] and is seen to produce better results. However, like CBSC, the performance of ARBic suffers when non-linear relations occur in the presence of highly variable marginal densities.

An algorithm that overcomes the problem of fragmentation is a density-based algorithm named RelDenClu [33]. Although RelDenClu has been successfully applied to find associations between demographic vaccine data and COVID-19 infection rates [34], it is difficult to use RelDenClu for datasets from diverse domains because it requires the users to provide several parameter values. The appropriate values of the parameters may change from one dataset to another, making its application to diverse real-world datasets difficult [35].

Some biclustering methods have been developed to overcome this problem. These include a biclustering method developed for recommender systems [36]. Work has been done to structure the data in coupled hierarchies of observations and features [37]. For spectral clustering, several researchers have analyzed the multiplicities of eigenvalues close to zero [13]. Recently a parameter-free biclustering method based on k Nearest Neighbour distance has been developed [35]. The authors of this method, ROCCO, have shown that the method works well whenever k lies within a particular range. Another method named MESBC finds mutually exclusive biclusters and only requires the number of clusters in the dataset as an input [38].

The proposed algorithm, PF-RelDenBi, finds non-linear continuous relationship-based biclusters and does not make assumptions of monotonicity, linearity or fixed relationships between features. It overcomes the problem of fragmentation by adapting to local variations in density along each dimension. The proposed method does not require the user to provide any input apart from the dataset, greatly improving its usability. Being a parameter-free algorithm, it is not dataset-dependent and has wide applicability across domains. Further, the utilization of MIDI contributes to significantly reduced execution time compared to algorithms like RelDenClu which search for the feature sets iteratively.

2.2. Contributions

The proposed method, PF-RelDenBi finds biclusters based on continuous non-linear relationships using marginal and joint densities and, information derived from 2D spaces corresponding to each pair of features. The advantages of this approach are listed below.

1. The users do not need to provide any parameter values.
2. The algorithm does not assume that the relationship between features has a particular form like linear or multiplicative. Thus the proposed method finds biclusters with both linear and non-linear relations among features.
3. The algorithm does not assume that the relationship between features is monotonous. In other words, biclusters based on both monotonous and non-monotonous relations in the data can be found.
4. It avoids the fragmentation of biclusters by adapting for variations in marginal densities.

Briefly speaking, the main objective of the proposed method is to find relation-based biclusters. This includes relations which are linear, non-linear and non-monotonous. Finding the latter two is more challenging. The proposed method is able to find biclusters having feature relations even if they are non-linear or non-monotonous.

3. Definitions

In this section, we present the objective of PF-RelDenBi and provide relevant definitions. The biclusters obtained by PF-RelDenBi are submatrices in which features are related directly or indirectly. Let the dataset be denoted by a matrix D containing N rows each corresponding to observation and M columns each corresponding to a feature. In this article, a direct relationship between two columns of a matrix means the following: dependence exists between the features corresponding to these two columns when the subset of observations contained in the bicluster is used as the base. Here dependence is estimated using MIDI [6] and is further discussed in Section 4.3. Two columns are connected indirectly if there exists a chain of columns connecting them, such that each consecutive pair of columns in this chain is directly connected. This has been stated in the definitions given below.

Definition 1. For a matrix A let the i^{th} column be denoted by $A_{*,i}$. In this article, we call the i^{th} and the j^{th} columns of a matrix to be directly connected if there is a dependence between these two columns. Perfect dependence occurs when $A_{*,i}$ can be used to determine $A_{*,j}$, or vice-versa.

Let us assume, each direct relationship between features to be an edge connecting two vertices, each representing a column. Two columns are said to be indirectly connected if they are connected by a path. It is to be noted that, the direct or indirect connection between the columns exists only on the base of rows contained in the submatrix.

Definition 2. We call the i^{th} and the j^{th} columns of a matrix A to be connected if these two columns are directly connected or there exist columns $A_{*,r_1}, \cdots, A_{*,r_k}$, such that (1) Each column in this list is directly connected to the consecutive column in the list (e.g., A_{*,r_2} is directly connected to A_{*,r_3}), (2) $A_{*,i}$ is directly connected to A_{*,r_1}, and (3) A_{*,r_k} is directly connected to $A_{*,j}$.

Using the definition of connected features we now define a bicluster as follows:

Definition 3. A bicluster in the data matrix D is a pair of two sets given by $O = \{oi_1, oi_2, \cdots, oi_n\}$ and $F = \{fi_1, fi_2, \cdots, fi_m\}$. The matrix D, restricted to observations in O and features in F gives us a submatrix A corresponding to the bicluster $< O, F >$. More

specifically, the submatrix corresponding to $< O, F >$ is given by A where the q^{th} element of the p^{th} row is given by, $A[p,q] = D[oi_p, fi_q]$. A bicluster is said to be relation-based if all the columns of the corresponding submatrix are connected, either directly or indirectly.

It may be noted that the proposed method finds biclusters based on the relation between feature pairs and so it can result in biclusters with disconnected regions. Thus, the proposed method can be seen as grouping related features based on observations.

4. Proposed method: PF-RelDenBi

We now present the procedure for finding dense sets for each pair of features and then combine them to obtain biclusters.

4.1. Finding the set of observations having dependence for a given feature pair

This section describes a method for identifying subsets of observations which show dependence between two features, by comparing joint distribution and marginal distributions using histograms. After normalizing the data to the range $[0, 1]$, we take $3\log(N)$ equal intervals along each axis, where N is the number of observations in the dataset.

Since, the data lies in the range $[0, 1]$, taking $3\log(N)$ cells along each axis in two-dimensional space leads to cell area $A = 1/(3\log(N))^2$. This leads to consistent density estimates [39] as we know that $\lim_{N\to\infty} A = \lim_{N\to\infty} 1/(3\log(N))^2 = 0$ and the product of the area of the cell and number of observations given by $\lim_{N\to\infty} NA = \lim_{N\to\infty} N/(3\log(N))^2 = \infty$. We have taken $3\log(N)$ instead of $\log(N)$ so that there is a sufficient number of cells even when N is small. Using this scheme, the joint density and marginal density of each cell are estimated. Of course, any histogram scheme which leads to consistent density estimates can be used here. We however chose to use the simple scheme with $3\log(N)$ cells since it leads to smaller execution times.

Thus, each cell obtained using the above-mentioned scheme can be represented as a region $I_x \times I_y$, where I_x and I_y are intervals of length $1/(3\log(N))$, along X and Y axes, respectively. The regions $I_x \times [0, 1]$ and $[0, 1] \times I_y$ can be used to calculate the marginal densities for the cell $I_x \times I_y$. We call these regions the marginal cells for convenience. We then decide whether a cell is dense if its density is greater than the densities of marginal cells as well as the average density of the entire space. The thought behind this step is that higher cell density compared to marginal density implies, a higher conditional probability of a point lying in a particular cell within a given marginal cell.

Two dense regions are merged if they are adjacent to each other horizontally or vertically or diagonally. A dense region can have at most 8 dense regions connected to it. After finding dense regions in 2D space, noise is removed by using the overlap between three two-dimensional spaces for every set of three features. Thus, redundancy introduced by using three features is used to weed out the noise. Up to this point, the procedure for finding the dense cells used by RelDenClu and the proposed method are the same. But in the next step, RelDenClu needs user-defined parameters for pruning the noise while the proposed method calculates the thresholds required for pruning automatically. A detailed description of this step is given below.

4.2. Finding the pruned set of observations for biclusters

To identify the set of observations related to each other, we find relative density-based subsets in two-dimensional Euclidean space given by each pair of features. However, because the background noise can be distributed in arbitrary ways, this procedure of finding relations can result in a lot of noise. Therefore, to weed out noise, we search for a set of three features, for which a given set of observations forms dense regions for each pair of features belonging to the set. Thus for a set of features $\{f_1, f_2, f_3\}$, we find dense regions for pairs $< f_1, f_2 >$, $< f_2, f_3 >$, and $< f_3, f_1 >$. The intersection of observations forming dense regions for these three pairs is found. This observation set is retained only if it contains a significant number of observations. Let us call this significant number $ObsMin$. The value of $ObsMin$ is calculated using the dense regions calculated earlier as discussed in Section 4.1. The method of calculating $ObsMin$ is discussed in the following paragraphs.

In Section 4.1, we calculated dense regions for each feature pair. A dataset having M features will have $M(M-1)/2$ feature pairs. The number of points lying in dense regions, for each feature pair may vary between 0 and N (number of observations in the dataset). Thus $M(M-1)/2$ values (number of observations lying in dense regions for each feature pair) are distributed over the range $[0, N]$. This distribution can give us an idea about the size of biclusters in the dataset. This is because the distribution will tend to peak around the actual size of biclusters, as many dense regions of this size will exist. This distribution can be examined using histograms. As already discussed, the number or size of histogram bins plays an important role. Since $M(M-1)/2$ values are distributed over $[0, N]$, according to the discussion in Section 4.1, we can divide this range to $3log_2(M(M-1)/2)$ bins to obtain consistent density estimates.

Since many of the feature pairs will have no dense regions or might have a very small number of points in dense regions caused by noise, we will ignore the first bin of the histogram. We then find the first bin b_{lim} such that the next two consecutive bins are populated (i.e. one or more feature pairs have dense regions such that 'the number of observations in the regions' lies in the bin range). The lower edge of b_{lim} denoted by $lEdgeB$, can be taken as the lower limit of the minimum number of observations in any bicluster. However, we do not want the lower limit to be too large; so, we take it to be $\min(lEdgeB, N/2)$. We denote this value as $Obsmin = \min(lEdgeB, N/2)$. The reason for considering two consecutive bins is that we want to identify the place where the distribution starts to increase.

For each set of three features, we can now retain the observation sets containing greater than $ObsMin$ observations and discard the rest. Let us refer to the retained observation sets as the "Triplet observation sets" for convenience. Each of these sets is now represented as a binary array. The presence of observation is indicated by 1 and absence by 0. However many of the sets may be very similar to each other. To combine these similar sets we perform single-linkage clustering on the binary arrays representing observation sets. Cosine similarity is used for this clustering. Thus we obtain several clusters of sets of observations.

The hierarchical clustering mentioned above requires a parameter that indicates the maximum number of clusters to be found by the algorithm. Let us say that PF-RelDenBi sets this value automatically to $NclusObsMax$. We will now discuss how the value of $NclusObsMax$ is calculated.

The maximum number of clusters that any clustering algorithm can find should be greater than 2. Thus $NclusObsMax$ should be at least 3. Further, we note that if $ObsMin$ (minimum number of observations in "Triplet observation sets") is larger, we would have a greater variety of observation sets. This is because a larger value of $ObsMin$ leads to a greater number of possible combinations.

On the other hand, if the sum of the number of rows in different biclusters in a dataset is fixed, we tend to have a greater number of "Triplet observation sets" if the number of biclusters is less and vice versa. This can be understood as follows. Suppose a dataset has i biclusters with $m_1, m_2, \cdots, m_i$ rows each while another dataset has single bicluster having m rows where $m = \sum_{k=1}^{i} m_k$. Let us assume that all the rows are connected to each other. We know that $m^3 \geq \sum_{k=1}^{i} m_k^3$. So the dataset with a smaller number of biclusters will have a larger number of "Triplet observation sets".

Thus we want $NclusObsMax$ to increase with $ObsMin$ but decrease with the number of "Triplet observation sets" denoted by $NtripleSets$. So we set $NclusObsMax = ceil(3 + ObsMin/NtripleSets)$, where the function $ceil(x)$ gives the smallest integer greater than or equal to x.

Thus, using hierarchical clustering, we may obtain many (less than or equal to $NclusObsMax$) clusters of "Triplet observation sets". From each of these clusters, we obtain a single set of observations, by finding the observations that appear in many "Triplet observation sets" lying in the cluster. We say that observations lie in many "Triplet observation sets" if they lie in more than $1/NclusObsMax$ of all the "Triplet observation sets" forming the cluster. This is because, a smaller number of biclusters means a greater number of "Triplet observation sets" in each, thus even if a smaller proportion of the "Triplet observation sets" contain an observation, we still have sufficient redundancy. At this stage, only the set of observations having greater than $ObsMin$ elements should be retained and the rest should be ignored. Note that the same observation may be present in several clusters.

4.3. Finding feature sets for each bicluster

In the previous step, we found several sets of observations. We now need to find corresponding feature sets. For each set of observations, we perform single-linkage clustering and choose the largest feature cluster. The maximum number of clusters is taken to be $M/2$, where M is the number of features in the dataset. Note that this is an upper limit and a smaller number of feature clusters can be obtained, thus a high value for the upper limit is chosen.

The distance between features f_1 and f_2 is taken as $1 - MIDI(f_1, f_2)$, where MIDI is a feature dependence index [6]. $MIDI$ uses normalized mutual information as feature dependence and is capable of finding non-linear dependence.

4.3.1. Feature dependence index MIDI used for finding feature sets

Since the Mutual Information based Dependence Index (MIDI) is used to find related features in the proposed method, we present the mathematical notion of feature dependence as defined for the index MIDI.

For two discrete random variables X, Y, the probability mass functions are denoted by $P_X(x)$ and $P_Y(y)$, and their joint probability mass function is denoted by $P(x, y)$. The entropy, $H(X)$, is defined by Equation (1) as:

$$H(x) = \sum_x P_X(x) log(\frac{1}{P_X(x)}). \tag{1}$$

For two discrete random variables X, Y, conditional entropy, $H(X|Y)$ and joint entropy $H(X, Y)$ are defined by Equations (2) and (3) as:

$$H(X|Y) = \sum_y \sum_x P(x, y) log(\frac{P_Y(y)}{P(x, y)}); \tag{2}$$

$$H(X, Y) = \sum_y \sum_x P(x, y) log(\frac{1}{P(x, y)}). \tag{3}$$

Using these notations mutual information MI is defined by Equations (4) and (5) as:

$$MI(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{4}$$

$$= H(X) + H(Y) - H(X, Y). \tag{5}$$

Finally, $MIDI$ is defined by Equation (6) as:

$$MIDI(X, Y) = \frac{MI(X, Y)}{\min(H(X), H(Y))}. \tag{6}$$

Normalized mutual information can be calculated easily for discrete variables. However, in the case of continuous variables, we need to estimate probability density before calculating mutual information and entropy. MIDI uses a histogram scheme that leads to consistent and fast density estimation for calculating the normalized mutual information leading to a smaller execution time.

Ideally, the value of a relation index between two variables should lie in the range $[0, 1]$. Mutual information between two variables is always positive but is not bounded above. Therefore it needs to be normalized. Normalization is often done using entropies of the variable or the cardinality of the partitions used for calculating mutual information. For example, MINE [40] uses the cardinality of partitions to normalize mutual information.

Some indices use the maximum or average of the entropies of the variables [41]. However, normalization by maximum or average entropy results in a low value of the calculated index for non-monotonous relations between variables.

Unlike these indices, MIDI normalizes the mutual information using the minimum of the entropies and can easily identify non-monotonous relations. It attains a value close to 1 if one of the variables is completely determined by the other.

For example, consider the relation $y = \sin(3\pi x)$, where x is drawn from the uniform distribution on range $(0, 1)$. For a set of 10^4 observations drawn from this distribution, MIDI attains a value of 1.0. On the other hand, mutual information normalized by maximum entropy [42] attains a value of 0.2 for this relation and normalization by average entropy leads to the value 0.3 [43]. Another method MINE [40] achieves the value of 1.0 for the example given above by dynamically choosing the partitions for estimating the mutual information and normalizing the estimated value using the number of partitions used for the estimation. However, its execution time is much higher. In the case of the example given above, the execution time for MIDI is 0.005 seconds while MINE takes 5.83 seconds. While finding biclusters these calculations have to be repeated several times making MIDI a better choice for this purpose. The use of MIDI allows us to find biclusters with non-linear and non-monotonous relations.

Algorithm 1: DenseRegions().

```
/* Finding dense regions in 2-dim space                                   */
/* N is the total number of observations                                  */
Find histogram cells, where the number of cells along each axis is given by 3 log(N)
for each cell in the grid do
    if density(cell) is greater than marginal densities then
        Mark as dense
for each region marked as dense do
    if At least one adjoining cell is marked dense then
        Mark the observations in the cell as True
Return state of all observations
```

Algorithm 2: GetBiClusters().

```
/* This algorithm discusses the process of obtaining biclusters from dense regions obtained by Algorithm 1   */
/* M is the total number of features                                      */
Normalize data using the procedure given in Section 4.4
/* Each pair of dimensions may contain several sets representing dense regions */
for each pair of features do
    Find observations forming dense regions using Algorithm 1
Find the number of observations in dense regions for each feature pair
/* There are M(M − 1)/2 values                                            */
Plot a histogram for these values with the number of bins given by 3 log_2(M(M − 1)/2). Ignore first bin
Find the first bin b_lim such that two consecutive bins after b_lim are populated. ObsMin = lower edge of the bin b_lim
for each set of 3 features < f_1, f_2, f_3 > do
    Find the intersection of observations forming dense regions for all 3 pairs < f_1, f_2 >< f_2, f_3 >< f_3, f_1 >
    if resulting set is larger than ObsMin then
        Store observation set as binary array
Let the number of observation sets be NtripleSets
/* The function ceil(x) returns the smallest integer greater than or equal to x */
Perform single-linkage clustering of observation sets with the maximum number of clusters given by NclusObsMax = ceil(3 + ObsMin/NtripleSets)
for each cluster obtained do
    Calculate the bicluster observation set given by observations occurring in more than 1/NclusObsMax of the 'Triplet observation sets' in the cluster
for each bicluster observation set found in the previous step do
    Calculate feature dependence (MIDI) using all features in the dataset and selected observations Perform single-linkage clustering using distance
      1 − MIDI between features, and select the largest feature cluster as the feature set for the bicluster
Return biclusters obtained
```

4.4. Algorithm

This section presents the algorithm and some details required for the implementation of PF-RelDenBi. For finding dense regions, we need to normalize the data to $[0,1]\times[0,1]$. We use a common procedure to normalize the data, which has also been used in CBSC [5]. It is to be mentioned that the data may come from a distribution with an unbounded base or a bounded base with an unknown range. As mentioned in Section 3, let the data be represented by matrix D. The element in the i^{th} row and the j^{th} column is denoted by $e_{i,j}$. To normalize data from different distributions to the range $[0,1]$ we have applied one of the following two transformations, presented in Equations (7) and (8):

$$norm_1(e_{i,j}) = \frac{e_{i,j} - \min\limits_{1\le k\le n}(e_{k,j})}{\max\limits_{1\le k\le n}(e_{k,j}) - \min\limits_{1\le k\le n}(e_{k,j})}; \tag{7}$$

$$norm_2(e_{i,j}) = norm_1(\tan^{-1}(e_{i,j})/\pi + 0.5). \tag{8}$$

The transformation $norm_1(x)$ (x represents elements of input data matrix) is commonly used for mapping data from a bounded base to $[0,1]$ interval while, transformation $norm_2(x)$ maps the interval $(-\infty,\infty)$ to interval $(0,1)$, which is a subset of $[0,1]$. The latter transformation is capable of mapping points from an unbounded region to a bounded interval. Thus it can be applied preferably if data follows distributions like Gaussian, which have a non-compact base.

The next step is to find the dense regions using the procedure described in Section 4.1. The pseudo-code is given as Algorithm 1. Once we have obtained dense regions in two-dimensional spaces, this information is used to obtain biclusters in higher dimensions. The details for the procedure are given in Sections 4.2 and 4.3, and the pseudo-code is given as Algorithm 2.

5. Results on simulated datasets

To test the efficacy of the proposed method we have conducted experiments on both simulated and real-world datasets. Experiments with simulated datasets allow us to study the behaviour of the proposed method on datasets with known characteristics. To check whether the proposed biclustering algorithm can detect biclusters based on linear and non-linear relationships, we have experimented with several simulated datasets. We also wanted to see the effect of applying different data transforms on the performance of the proposed algorithm.

Each column in Tables 1, 2, 3 and 4, corresponds to datasets with particular characteristics. For each column, ten simulated dataset instances are generated by the method given in Appendix A and the average accuracy and standard deviation obtained by different algorithms are reported.

5.1. Methods used for comparison with the proposed method

The performance of the proposed algorithm is compared with eleven other algorithms namely, CLIQUE [25], Proclus [44], ITL [45], Subclu [24], P3C [46], UniBic [4], CBSC [5], ROCCO [35], RelDenClu [34], MESBC [38] and ARBic [30]. CLIQUE, P3C and Subclu are density-based methods. ITL is an information-theoretic method, while Proclus is known to provide stable results. ROCCO is a parameter-free co-clustering method. ARBic is a recent biclustering method for finding feature relation biclusters which is shown to have better results in comparison with several established methods like FABIA, OPSM and UniBic. It also shows better results in comparison with other recent methods like Qubic2 [32]. Comparison with UniBic, CBSC, RelDenClu and ARBic is important as they specifically aim at finding biclusters based on non-linear dependence. MESBC is a recent biclustering method that uses spectral analysis to find biclusters. The single parameter it requires from the user is the number of clusters in the dataset. This parameter is set to the minimum value of 2 for MSBEC as we know that each of the simulated datasets, contains a single embedded bicluster. For ARBic, the implementation provided by the author [30] is used and the default parameters are used.

Some of the parameters used by RelDenClu and CBSC are similar. The authors of these methods have provided a range of parameters that can be used. We have conducted our experiments with several different values for these methods and reported the average performance. The parameter $MinSeedSize$ for RelDenClu should be smaller than the number of observations in the biclusters. For this parameter, we have taken the values $\{N/8, N/16\}$ where N is the number of observations in the dataset. The author recommends that the value of $ObsInMinBase$ can be 3 or more. We have experimented with the values $\{3,5\}$. The author recommends a range $(0.6, 0.98)$ for the parameter $Sim2Seed$ and we have taken the values $\{0.65, 0.8, 0.95\}$. The parameter $ReuseAllSeeds$ can be set to $True$ or $False$. We have experimented with both of these values. When this parameter is set to $False$, another parameter $ReuseSeedSim$ is required. The author recommends its range as $(0,1)$. We have experimented with the values $\{0.25, 0.5, 0.75\}$. The parameter $ClusSim$ is used to filter similar biclusters. Its value can lie in the range $[0,1]$. We have experimented with values $\{0.5, 1\}$. We have used all the combinations of the parameter values mentioned above and reported the average performance.

For CBSC, the parameter $MinCompts$ should be smaller than the number of observations in the biclusters. For this parameter, we have taken the values $\{N/8, N/16\}$ where N is the number of observations in the dataset. For $RatioMerge$, the author recommends a range of $(0.75, 0.99)$, we have taken the values $\{0.75, 0.85, 0.95\}$. The parameter $Reuse\_All\_Bases$ can be set to $True$ or $False$. We have experimented with both of these values. When this parameter is set to $False$, another parameter $Reuse\_Base\_Ratio$ is required. The author recommends its range as $(0,1)$. We have experimented with the values $\{0.25, 0.5, 0.75\}$ when $Reuse\_All\_Bases = False$. The parameter $Clus\_Overlap$ is used to filter similar biclusters. Its value can lie in the range $[0,1]$. We have experimented with values $\{0.5, 1\}$. For density estimates, CBSC requires two more parameters c and n_g. The author recommends the value $n_g = 10$ and

Table 1
Accuracy for Non-Linear simulated datasets (Mean and Standard Deviation for 10 datasets of each type).

Method	Accuracy	Datasets		
		Non-Monotonous	Non-Linear 1	Non-Linear 2
Proposed	Mean	0.973	0.939	0.927
	Std. Dev.	0.025	0.034	0.005
MSBEC	Mean	0.671	0.767	0.704
	Std. Dev.	0.023	0.001	0.043
ARBic	Mean	0.76	0.848	0.867
	Std. Dev.	0.022	0.029	0.040
RelDenClu	Mean	0.786	0.912	0.913
	Std. Dev.	0.069	0.077	0.031
ROCCO	Mean	0.751	0.82	0.836
	Std. Dev.	0.020	0.015	0.020
CBSC	Mean	0.79	0.827	0.892
	Std. Dev.	0.086	0.049	0.022
UniBic	Mean	0.764	0.804	0.839
	Std. Dev.	0.004	0.142	0.107
P3C	Mean	0.75	0.765	0.764
	Std. Dev.	0.001	0.004	0.003
Subclu	Mean	0.746	0.785	0.793
	Std. Dev.	0.005	0.017	0.016
ITL	Mean	0.75	0.75	0.75
	Std. Dev.	0.009	0.004	0.002
Proclus	Mean	0.752	0.781	0.792
	Std. Dev.	0.009	0.014	0.007
CLIQUE	Mean	0.75	0.784	0.782
	Std. Dev.	0.000	0.002	0.002

the value for c in the range $(0.75, 0.99)$. We have taken $n_g = 10$ as suggested by the author. For c we have considered the values $\{0.75, 0.85, 0.95\}$. With CBSC too we have experimented with all combinations of above mentioned parameter values.

For ROCCO the parameter k is required for finding the k nearest neighbours adjacency matrix for spectral analysis. The authors have shown that the performance of ROCCO is consistent for any of the values of k from 8, 9 or 10, making it a hyperparameter-free algorithm [35]. Thus, we have chosen $k = 9$ for our experiments.

For UniBic, P3C, SubClu, ProClus and CLIQUE the R packages 'runibic' and 'subspace' have been used for the experiments with default parameters [47,48]. For ITL, the Matlab Toolbox for Biclustering Analysis has been used with default parameters [49].

5.2. Evaluation of performance on simulated datasets

For the simulated datasets, the evaluation is done in the following manner. For data represented as an $N \times M$ matrix, accuracy is calculated as follows: For each element in the dataset let $e_i^j = 1$ if the element in row i and column j is selected, otherwise $e_i^j = 0$. Similarly, for each element in the dataset let $a_i^j = 1$ if the element belongs to the actual bicluster and $a_i^j = 0$ otherwise. The accuracy is calculated as $sum(XNOR(e_i^j, a_i^j))/(NM)$. In other words, an element is said to be correctly identified if it is either present in the actual bicluster and also in the bicluster found by the algorithm or if the element is absent in both of them. The expectation of correctness over all the elements of the data matrix is the overall accuracy while identifying a bicluster. For each bicluster, present in the data, the best match is reported for each algorithm. These results are reported in Section 5.3 and corresponding execution times are reported in Section 5.4.

5.3. Accuracy for simulated datasets

For simulated data, experiments are conducted with 10 different simulations and the average (over 10 simulations) accuracy and corresponding standard deviation (denoted as "deviation") values are reported in Tables 1, 2, 3 and 4. Comparison is done with eleven other methods. We see that the proposed method provides better accuracy for fifteen out of the sixteen simulated datasets. Data is normalized using $norm_2()$ for "Normal" and "Noisy Normal" datasets. For all other datasets $norm_1()$ is used.

The columns ("Non-Monotonous", "Non-linear 1" and "Non-linear 2") of Table 1 represent datasets having biclusters based on non-linear relations. We see that the proposed algorithm provides higher accuracy for these three datasets. Unlike other algorithms,

Table 2
Accuracy for Base and transformed simulated datasets (Mean and Standard Deviation for 10 datasets of each type).

Method	Accuracy	Datasets							
		Base	Scaled	Translated	Linear Transform	Point Proportion	Cluster Proportion	Noisy Uniform	Permutations
Proposed	Mean	0.986	0.983	0.985	0.986	0.984	0.981	0.984	0.986
	Std. Dev.	0.036	0.036	0.038	0.037	0.026	0.026	0.032	0.036
MSBEC	Mean	0.600	0.601	0.604	0.654	0.603	0.641	0.595	0.600
	Std. Dev.	0.041	0.041	0.059	0.033	0.037	0.042	0.054	0.034
ARBic	Mean	0.968	0.973	0.857	0.633	0.968	0.964	0.874	0.867
	Std. Dev.	0.032	0.060	0.100	0.103	0.032	0.036	0.051	0.033
RelDenClu	Mean	0.980	0.980	0.981	0.981	0.980	0.980	0.974	0.980
	Std. Dev.	0.057	0.057	0.073	0.057	0.042	0.051	0.058	0.057
ROCCO	Mean	0.813	0.881	0.883	0.851	0.881	0.829	0.839	0.819
	Std. Dev.	0.019	0.015	0.014	0.020	0.034	0.036	0.012	0.019
CBSC	Mean	0.914	0.914	0.913	0.914	0.915	0.924	0.906	0.913
	Std. Dev.	0.063	0.063	0.063	0.062	0.068	0.070	0.049	0.063
UniBic	Mean	0.900	0.848	0.802	0.793	0.603	0.641	0.832	0.900
	Std. Dev.	0.034	0.211	0.159	0.034	0.046	0.078	0.124	0.034
P3C	Mean	0.781	0.782	0.779	0.779	0.787	0.802	0.772	0.783
	Std. Dev.	0.040	0.040	0.041	0.041	0.022	0.092	0.032	0.015
Subclu	Mean	0.794	0.759	0.788	0.765	0.802	0.724	0.795	0.798
	Std. Dev.	0.023	0.016	0.019	0.025	0.030	0.020	0.017	0.022
ITL	Mean	0.751	0.752	0.750	0.751	0.750	0.750	0.750	0.750
	Std. Dev.	0.005	0.004	0.004	0.001	0.001	0.002	0.002	0.001
Proclus	Mean	0.785	0.768	0.803	0.770	0.800	0.711	0.792	0.785
	Std. Dev.	0.013	0.015	0.023	0.028	0.024	0.024	0.010	0.015
CLIQUE	Mean	0.776	0.776	0.776	0.776	0.776	0.683	0.720	0.773
	Std. Dev.	0.016	0.016	0.016	0.016	0.016	0.012	0.011	0.015

the proposed method finds the biclusters accurately for datasets containing highly non-monotonous relations as seen from the first column. Despite the highly variable distribution of background observations, the proposed method finds the biclusters with higher accuracy as seen from the column “Non-Linear 1”. Its accuracy for another dataset containing bicluster based on non-linear relations is also higher in comparison with other algorithms as seen from column “Non-Linear 2”. The values of accuracy obtained using different algorithms on these datasets are presented in Fig. 1 as bar-graphs, for better illustration. The experiments on these three datasets illustrate that the proposed method can find biclusters where features are related to each other with non-linear or non-monotonous relations as mentioned in Section 2.2. The proposed method is able to identify these relations due to its reliance on relative density to find the initial set of observations and the use of the relation index MIDI to find the final biclusters.

The first column of Table 2 (“Base”) represents a dataset containing biclusters based on linear relation. The other seven columns of the table (“Scaled”, “Translated”, ..., “Permutations”) give results for datasets obtained by applying different transforms to the “Base” dataset. These datasets allow us to analyze the behaviour of the proposed method in a way similar to the analysis done by Fisher and Ness [50] for clustering algorithms. From Fig. 2 (shown in parts 2a and 2b) and the above-mentioned table, we find that the proposed method has produced better results as compared to other methods for the “Base” dataset and its transforms, i.e. “Scaled”,..., “Permutations”.

The three columns of Table 3 (“Normal”, “Noisy Normal 1” and “Noisy Normal 2”), contain results for datasets generated using the normal distribution. In the case of the “Normal” dataset, we find that ARBic has slightly better performance than the proposed method. But from the results on the “Noisy Normal 1” (noise with standard deviation 0.1) and “Noisy Normal 2” (noise with standard deviation 0.2) datasets, we find that the performance of ARBic decreases more sharply when noise is added. On the other hand, the proposed method shows a more robust performance and achieves higher accuracy than other methods on “Noisy Normal 1” and “Noisy Normal 2” datasets. The performance for these datasets is reported in Table 3 and Fig. 3.

The first two columns of Table 4 (“Overlap 1”), present results for a dataset containing two biclusters with an overlap of 12%. We find that the proposed method has the best performance among the methods used for comparison. The third and fourth columns present results for a similar dataset with two biclusters but with an overlap of 20%. In this case, we find that RelDenClu finds the second bicluster with the highest accuracy. On the other hand, the proposed method finds the first bicluster with the highest accuracy. If we take the average over both biclusters in the dataset “Overlap 2”, the proposed method achieves the highest accuracy. These results are presented graphically in Fig. 4.

These results for simulated datasets suggest some properties of the proposed method which will be discussed in Section 5.5.

Table 3
Accuracy for simulated datasets with Normal Distribution (Mean and Standard Deviation for 10 datasets of each type).

Method	Accuracy	Datasets		
		Normal	Noisy Normal 1	Noisy Normal 2
Proposed	Mean	0.971	0.964	0.934
	Std. Dev.	0.022	0.015	0.020
MSBEC	Mean	0.746	0.747	0.746
	Std. Dev.	0.009	0.009	0.010
ARBic	Mean	0.975	0.812	0.784
	Std. Dev.	0.010	0.009	0.002
RelDenClu	Mean	0.958	0.941	0.810
	Std. Dev.	0.043	0.074	0.010
ROCCO	Mean	0.753	0.755	0.752
	Std. Dev.	0.011	0.009	0.003
CBSC	Mean	0.951	0.950	0.886
	Std. Dev.	0.075	0.074	0.074
UniBic	Mean	0.869	0.794	0.776
	Std. Dev.	0.025	0.098	0.009
P3C	Mean	0.767	0.758	0.765
	Std. Dev.	0.017	0.047	0.009
Subclu	Mean	0.783	0.78	0.769
	Std. Dev.	0.010	0.015	0.006
ITL	Mean	0.75	0.749	0.750
	Std. Dev.	0.003	0.001	0.001
Proclus	Mean	0.783	0.772	0.769
	Std. Dev.	0.017	0.011	0.006
CLIQUE	Mean	0.799	0.787	0.776
	Std. Dev.	0.011	0.015	0.008

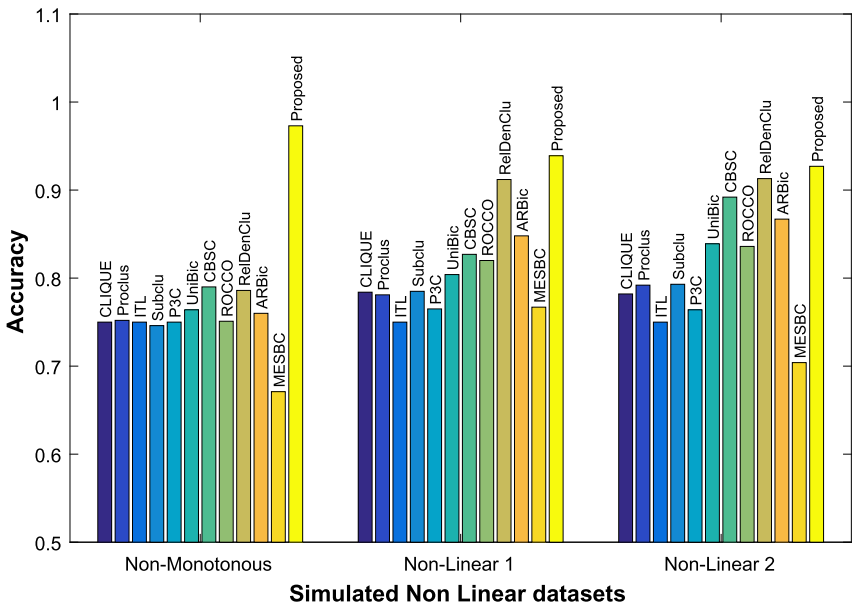


Fig. 1. Accuracy obtained using various algorithms for Non-Linear datasets.

Table 4
Accuracy for Overlapping simulated datasets (Mean and Standard Deviation for 10 datasets of each type).

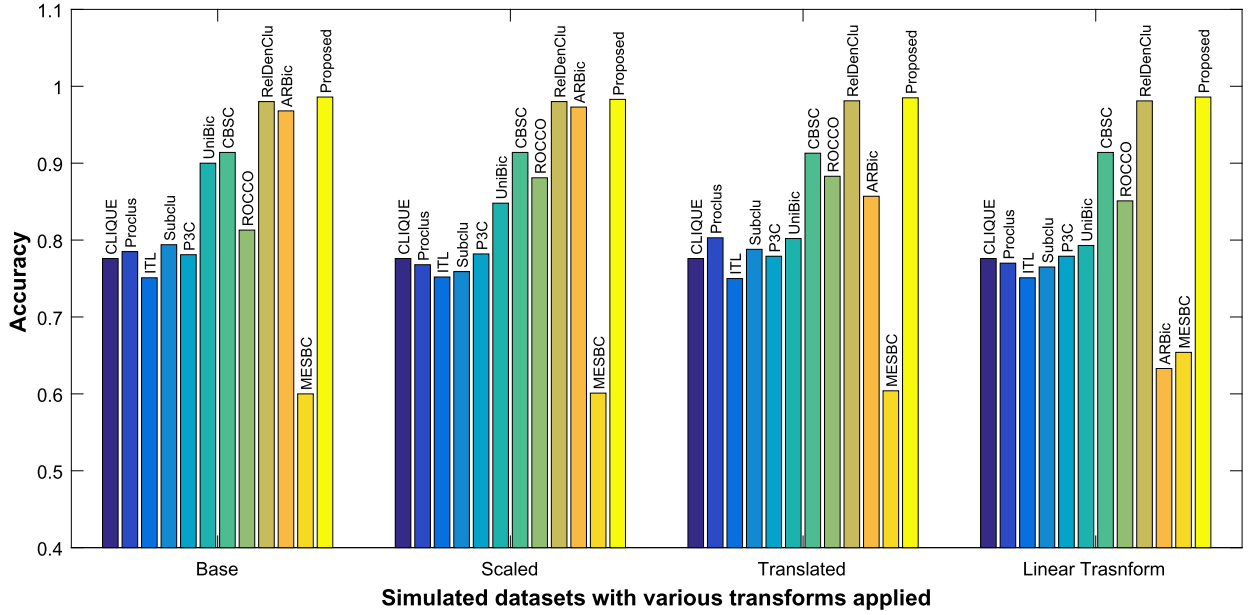
Method	Accuracy	Datasets			
		Overlap 1		Overlap 2	
		First	Second	First	Second
Proposed	Mean	0.994	0.999	0.973	0.955
	Std. Dev.	0.010	0.001	0.028	0.108
MESBC	Mean	0.727	0.779	0.701	0.751
	Std. Dev.	0.015	0.009	0.005	0.005
ARBic	Mean	0.927	0.584	0.905	0.603
	Std. Dev.	0.029	0.020	0.031	0.016
RelDenClu	Mean	0.980	0.996	0.906	0.965
	Std. Dev.	0.004	0.002	0.002	0.004
ROCCO	Mean	0.773	0.817	0.751	0.801
	Std. Dev.	0.008	0.007	0.002	0.002
CBSC	Mean	0.924	0.935	0.872	0.883
	Std. Dev.	0.036	0.006	0.012	0.012
UniBic	Mean	0.940	0.626	0.833	0.775
	Std. Dev.	0.037	0.032	0.130	0.143
P3C	Mean	0.751	0.801	0.750	0.800
	Std. Dev.	0.002	0.004	0.001	0.001
Subclu	Mean	0.761	0.801	0.763	0.801
	Std. Dev.	0.005	0.002	0.004	0.002
ITL	Mean	0.750	0.800	0.750	0.800
	Std. Dev.	0.001	0.001	0.001	0.001
Proclus	Mean	0.760	0.804	0.761	0.802
	Std. Dev.	0.009	0.004	0.008	0.003
CLIQUE	Mean	0.750	0.800	0.750	0.800
	Std. Dev.	0.001	0.001	0.001	0.001

Table 5
Execution time in seconds for Non-Linear simulated datasets.

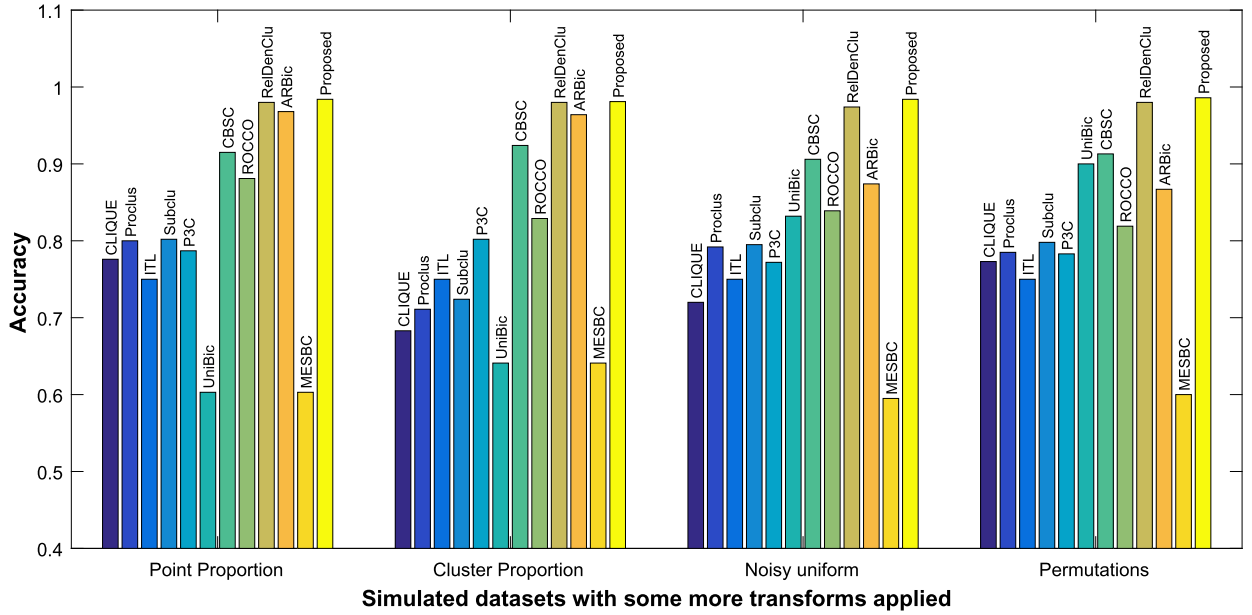
Method	Datasets		
	Non-Monotonous	Non-Linear 1	Non-Linear2
Proposed	0.287	0.221	0.232
MSBEC	0.021	0.069	0.075
ARBic	0.258	0.232	0.254
RelDenClu	0.195	0.295	0.275
ROCCO	31.402	4.087	3.164
CBSC	19.294	37.464	31.82
UniBIc	0.012	0.583	0.612
P3C	1.440	0.359	0.650
SubClu	0.037	0.064	0.052
ITL	15.000	9.096	9.248
Proclus	0.026	0.060	0.055
Clique	0.001	0.011	0.007

5.4. Execution time for simulated datasets and time complexity

Execution time is considered as another evaluation criterion and the results are reported in Tables 5, 6, 7 and 8. The datasets used here are the same as those used in Tables 1, 2, 3 and 4. The algorithm is executed with Intel Core i7 CPU and 8 GB memory. As seen from the above-mentioned tables, the execution time for the proposed method is less than ITL, P3C, UniBic and CBSC. The execution time of ARBic and the proposed method are similar. It is high compared to MESBC and density-based methods like Subclu, Proclus, and CLIQUE. However, it is already seen in Tables 1, 2, 3 and 4, that the accuracy of the proposed method is better than these methods for most of the simulated datasets.



(a) Accuracy obtained for Base dataset and its transforms



(b) Accuracy for 4 more transforms applied to Base dataset

Fig. 2. Accuracy obtained for simulated datasets with various transforms applied.

5.4.1. Time complexity of the proposed method

The time complexity of ITL has been reported by Dhillon et al. [45] and Liu et al. have reported the time complexity of ARBic [30]. The time complexity of ROCCO has been reported by He and Moreira-Matias [35]. For MESBC, the most computationally expensive operation is to calculate the Singular Value Decomposition which is known to have a complexity of MN^2, where N and M, respectively are the number of observations and the number of features. However faster approximations have recently been designed which may lead to faster implementations [51]. The time complexity of other methods used for comparison has been reported by Jain and Murthy [5]. We are reporting the time complexity of the proposed method, where N and M, respectively are the number of observations and the number of features in a given dataset.

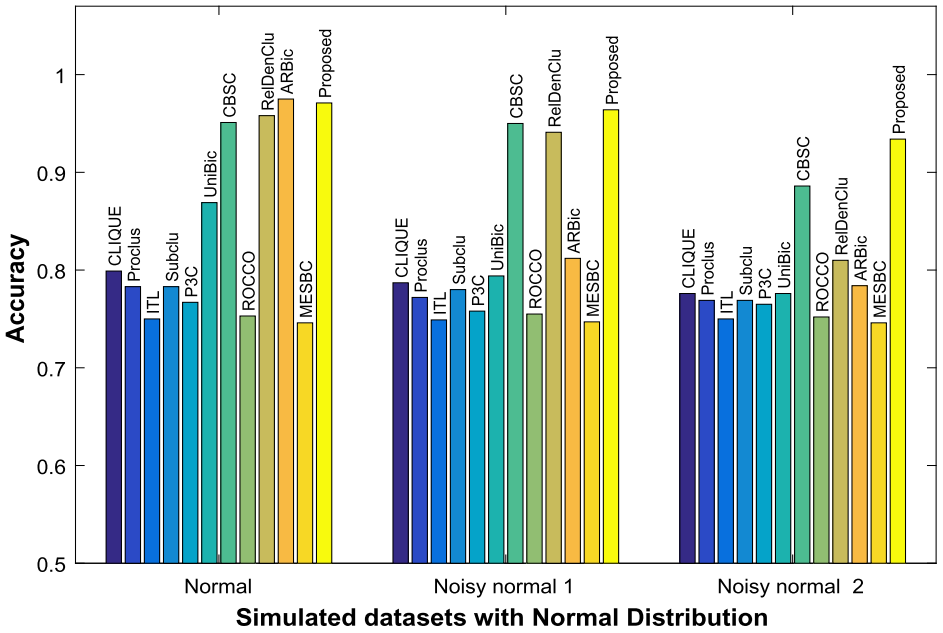


Fig. 3. Accuracy obtained using various algorithms for Normal and Noisy Normal datasets.

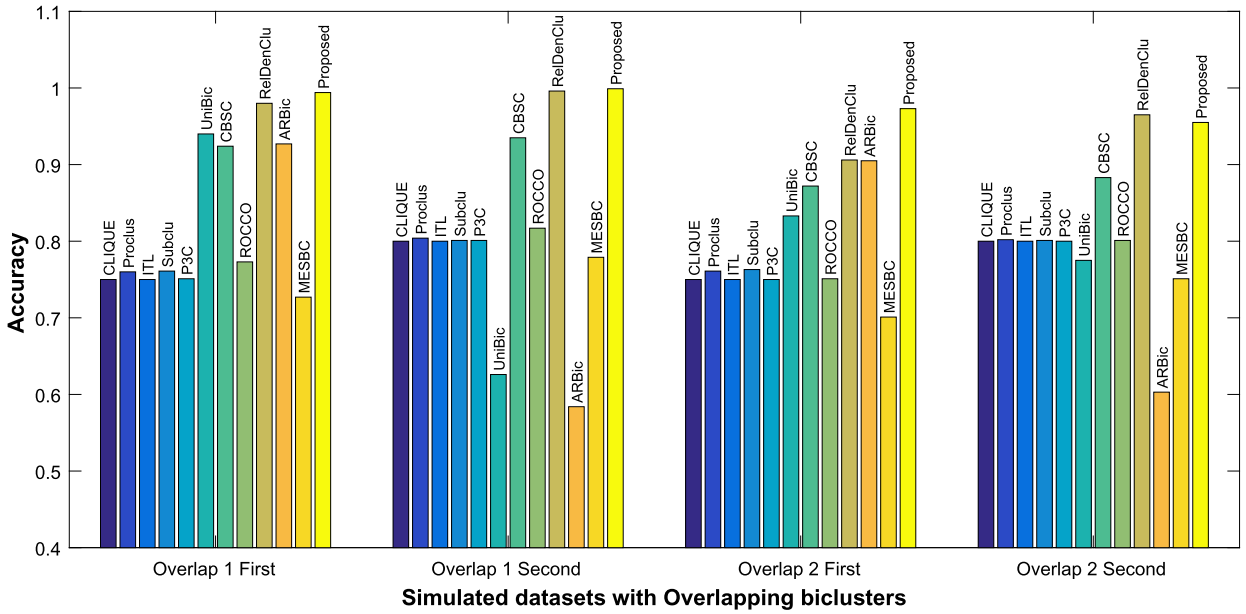


Fig. 4. Accuracy obtained using various algorithms for Overlapping datasets.

The time complexity of the method proposed in this article is $(log(N)M)^2 + NM^3$ where N and M represent the number of observations and features in the dataset, respectively. For finding dense regions the complexity is given by $(log(N)M)^2$. The complexity is NM^3 for the pruning step because the intersection of observations is calculated for a set of 3 dimensions. However, this is the worst-case scenario and in reality, we do not check for all feature triplets but only find the intersections for dense regions calculated previously.

Theoretically, the time complexity of CBSC [5] and the proposed method are similar. However, the execution time for the proposed method is found to be much lower as compared to CBSC for the high dimensional dataset. This happens because the proposed method does not calculate the Minimal Spanning Tree for each pair of dimensions, which is done in CBSC. The time complexity of the proposed method is similar to RelDenClu. However, in practical applications, the proposed method may save time, as the user would not need to experiment with different parameter settings.

Table 6
Execution time in seconds for Base and transformed simulated datasets.

Method	Datasets							
	Base	Scale	Translated	Linear	Point Proportion	Cluster Proportion	Noisy	Permutations
Proposed	0.250	0.238	0.243	0.236	0.488	0.549	0.228	0.243
MSBEC	0.104	0.089	0.063	0.071	0.209	0.153	0.073	0.136
ARBic	0.254	0.229	0.248	0.282	2.751	2.105	0.287	0.216
RelDenClu	0.293	0.291	0.273	0.302	11.103	7.768	0.288	0.293
ROCCO	2.939	3.021	2.681	2.794	29.275	7.509	2.874	2.958
CBSC	42.416	51.400	43.964	41.313	91.973	83.796	42.304	41.654
UniBIc	0.583	0.612	0.649	0.673	2.596	1.436	0.621	0.583
P3C	0.513	0.955	0.952	0.433	1.083	0.848	0.458	0.588
SubClu	0.056	0.043	0.050	0.055	0.120	0.082	0.049	0.070
ITL	3.171	3.302	3.498	3.276	9.994	6.519	2.94	3.272
Proclus	0.052	0.055	0.049	0.055	0.120	0.088	0.056	0.063
Clique	0.105	0.015	0.015	0.018	0.033	0.021	0.014	0.013

Table 7
Execution time in seconds for simulated datasets with Normal Distribution.

Method	Datasets		
	Normal	Noisy Normal 1	Noisy Normal 2
Proposed	0.234	0.225	0.296
MSBEC	0.069	0.076	0.859
ARBic	0.288	0.264	0.918
RelDenClu	1.398	1.395	71.117
ROCCO	6.258	5.128	2.938
CBSC	66.659	63.821	267.958
UniBic	0.531	0.577	0.001
P3C	18.840	9.751	6.322
SubClu	0.071	0.061	0.050
ITL	1.187	1.259	20.639
Proclus	0.054	0.052	0.084
Clique	0.071	0.044	0.715

Table 8
Execution time in seconds for Overlapping simulated datasets.

Method	Datasets	
	Overlap 1	Overlap 2
Proposed	0.379	0.430
MESBC	0.240	0.142
ARBic	0.590	0.811
RelDenClu	27.090	27.103
ROCCO	2.571	25.663
CBSC	110.521	115.285
UniBic	2.092	2.775
P3C	0.049	0.016
Subclu	0.074	0.158
ITL	15.102	15.091
Proclus	0.074	0.087
CLIQUE	0.014	0.032

5.5. A discussion on properties of PF-RelDenBi as seen from experiments on simulated datasets

It is already seen that the proposed algorithm can find relation-based biclusters with better accuracy as compared to other algorithms for most of the simulated datasets as it is seen from Tables 1, 2, 3, 4 and Figs. 1, 2, 3 and 4. In this section, we discuss some of the observed properties of the proposed algorithm.

The proposed algorithm is invariant to scaling, translation and linear transforms because these transforms affect neither bin length nor estimated densities. Theoretically, the procedure is not invariant to arbitrary order-preserving transforms, which may change the density of the observations. We compare the accuracy obtained for the “Base” dataset and its various transforms. We find that there is no significant difference in the performance of PF-RelDenBi for “Scaled”, “Translated”, “Linear Transform”, “Point Proportion”, “Cluster Proportion” and “Permutations” datasets. Accuracy for the “Noisy Uniform”, “Noisy Normal 1” and “Noisy Normal 2” datasets

is only slightly lower than the "Base" and "Normal" datasets. As compared to other algorithms the proposed method still has higher accuracy for noisy datasets.

We may note, that the biclusters in "Base" dataset can also be seen as scaling biclusters, as each column in the bicluster can be obtained by multiplying another column by a constant value. Further, the biclusters in "Translated" datasets are obtained by adding a randomly selected constant value to each column of "Base" dataset. Hence, the resulting biclusters can be seen as a combination of scaling and shift biclusters. This happens because scaling and shifting are examples of linear and affine transformations, respectively. Since the proposed method finds biclusters based on feature relations, it will be able to find biclusters based on scaling and shifting of columns. However, the proposed method will not be able to find biclusters based on the shifting and scaling of rows, as it specifically aims to find biclusters based on feature relations.

Overall, we find that the proposed method PF-RelDenBi performs well on data after applying different transforms or adding noise. Its performance on datasets containing non-linear datasets is also good, which is in line with its objectives discussed in Section 3.

6. Experiments with real-world datasets

In this section, we demonstrate the use of the proposed method in three different ways. First, as an application to supervised learning, the biclustering algorithms have been used to generate new features for two different datasets that lead to improved classification accuracy. This is discussed in Section 6.1

Second, in Section 6.2, we use the proposed method for clustering three datasets and then compare the obtained clusters with known classes in the data. The datasets used for these experiments are taken from the UCI ML repository [52] and are suitable for the task of classification. These datasets are from diverse application areas namely, Banking, Education, Medical Science, Particle Physics, and Electrical Engineering. For the chosen datasets each feature is presented in numeric form. Each of these datasets has at least 600 observations and a minimum of 9 features. For each application area, the dataset with the highest number of views is selected. The number of views as of 7 May 2024, for each of the datasets, is given along with the description of the datasets in Sections 6.1 and 6.2.

Third, we use the proposed method for identifying factors impacting the spread of COVID across different countries. This is discussed in Section 6.3.

For the real-world datasets, we initially apply the normalization $norm_1$. If no biclusters are produced using $norm_1$ then the biclusters are found after applying $norm_2$ to the dataset.

6.1. Experiments with supervised learning

We have used PF-RelDenBi and other biclustering methods to generate new binary features that are further used to improve the classification performance on two UCI ML datasets [52]. These are: (1)"Default of credit card clients" dataset [53] from the banking sector having 30000 observations with 23 features and two classes; this dataset has 97178 views as of May 7 2024. (2) "Predict Students' Dropout and Academic Success" dataset [54] from the education sector has 4424 observations with 36 features and three classes; this dataset was viewed 128520 times as of 7 May 2024.

Biclusters are found using different methods (proposed and eleven other methods). For a given algorithm, each bicluster obtained results in a new feature. If an observation belongs to this bicluster, the new feature value is taken as 1 otherwise 0 i.e., the corresponding membership value is taken as a new feature. Naive Bayes classifier, which is a very simple classifier is used after adding new features to a database. For MESBC, the number of clusters is taken as the known number of classes in the data. For the methods RelDenClu and CBSC we needed to choose one set of parameter values for which the biclusters could be used to generate new features. The choice is made by selecting parameter values using 10-fold cross-validation from the combinations of parameters mentioned in Section 5.3. All other parameters for various algorithms are the same as those discussed in Section 5.3. The performance is evaluated in terms of Accuracy, Normalized Mutual Information (NMI) [41], Adjusted Rand Index (ARI) [55], Precision, Recall and G-Score [56]. Note that ARI can sometimes attain negative values and the lower bounds have been well studied [57]. Also note that, since average Precision, Recall and G-Score are being reported, the reported G-Score values may be less than the geometric mean of reported Precision and Recall. The results are reported in Table 9 and the above-mentioned performance indices are presented as bar-graphs in Figs. 5a, 5b, 5c, 5d, 5e, 5f. We find that for the Credit Card dataset, the features generated by PF-RelDenBi provide a greater improvement in accuracy as compared to eleven other algorithms, thereby leading to better identification of credit card defaulters. RelDenClu and CBSC also resulted in similar improvements. However, this was achieved only after determining the parameters using cross-validation, making the process more computationally costly and time-consuming. For the Student Dropout dataset, the proposed method provides improvement over the original accuracy. It also results in better NMI and ARI. Here too RelDenClu provides similar improvement only after determining the parameters with cross-validation. Overall, we find that the proposed method provides better accuracy without the hassle of finding suitable parameters.

6.2. Experimental results for unsupervised learning

We apply the proposed method for unsupervised learning to see whether the biclusters detected by the proposed algorithm correspond to the meaningful structures in data. We check if one of the biclusters detected corresponds to one of the known classes in the data or not. This is done for three UCI ML [52] datasets: (1) "Magic gamma telescope" dataset [58] from the application area named particle physics, has 19020 observations with 10 features; it has 74568 views as of 7 May 2024. (2) "Breast cancer (Wisconsin

Table 9
Classification performance for Credit card and Student Dropout datasets (after adding features generated using different biclustering methods).

Dataset	Credit Card [53]						
Method	Accuracy		NMI	ARI	Average		
	Mean	S. Dev.			Prec.	Recall	G-Score
Proposed	0.775	0.007	0.116	0.241	0.680	0.694	0.686
MESBC	0.697	0.015	0.086	0.140	0.638	0.687	0.650
ARBic	0.736	0.012	0.097	0.188	0.654	0.692	0.668
RelDenClu	0.761	0.006	0.108	0.227	0.672	0.702	0.685
ROCCO	0.269	0.005	0.000	-0.007	0.510	0.504	0.355
CBSC	0.755	0.006	0.074	0.185	0.647	0.651	0.649
UniBic	0.397	0.006	0.015	-0.047	0.559	0.562	0.472
P3C	0.702	0.012	0.090	0.149	0.642	0.690	0.654
Subclu	0.547	0.098	0.033	0.004	0.581	0.616	0.560
ITL	0.269	0.010	0.006	-0.035	0.562	0.520	0.359
Prolus	0.722	0.015	0.097	0.172	0.650	0.696	0.665
CLIQUE	0.544	0.006	0.032	0.003	0.580	0.614	0.558
Original	0.703	0.028	0.088	0.147	0.640	0.689	0.654
Dataset	Student Dropout [54]						
Method	Accuracy		NMI	ARI	Average		
	Mean	S. Dev.			Prec.	Recall	G-Score
Proposed	0.692	0.015	0.222	0.325	0.624	0.600	0.609
MESBC	0.664	0.021	0.192	0.285	0.566	0.554	0.555
ARBic	0.679	0.020	0.209	0.307	0.602	0.583	0.589
RelDenClu	0.685	0.018	0.219	0.317	0.616	0.602	0.608
ROCCO	0.221	0.011	0.010	-0.003	0.434	0.360	0.264
CBSC	0.681	0.016	0.207	0.302	0.604	0.583	0.589
UniBic	0.610	0.009	0.144	0.125	0.550	0.450	0.455
P3C	0.679	0.020	0.215	0.312	0.595	0.581	0.585
Subclu	0.674	0.018	0.197	0.291	0.602	0.586	0.591
ITL	0.185	0.003	0.004	-0.002	0.417	0.336	0.184
Proclus	0.676	0.018	0.206	0.305	0.601	0.589	0.593
CLIQUE	0.675	0.022	0.195	0.295	0.597	0.576	0.582
Original	0.678	0.020	0.204	0.303	0.597	0.580	0.585

Original)" dataset [59] from the sector of medical science has 683 observations and 9 features; it has 121901 views as of 7 May 2024. (3) "Electrical Grid Stability Simulated Data" dataset [60] from the application area electrical engineering, has 10000 observations and 12 features; the dataset has 13797 views as of 7 May 2024. These datasets have numeric features and the associated task is binary classification.

The methods used for comparison and parameter settings are the same as those used in Section 5. The performance evaluation for these datasets is done in the following manner. Each of the datasets used contains two classes labelled as 0 or 1. Let $O_C(i)$ denote whether observation i is in class 1 or 0. Let $O_E(i) = 1$, if observation i is included in the estimated bicluster and 0 otherwise. For each bicluster, the number of matches between the bicluster membership and the class label is calculated as below:

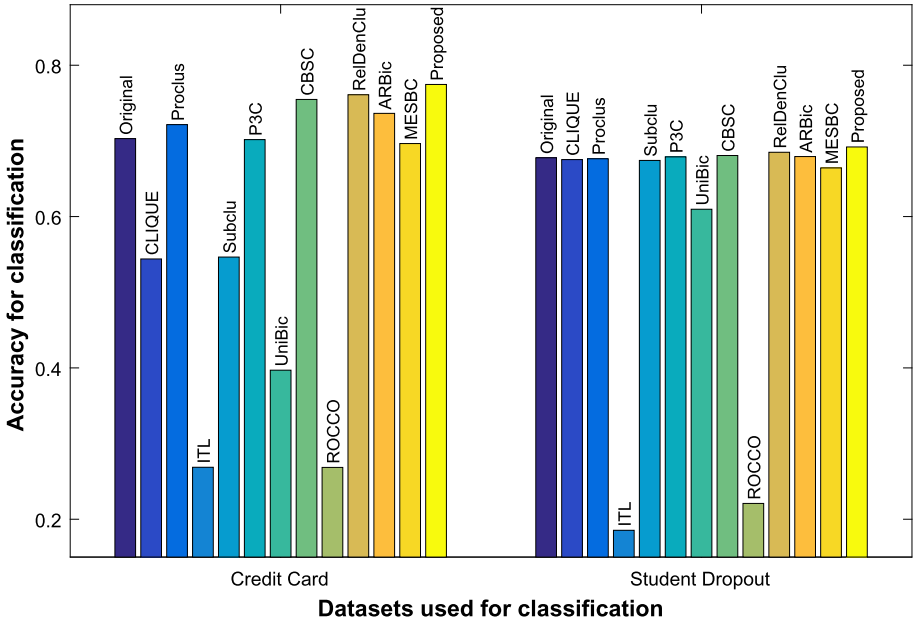
$$Acc = \frac{max(sum(XNOR(O_E(i), O_C(i))), sum(XOR(O_E(i), O_C(i))))}{N}; \tag{9}$$

where N is the number of observations in the dataset and $XNOR$ and XOR are the logical operators. Using this formula we check whether a bicluster is similar to any one of the two classes present in the dataset. For the bicluster that obtains the best accuracy according to Equation (9), six performance indices are reported. These indices are Accuracy, NMI, ARI, Precision, Recall and G-score.

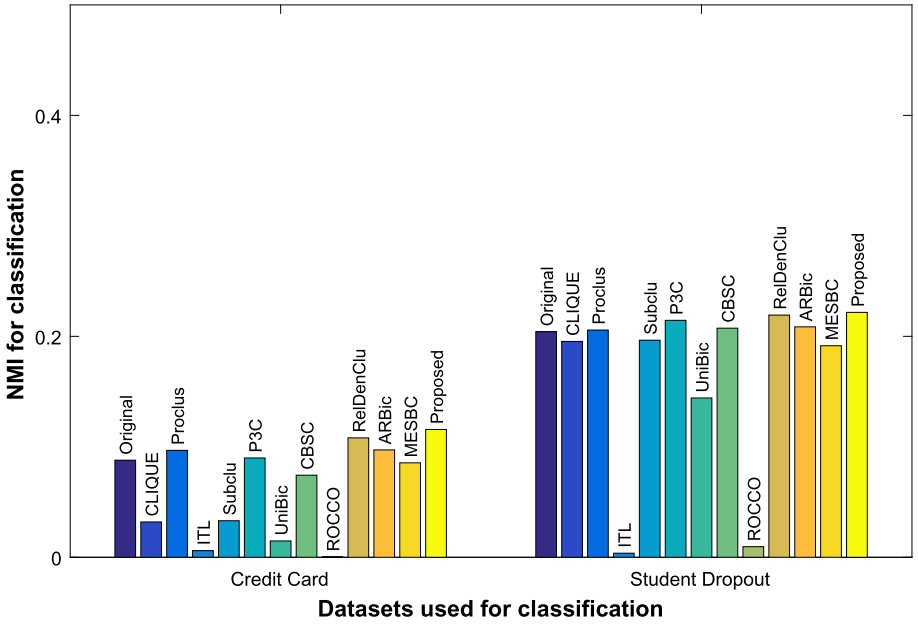
The results are reported in Table 10. It is seen that the proposed method yields the best accuracy for the three datasets used for investigation. For better visualization, the accuracies obtained by different methods are presented in Fig. 6a as bar graphs. NMI and ARI are shown in Figs. 6b and 6c, respectively. The average Precision, Recall and G-score of both classes have been shown in Figs. 6d, 6e and 6f, respectively. We find that the proposed method obtains better accuracy for all three datasets. It obtains the best NMI and ARI for Magic and Breast Cancer datasets. It does not achieve a better G-Score for each class individually but achieves a higher G-Score averaged over the two classes for Magic and Breast Cancer datasets.

6.3. Applicability of the proposed method for COVID-19

We have also utilized the proposed method to find lifestyle factors impacting COVID-19 infection rates. Toward the end of 2019, a highly contagious disease named COVID-19 emerged and caused a pandemic of unprecedented scale. However, it is seen that some countries were affected by this disease to a greater extent than others. During the early phase of the spread of COVID-19, it was



(a) Accuracy of various biclustering algorithms for Supervised Learning

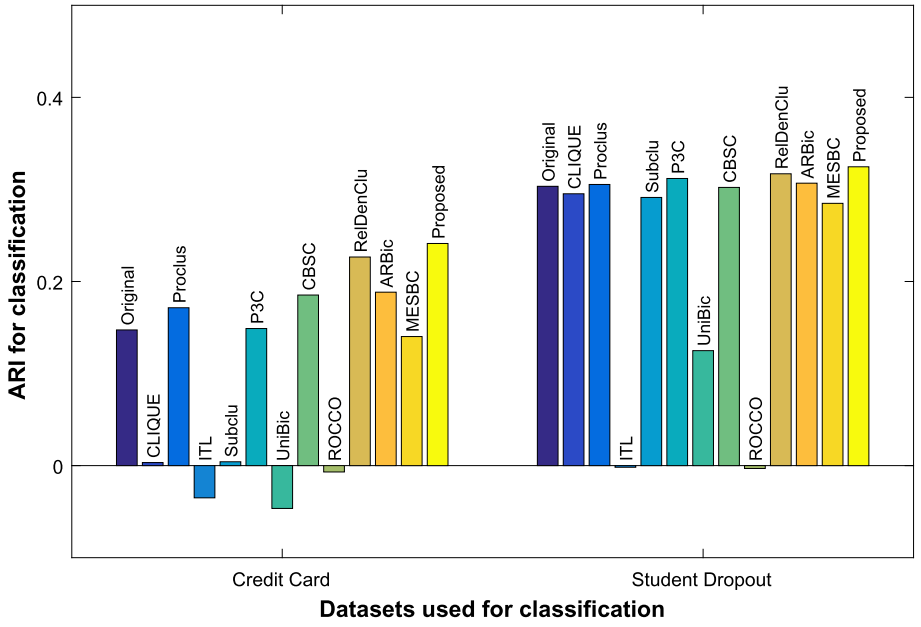


(b) NMI obtained by various algorithms for Supervised Learning

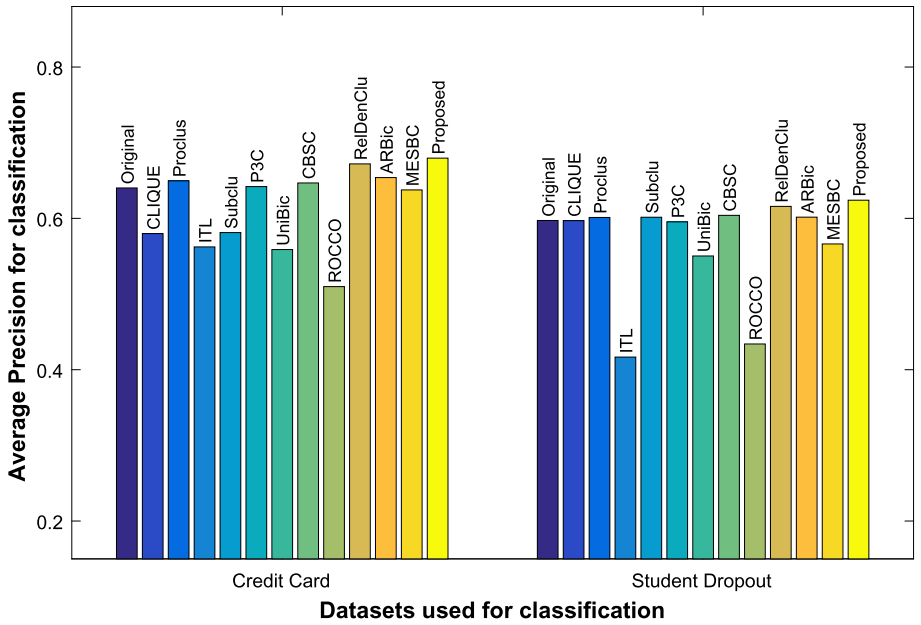
Fig. 5. Performance of various biclustering algorithms for Supervised Learning.

important for policymakers and medical practitioners to understand the behaviour of the disease. For this, there were attempts to find the relation between existing lifestyle factors and COVID-19 rates.

One such dataset was created by selecting features from World Development Indicators taken from the site of the World Bank [61] and the script for generating the dataset was made available at Kaggle [62]. In brief, the creator of the dataset considered 74 features from the World Bank Indicators dataset, that intuitively seemed related to COVID-19 infection rates and dropped the features in which the missing values were more than 25%. This resulted in a WDI (World Bank Indicators) dataset with 27 columns. The missing values were treated using KNN imputation. The creator then joined the WDI dataset with the COVID-19 rate dataset for correlation



(c) ARI obtained by various biclustering methods for Supervised Learning



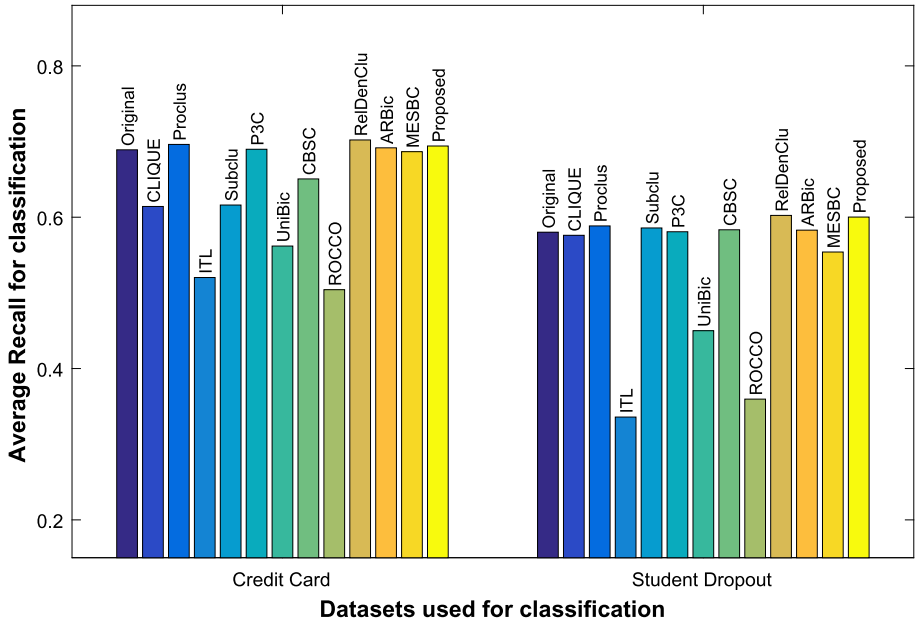
(d) Average precision obtained by various biclustering algorithms for Supervised Learning

Fig. 5. (*continued*)

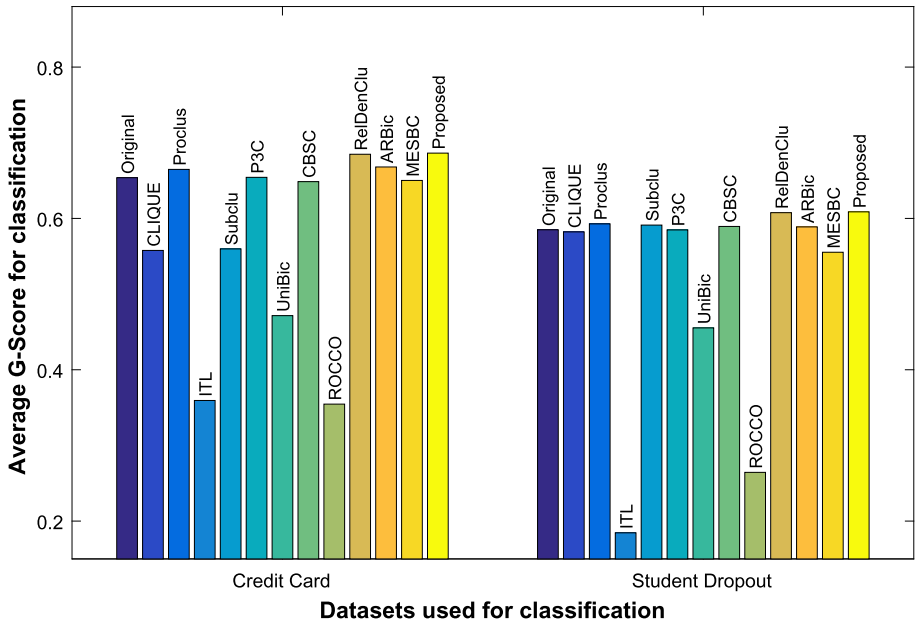
analysis. However, the values of the correlation coefficient at this time were small as the disease had spread to few countries and the relation between COVID-19 rates and other features was obscured.

For our analysis using the proposed method, we have utilized the WDI features extracted in the same way as described above and joined this dataset with the table of the cumulative number of confirmed COVID-19 cases on 31 January 2020 [63,64].

The proposed method does not find the correlation between features for the entire set of observations. Instead, it finds subsets of observations where features are related to each other by linear or non-linear relations. This allows the proposed method to find relations limited to a smaller subset of data. We may also say that the relation between the selected features for the selected observations is different from the relation between the same set of features for the observations outside the bicluster. Thus a bicluster that separates countries with a higher occurrence of COVID-19 from other countries would contain features that are more likely to



(e) Average recall obtained by various biclustering algorithms for Supervised Learning



(f) Average G-Score obtained by various biclustering algorithms for Supervised Learning

Fig. 5. (*continued*)

impact the COVID-19 infection rates. Note that traditional clustering methods cannot be used for this analysis, as they do not select features. Biclustering is useful in scenarios where subsets of observations as well as features are required together. Thus we applied the proposed method for analysis of lifestyle factors impacting COVID-19 rates. The details of the methodology and results are provided in the following sections.

6.3.1. Methodology for finding relevant features in COVID-19 dataset

To understand the features affecting the infection rate, we use the number of confirmed COVID-19 cases on 31 January 2020 i.e., the cumulative number of confirmed cases on that day. This along with WDI features, is used as input for PF-RelDenBi. We define the infection rate as the number of confirmed COVID-19 cases per million people living in a region. We find countries lying above the

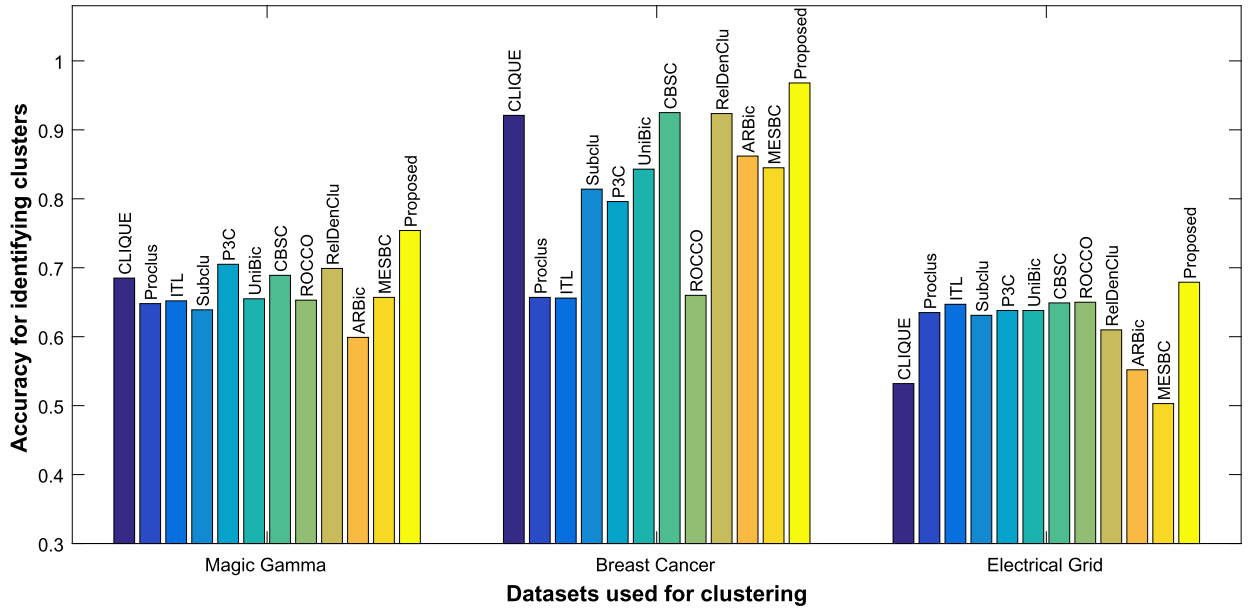
Table 10
Results for Unsupervised Learning.

Dataset	Magic Gamma [58]								
Method	Accuracy	NMI	ARI	Class 1			Class 2		
				Prec.	Recall	G-Score	Prec.	Recall	G-Score
Proposed	0.754	0.181	0.256	0.848	0.755	0.800	0.625	0.751	0.685
MESBC	0.657	0.023	0.068	0.690	0.857	0.769	0.522	0.289	0.388
ARBic	0.599	0.005	0.024	0.674	0.739	0.706	0.414	0.341	0.376
RelDenClu	0.699	0.096	0.138	0.629	0.500	0.536	0.770	0.807	0.779
ROCCO	0.653	0.006	0.009	0.845	0.016	0.117	0.652	0.998	0.807
CBSC	0.689	0.062	0.089	0.839	0.225	0.376	0.700	0.941	0.808
UniBic	0.655	0.006	0.015	0.654	0.991	0.805	0.677	0.035	0.153
P3C	0.705	0.080	0.110	0.690	0.992	0.827	0.919	0.177	0.404
Subclu	0.639	0.001	0.011	0.653	0.943	0.785	0.425	0.077	0.181
ITL	0.652	0.027	0.007	0.953	0.012	0.107	0.651	1.000	0.807
Proclus	0.648	0.013	0.048	0.676	0.878	0.770	0.498	0.223	0.333
CLIQUE	0.685	0.044	0.104	0.703	0.889	0.791	0.600	0.308	0.430
Dataset	Breast Cancer (Wisconsin Original) [59]								
Method	Accuracy	NMI	ARI	Class 1			Class 2		
				Prec.	Recall	G-Score	Prec.	Recall	G-Score
Proposed	0.968	0.784	0.874	0.986	0.964	0.975	0.936	0.975	0.955
MESBC	0.845	0.452	0.475	0.697	0.983	0.828	0.988	0.770	0.873
ARBic	0.862	0.389	0.520	0.887	0.903	0.895	0.814	0.787	0.800
RelDenClu	0.924	0.632	0.718	0.834	0.981	0.904	0.989	0.893	0.939
ROCCO	0.660	0.011	0.022	0.733	0.046	0.184	0.659	0.991	0.808
CBSC	0.925	0.607	0.719	0.866	0.932	0.898	0.963	0.920	0.941
UniBic	0.843	0.330	0.462	0.833	0.690	0.758	0.847	0.926	0.886
P3C	0.796	0.261	0.322	0.981	0.427	0.647	0.763	0.995	0.872
Subclu	0.814	0.296	0.370	0.975	0.481	0.685	0.781	0.993	0.880
ITL	0.656	0.007	0.011	0.833	0.021	0.132	0.654	0.998	0.808
Proclus	0.657	0.008	0.023	0.600	0.063	0.194	0.660	0.977	0.803
CLIQUE	0.921	0.579	0.706	0.881	0.895	0.888	0.943	0.935	0.939
Dataset	Electrical Grid [60]								
Method	Accuracy	NMI	ARI	Class 1			Class 2		
				Prec.	Recall	G-Score	Prec.	Recall	G-Score
Proposed	0.679	0.041	0.088	0.680	0.939	0.799	0.673	0.220	0.385
MESBC	0.503	0.000	0.000	0.363	0.497	0.425	0.639	0.506	0.569
ARBic	0.552	0.009	-0.022	0.255	0.124	0.177	0.615	0.795	0.699
RelDenClu	0.610	0.045	0.032	0.415	0.497	0.437	0.737	0.674	0.693
ROCCO	0.650	0.011	0.030	0.651	0.973	0.796	0.631	0.082	0.228
CBSC	0.649	0.055	0.092	0.503	0.487	0.491	0.724	0.741	0.731
UniBic	0.638	0.001	0.006	0.497	0.026	0.114	0.641	0.985	0.794
P3C	0.638	0.000	0.000	0.638	1.000	0.799	0.000	0.000	0.000
Subclu	0.631	0.000	0.001	0.383	0.030	0.107	0.639	0.973	0.788
ITL	0.647	0.009	0.029	0.572	0.096	0.235	0.652	0.959	0.791
Proclus	0.635	0.001	0.006	0.444	0.038	0.129	0.641	0.973	0.790
CLIQUE	0.532	0.000	0.002	0.372	0.425	0.398	0.645	0.593	0.619

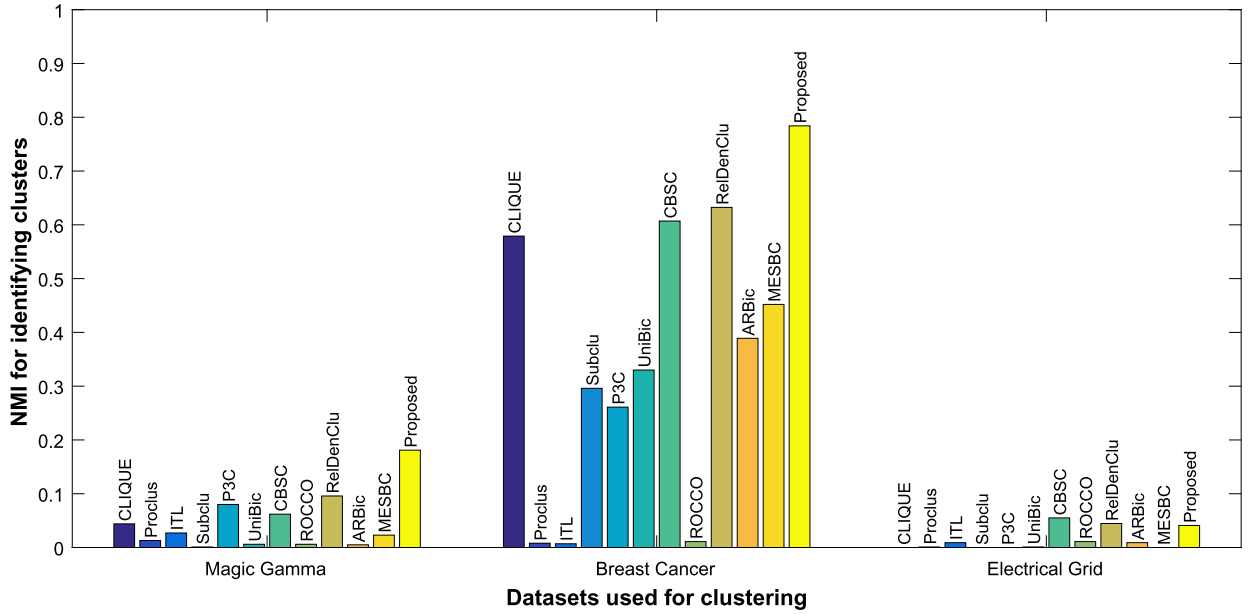
90 percentile in terms of infection rate. The bicluster having a maximum match for finding the set of countries with high COVID-19 rates is identified among all the biclusters obtained using PF-RelDenBi. The maximum match is found using the accuracy calculated using Equation (9). Note that this accuracy does not indicate the similarity between the bicluster and the group of countries with higher rates of COVID-19 infections, but indicates the ability of the bicluster to separate the countries with high and low COVID-19 infection rates. Since the biclusters are obtained based on the similarity between features for chosen observations, it can be inferred that the relationship between these features distinguishes the given set of observations from other observations. Thus these features are likely to affect the spread of the disease.

For this analysis, we have used the data available at an early stage on 31 January 2020. We have taken 90 percentile to cover the major portion of the dataset as the size of the dataset is very small. We have also repeated the analysis with the top 80 percentile countries and obtained the same results for all the methods. The major aim was to take information from a large part of the available data.

Using the data for January 2020, we found that the proposed method identified some factors related to the disease occurrence rate. At this time, the Spearman correlations of selected features with disease occurrence were low. However, the Spearman correlations of the features selected by PF-RelDenBi with disease occurrence on 31 December 2020 (after 1 year of the spread of the disease, before vaccination programs started in many countries) are high (Table 11), indicating that the selected features are indeed important.



(a) Accuracy of various biclustering algorithms for Unsupervised Learning (calculated using Equation 9)

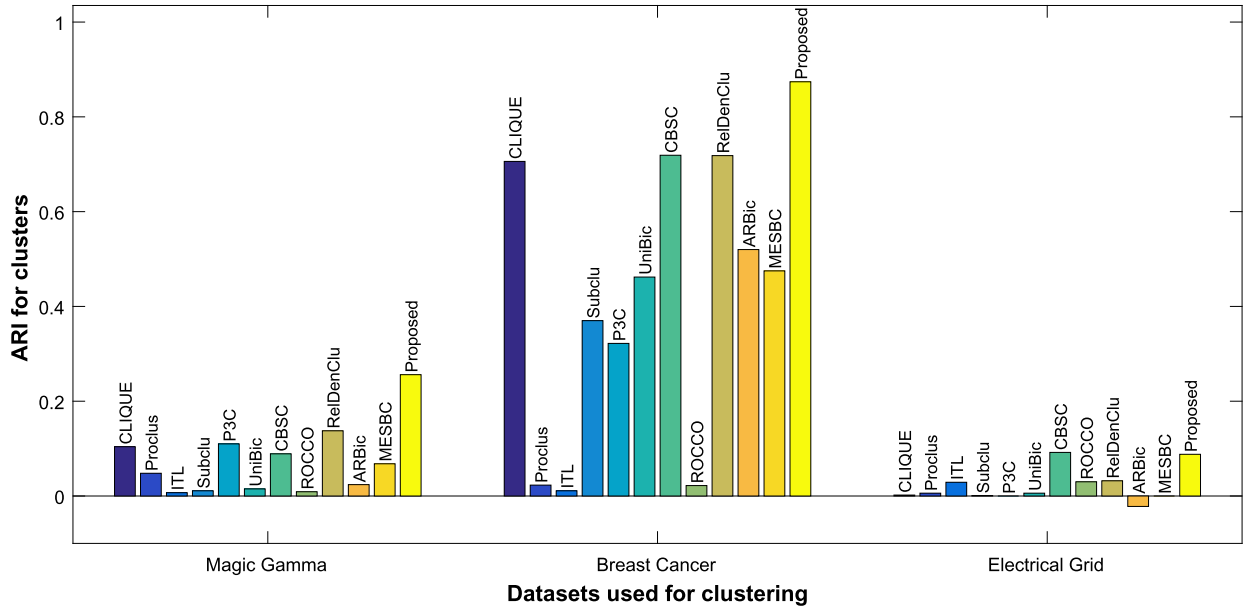


(b) NMI obtained by various biclustering algorithms for Unsupervised Learning

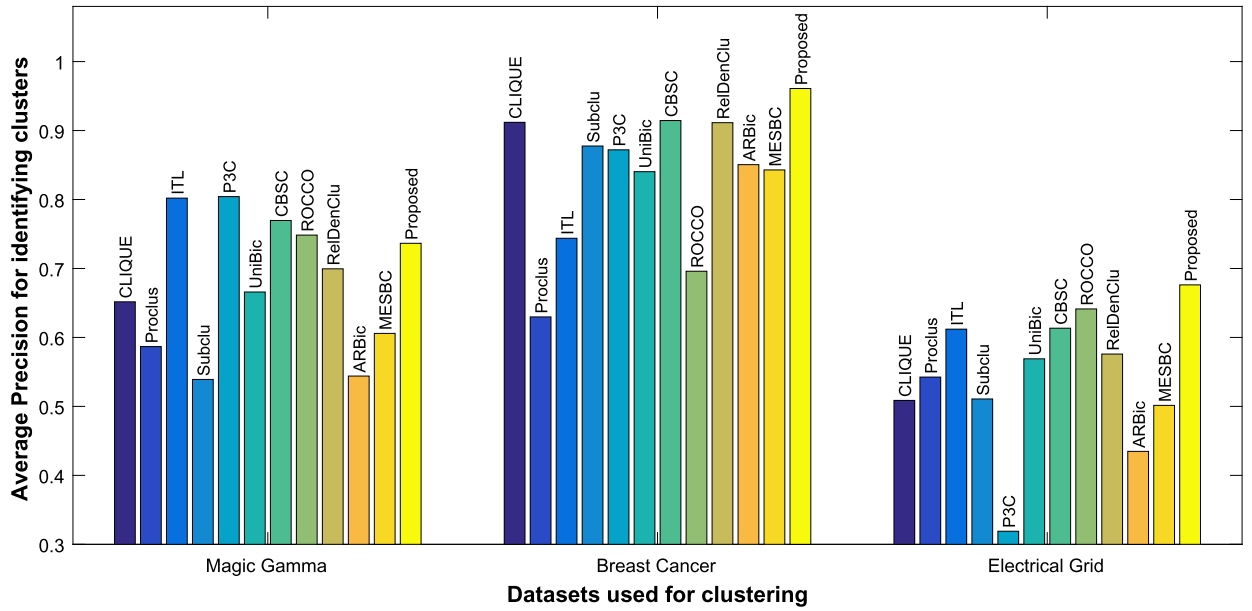
Fig. 6. Performance of various biclustering algorithms for Unsupervised Learning.

During the early phases of the disease, these relations were obscured by data from countries where the disease was still in its nascent stages. The proposed method is capable of mining this obscured information as it searches for relations between subsets of data. Early detection of factors impacting the spread of diseases makes the proposed method a useful tool for understanding new diseases and epidemics.

A similar analysis has also been done using CBSC, ROCCO, RelDenClu, ARBic and MESBC which are the most recent algorithms among the eleven methods mentioned above. For MESBC the parameter determining the number of classes is set to two. For ROCCO we take $k = 9$ as before. The parameters used for RelDenClu for this analysis are $MinSeedSize = 10$, $ObsInMinBase = 5$, $Sim2Seed = 0.95$, $ReuseAllSeeds = False$, $ReuseSeedSim = 0.5$, $ClusSim = 1$. The parameters that are used for CBSC are $MinCompts = 10$, $RatioMerge = 0.95$, $Reuse\_All\_Bases = False$, $Reuse\_Base\_Ratio = 0.5$, $Clus\_Overlap = 1$, $n_g = 10$ and $c = 0.85$. We have chosen



(c) ARI obtained by various biclustering algorithms for Unsupervised Learning



(d) Average precision obtained by various biclustering algorithms for Unsupervised Learning

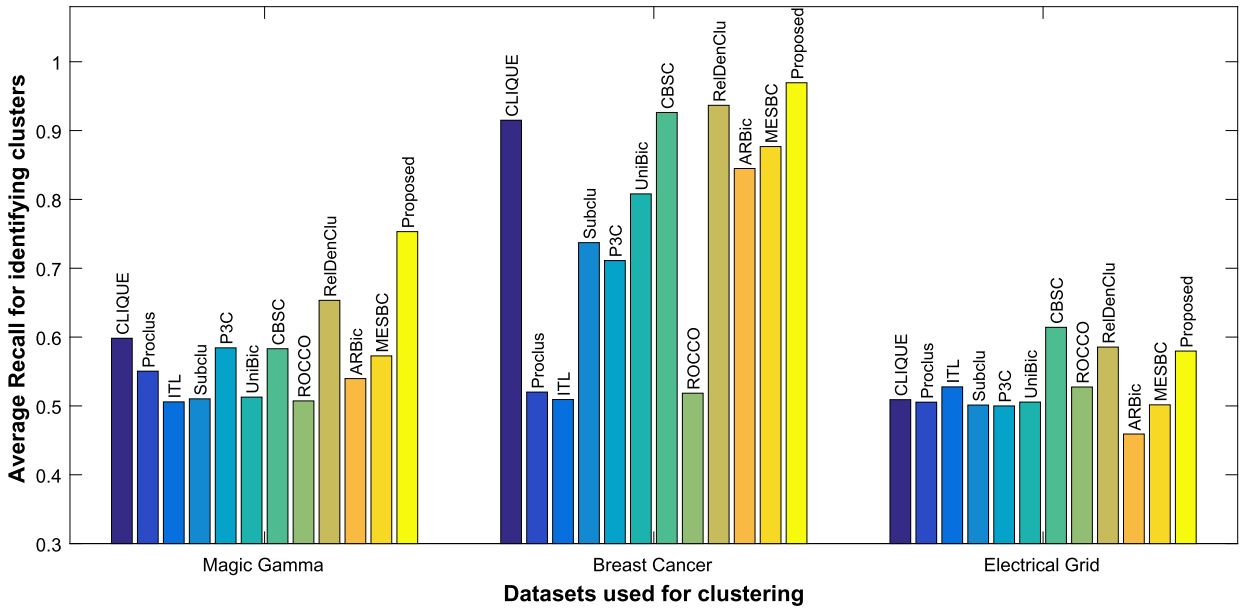
Fig. 6. (*continued*)

a smaller value of $MinSeedSize$ or $MinCompts$ to explore the relatively small dataset. A higher value is chosen for $Sim2Seed$ or $RatioMerge$ so that we choose features having higher similarity. For the method ARBic, default parameters are used.

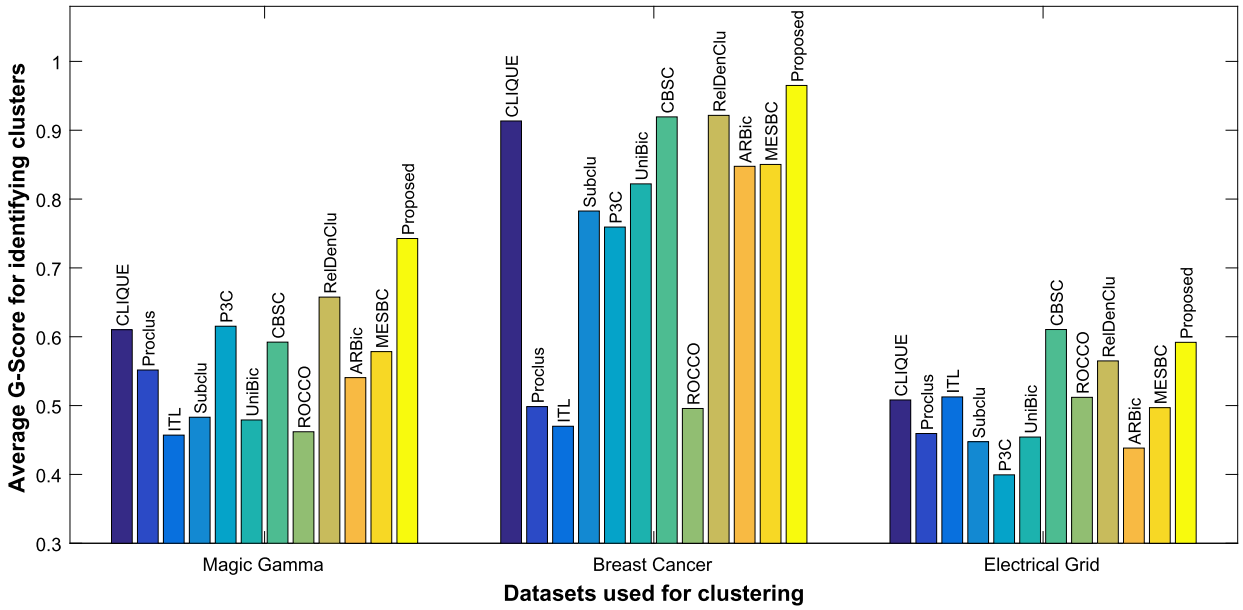
6.3.2. *Relevant features in COVID-19 dataset obtained using PF-RelDenBi during early stages of the pandemic*

Table 11 lists the 27 features used for analysis [61]. Features obtained by all the methods, using the data from January 2020 have been indicated by a tick in respective columns. The table also reports the values of Spearman correlation (ρ_s) between the chosen feature and the infection rate of COVID-19 for various countries for 31 January 2020 (denoted as ρ_s(Jan)) and 31 December 2020 (denoted as ρ_s(Dec)). The features are listed in decreasing order of absolute value of ρ_s(Dec).

The proposed method selected 13 out of 27 WDI features capable of distinguishing regions with high infection rates. We find that 11 out of 13 features selected by PF-RelDenBi have a Spearman correlation (absolute value) greater than 0.5 with the disease



(e) Average recall obtained by various biclustering algorithms for Unsupervised Learning



(f) Average G-Score obtained by various biclustering algorithms for Unsupervised Learning

Fig. 6. (*continued*)

occurrence rate on 31 December 2020. This indicates that PF-RelDenBi could identify the importance of these features much earlier i.e. from the January data when the Spearman correlation could not reflect this relation. Amongst the features not included in the bicluster, only 3 out of the 14 features have a Spearman correlation (absolute value) greater than 0.5 with disease occurrence rates on 31 December 2020.

The above-mentioned results suggest that the proposed algorithm could be explored to identify important factors impacting a new disease and, this, in turn, will aid in preparing a strategy, more scientifically, to combat the pandemic situation across countries. Indeed some of these selected features for COVID-19 have already been studied and are found to be important, as discussed in the following paragraphs, where we refer to features using their serial number (S.No.) given in Table 11.

Among the features selected by PF-RelDenBi, the features “Current health expenditure per capita” and “GDP per capita” (S.No. 1 and 8 in Table 11) may be related to the higher COVID-19 rate due to the higher number of tests. The percentage of deaths in

Table 11
Features (WDI dataset) selected by biclusters based on countries having high COVID-19 occurrence (90 percentile) in January 2020.

S.No.	Feature description	ρ_s(Jan)	ρ_s(Dec)	CBSC	ROCCO	RelDenClu	ARBic	MESBC	Proposed
1	Current health expenditure per capita, PPP (current international $)	0.22	0.69	-	-	-	-	-	✓
2	Cause of death, by communicable diseases and maternal, prenatal and nutrition conditions (% of total)	-0.09	-0.68	✓	-	-	-	✓	✓
3	Cause of death, by non-communicable diseases (% of total)	0.12	0.67	✓	-	-	-	-	✓
4	Mortality rate, adult, female (per 1,000 female adults)	-0.27	-0.67	✓	-	✓	-	✓	✓
5	Survival to age 65, female (% of cohort)	0.27	0.67	✓	-	✓	✓	-	✓
6	People using at least basic sanitation services (% of population)	0.23	0.64	✓	-	-	-	-	✓
7	Life expectancy at birth, total (years)	0.26	0.64	-	-	✓	✓	-	✓
8	GDP per capita, PPP (current international $)	0.22	0.62	✓	-	✓	✓	-	✓
9	Tuberculosis case detection rate (%, all forms)	0.13	0.61	-	-	-	-	-	-
10	Population ages 65 and above (% of total)	0.25	0.59	-	-	✓	✓	-	-
11	Survival to age 65, male (% of cohort)	0.24	0.58	✓	-	✓	-	-	✓
12	Urban population (% of total)	0.12	0.58	-	-	-	-	-	-
13	Mortality rate, adult, male (per 1,000 male adults)	-0.23	-0.57	✓	-	✓	-	✓	✓
14	Incidence of tuberculosis (per 100,000 people)	-0.05	-0.51	✓	-	✓	-	✓	✓
15	Population ages 15-64 (% of total)	0.07	0.46	-	-	✓	-	-	✓
16	International tourism, number of arrivals	0.4	0.44	-	-	✓	-	-	-
17	Mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 (%)	-0.23	-0.4	✓	-	-	-	✓	✓
18	PM2.5 air pollution, population exposed to levels exceeding WHO guideline value (% of total)	-0.23	-0.38	-	✓	-	✓	✓	-
19	Air transport, passengers carried	0.44	0.37	-	-	-	-	-	-
20	International migrant stock, total	0.35	0.27	-	-	-	-	-	-
21	Trade (% of GDP)	0.03	0.26	-	-	✓	-	-	-
22	Out-of-pocket expenditure (% of current health expenditure)	-0.09	-0.25	✓	-	-	-	✓	-
23	Labor force participation rate, total (% of total population ages 15+) (modelled ILO estimate)	0.04	-0.24	-	-	-	-	✓	-
24	Population density (people per sq. km of land area)	0.23	0.16	-	-	✓	-	-	-
25	Population, total	0.34	-0.16	-	-	✓	✓	✓	-
26	Tuberculosis treatment success rate (% of new cases)	-0.08	-0.15	-	-	✓	-	-	-
27	Death rate, crude (per 1,000 people)	0.02	0.07	-	-	-	-	-	-

region due to communicable and non-communicable diseases (S.No. 2 and 3 of Table 11) falls in line with the report given by Centers for Disease Control and Prevention [65]. Features showing life expectancy and mortality rate for the overall population and for each gender affect the age distribution of people in a region (S.No. 4,5,7,11,13,15,17 of Table 11) is selected by PF-RelDenBi. The feature “People using at least basic sanitation services” (S.No. 6 of Table 11) is related to the spread of communicable diseases and corresponding immunity. The feature “Incidence of tuberculosis” (S.No. 14 of Table 11), is interesting. It is also seen that countries with higher tuberculosis incidences have relativity lower COVID-19 infection rates. A study [66] has already noted that administering the BCG vaccine is likely to reduce the severity of COVID-19 symptoms. Another factor that may contribute to this association is that tuberculosis is more rampant in warm humid climates while COVID-19 is expected to spread faster in cold dry climates [67].

As mentioned earlier, all features selected by PF-RelDenBi, except two (S.No. 15 and 17 of Table 11) are found to have a high correlation with COVID infection rates on 31 Dec 2020, although the biclusters are obtained using the least amount of data from 31 Jan 2020.

Performance comparison between the proposed method, and other methods is shown in Fig. 7, which shows a scatter plot for Spearman correlation (absolute values). For each method, selected features are marked in blue and rejected features in red. For the proposed method the selected features are mostly the ones having a higher correlation. In fact, the minimum absolute value attained by any feature selected by PF-RelDenBi is 0.4, another selected feature attains a value of 0.46, while all other selected features lie above 0.5.

In a nutshell, most of the features selected by the proposed algorithm are relevant and the presented methodology allows us to explore some previously unknown associations. Once features are identified from larger datasets, a more detailed analysis could be done for individual features.

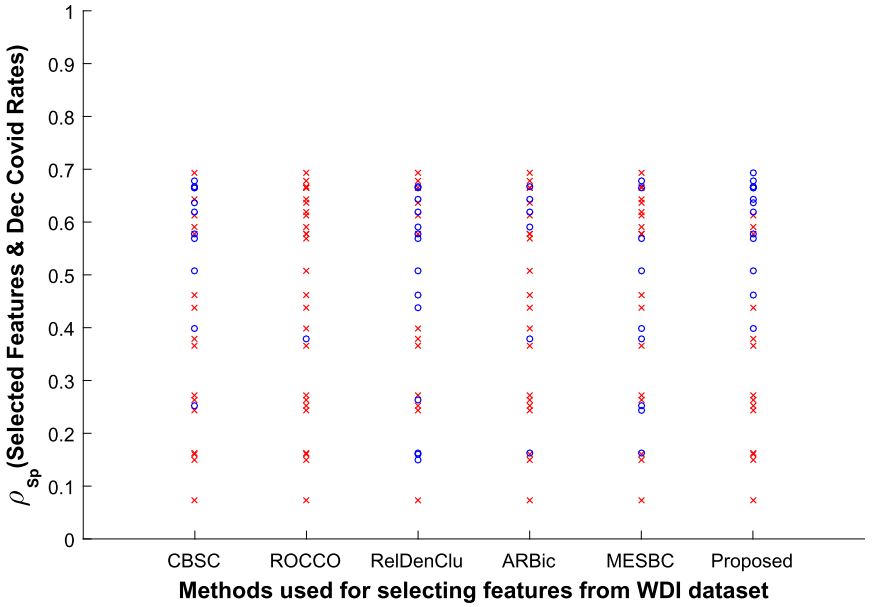


Fig. 7. Spearman correlation (absolute value) between Dec 31 2020 COVID infection rates and features selected from WDI database by biclustering methods. Selected features are marked in blue and rejected in red.

7. Conclusion and future work

In this article, we proposed a parameter-free algorithm that finds biclusters based on non-linear relations between features. By analysing local variations in marginal and joint densities, the algorithm is seen to perform well on non-linear datasets. The proposed algorithm does not require any user-defined parameter, making it a versatile method that can be used for analyzing datasets across varied domains. Experiments on simulated datasets have shown that the proposed algorithm is consistent under linear transforms. It is also seen to provide better performance on datasets obtained by adding noise. The performance of the proposed method is compared with eleven state-of-the-art methods for these simulated datasets. PF-RelDenBi is seen to provide better accuracy in most cases.

PF-RelDenBi is seen to improve the classification accuracy when used as a precursor for supervised learning for the Credit Card dataset and Student Dropout datasets. As an unsupervised learning method, it is able to find classes hidden in Magic Gamma, Breast Cancer and Electrical grid datasets with higher accuracy.

It has also been applied to a dataset containing information about the number of COVID-19 cases in different regions and respective development indicators, to obtain factors (demographic and others) impacting the number of confirmed infections in a region. Finally, the proposed algorithm facilitates us to understand the relationship between subsets of a dataset that is otherwise obscured by unrelated subsets of observations or overlooked due to non-linearity.

With the encouraging results obtained on several datasets, we would like to develop the proposed method further. Some possibilities for future work are discussed below.

It may be noted that the main objective of the proposed method is to find relations between features for a subset of observations. When used for clustering, it would work best for datasets where different clusters are defined by different relations between the features. However, in datasets where clusters are well separated only by spatial distances, the proposed method may not perform well. Therefore in future, we would like to experiment with ensemble methods utilizing the proposed method. We would also like to enhance the proposed algorithm so that it can utilize parallel computing allowing its application to larger datasets. Further, the proposed method either includes or excludes an observation or a feature in a bicluster; but does not provide a membership value to it. In future, we would like to work on assigning a membership value to observations and features using fuzzy sets.

CRediT authorship contribution statement

Namita Jain: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Susmita Ghosh:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Ashish Ghosh:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dr. Susmita Ghosh, one of the authors of the article ”A Parameter free Relative Density based Biclustering Method for

Table A.1
Functions used to generate datasets of type Non-Monotonous, Non-Linear 1 and Non-Linear 2. Both Non-Linear 1 and 2 datasets are similar but the range of some features is different.

h_i	Non-Monotonous	Non-Linear 1	Non-Linear 2
$h_1(x)$	$I(x) = x$	$I(x) = x$	$I(x) = x$
$h_2(x)$	$0.5\sin(2\pi x) + 0.5$	$\sin(x)$	$\sin(x)$
$h_3(x)$	$0.5\sin(3\pi x) + 0.5$	x^2	x^2
$h_4(x)$	$0.5\sin(15\pi x) + 0.5$	x^{10}	x^{10}
$h_5(x)$	$0.5\sin(25\pi x) + 0.5$	$\sin(\pi x)$	$0.5\sin(\pi x)$
$h_6(x)$	$2x$	$\sin(2\pi x)$	$0.5\sin(2\pi x) + 0.5$
$h_7(x)$	$\sin(2\pi x) + 1$	x^3	x^3
$h_8(x)$	$\sin(3\pi x) + 1$	$4x^2$	x^2
$h_9(x)$	$\sin(15\pi x) + 1$	$\sin(4\pi x)$	$0.5\sin(4\pi x) + 0.5$
$h_{10}(x)$	$\sin(25\pi x) + 1$	$4x^3$	x^3

identifying non-linear feature relations" is an associate editor of Heliyon. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

The real-world datasets used in this article are available in public repositories and have been appropriately cited (UCI ML repository [52], The World Bank [61], COVID-19 Dashboard [63] and Our World in data [64]). Simulated data is used to demonstrate the properties of the proposed method. The procedure for generating simulated data is provided in the appendix (supplementary material) in detail. The code for generating the data, as well as the simulated datasets used for experiments, have been uploaded to the GitHub repository as mentioned below in the code availability statement.

Code availability statement

The MATLAB code for the algorithm presented in this article along with the code to create simulated data has been uploaded to a GitHub repository and can be found at https://github.com/namitaML/RelDenClu-Non-linear-feature-relation-based-biclustering.

Acknowledgement

We would like to acknowledge that Prof. C.A. Murthy has made a major contribution to the development of the ideas and methodology presented in this manuscript. Unfortunately, he is not with us anymore. We have deep gratitude for the contribution made by him.

Appendix A. Detailed procedure for generating simulated datasets

This section outlines our method of generating synthetic datasets for analyzing the properties of the proposed algorithm. The datasets for Table 1 require several functions which are reported in Table A.1. The rest of the data is generated using a single function.

In Table 1, the first column shows the results on datasets generated in the following manner. A random matrix of size 1000×20 is generated. Now we replace 500×10 submatrix of data in a way so that the i^{th} column of this submatrix is given by $h_i(x)$, where x is the value in the first column and the definitions of h_i for $i = 1, 2, \cdots, 10$ are given in Table A.1. Thus we embed a bicluster of size 500×10 in the data.

Similarly, the second column shows the results for data generated using functions in the second column of Table A.1. Note that for some columns, the range in which the bicluster elements lie is different from the range of remaining data. Thus the background for the biclusters is highly variable.

Datasets used for results shown in the third column of Table 1 are generated using a set of functions, which are modifications of the functions used in the second column. The modifications are made so that data in bicluster lies in the same range as the rest of the data. The modified functions have been reported in the third column of Table A.1. It may be noted that the function x^2 occurs both as h_3 and h_8 in the third column and x^3 occurs both as h_7 and h_{10}, amounting to the repetition of a column in the bicluster. However, we have retained the repeated columns to retain the similar structure of bicluster as in datasets for the first column.

The first column of Table 2 presents the results on datasets of size 1000×20 drawn from the uniform distribution. It has a bicluster of size 500×10 generated using functions $h_1(x) = I(x) = x$, where I is the identity function and $h_i(x) = a_i * x$ with $i = 2, 3 \cdots, 10$. a_i with $i = 2, 3, \cdots, 10$ are different random values lying in interval $(0, 1)$. This is our base data for Uniform distribution datasets. Different transformations have been applied to this data for analysing properties of biclustering algorithms and corresponding results have been reported in the following columns.

The second column of Table 2 contains results for datasets obtained by scaling each column of data generated for the first column of Table 2 with a random number lying in the interval $(0, 1)$.

The third column of Table 2 contains results for datasets obtained by adding a random number lying in $(0, 1)$ to each column of the data generated for the 2. These are used to analyze the scaling and translation properties of the algorithms.

The fourth column of Table 2 contains results for datasets obtained by linear transform to each column of data generated for the 2. In a way, this is a combination of scaling and translation. Two random numbers r_1 and r_2 are generated for each column and the transform $r_1 x + r_2$ is applied. Scaling, translation and linear transforms are special cases of distance-preserving transforms.

The fifth column of Table 2 contains results for datasets obtained by duplicating each observation of datasets used for reporting the results in the 2. Thus these datasets contain 2000 observations.

The sixth column of Table 2 contains results for datasets obtained by duplicating each observation of bicluster in datasets generated for the 2. Thus these datasets contain 1500 observations.

The seventh column of Table 2 contains results for datasets obtained by adding a random number in the range (0,0.1) to each element of data matrices. The random noise is drawn from a uniform distribution. The robustness of the algorithms regarding noise is checked through this.

The eighth column of Table 2 contains results for datasets obtained by randomly shuffling rows and columns of the data generated for the 2.

The first column of Table 3 presents the results on datasets of size 1000×20 drawn from Gaussian distribution. The bicluster submatrix of size 500×10 is generated in the same way as the first column of Table 2.

The second column of Table 3 contains results for datasets obtained by adding noise to each element of the data matrix generated in the first column. Noise is generated using random numbers drawn from Gaussian distribution with a standard deviation 0.1. Thus we test the robustness of the algorithms for Gaussian data.

The third column of Table 3 contains results for datasets obtained by adding noise to each element of the data matrix generated in the first column. Noise is generated using random numbers drawn from Gaussian distribution with a standard deviation 0.2. Thus we test the robustness of the algorithms for Gaussian data.

The first and second columns of Table 4 contain results for datasets containing two overlapping clusters. The data is uniformly distributed and each row of the biclusters has a constant value. The first column shows accuracy for a bicluster with 500 rows and 10 columns. The second column shows accuracy for another bicluster with 400 rows and 10 columns. There is an overlap of 200 rows and 3 features in both biclusters (12% of the larger bicluster).

The third and fourth columns of Table 4 contain results for datasets containing two overlapping clusters. The data is uniformly distributed and each row of the biclusters has a constant value. The first column shows accuracy for a bicluster with 500 rows and 10 columns. The second column shows accuracy for another bicluster with 400 rows and 10 columns. There is an overlap of 200 rows and 5 features in both biclusters (20% of the larger bicluster).

References

[1] R. Vidal, Subspace clustering, IEEE Signal Process. Mag. 28 (2011) 52–68.
[2] H.A. Ahmed, P. Mahanta, D.K. Bhattacharyya, J.K. Kalita, Shifting-and-scaling correlation based biclustering algorithm, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (2014) 1239–1252.
[3] L. Cheung, K.Y. Yip, D.W. Cheung, B. Kao, M.K. Ng, On mining micro-array data by order-preserving submatrix, in: 21st International Conference on Data Engineering Workshops (ICDEW'05), 2005, p. 1153.
[4] Z. Wang, G. Li, R.W. Robinson, X. Huang, UniBic: sequential row-based biclustering algorithm for analysis of gene expression data, Sci. Rep. 6 (2016).
[5] N. Jain, C.A. Murthy, Connectedness-based subspace clustering, Knowl. Inf. Syst. 58 (2019) 9–34.
[6] N. Jain, C.A. Murthy, A new estimate of mutual information based measure of dependence between two variables: properties and fast implementation, Int. J. Mach. Learn. Cybern. 7 (2016) 857–875.
[7] J.A. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. 67 (1972) 123–129.
[8] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE/ACM Trans. Comput. Biol. Bioinform. 1 (2004) 24–45.
[9] P. Carmona-Saez, R.D. Pascual-Marqui, F. Tirado, J.M. Carazo, A. Pascual-Montano, Biclustering of gene expression data by non-smooth non-negative matrix factorization, BMC Bioinform. 7 (2006) 78.
[10] Q. Huang, D. Tao, X. Li, L. Jin, G. Wei, Exploiting local coherent patterns for unsupervised feature ranking, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 41 (2011) 1471–1482.
[11] Q. Huang, J. Yang, X. Feng, A.W.-C. Liew, X. Li, Automated trading point forecasting based on bicluster mining and fuzzy inference, IEEE Trans. Fuzzy Syst. 28 (2020) 259–272.
[12] Q. Huang, T. Wang, D. Tao, X. Li, Biclustering learning of trading rules, IEEE Trans. Cybern. 45 (2015) 2287–2298.
[13] U. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.
[14] Q. Huang, X. Huang, Z. Kong, X. Li, D. Tao, Bi-phase evolutionary searching for biclusters in gene expression data, IEEE Trans. Evol. Comput. 23 (2019) 803–814.
[15] A. José-García, J. Handl, W. Gómez-Flores, M. Garza-Fabre, An evolutionary many-objective approach to multiview clustering using feature and relational data, Appl. Soft Comput. 108 (2021) 107425.
[16] O. Maatouk, E. Ayari, H. Bouziri, W. Ayadi, Bobea: a bi-objective biclustering evolutionary algorithm for genome-wide association analysis, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2022, pp. 344–347.
[17] P. Swathypriyadharsini, K. Premalatha, Hybrid cuckoo search with clonal selection for triclustering gene expression data of breast cancer, IETE J. Res. 69 (2023) 2328–2336.
[18] G. Getz, E. Levine, E. Domany, Coupled two-way clustering analysis of gene microarray data, Proc. Natl. Acad. Sci. 97 (2000) 12079–12084.
[19] A. Tanay, R. Sharan, R. Shamir, Discovering statistically significant biclusters in gene expression data, Bioinformatics 18 (2002) S136–S144.
[20] X. Peng, J. Feng, J.T. Zhou, Y. Lei, S. Yan, Deep subspace clustering, IEEE Trans. Neural Netw. Learn. Syst. 31 (2020) 5509–5521.
[21] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Trans. Knowl. Discov. Data 3 (2009) 1–58.
[22] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, E. Zitzler, A systematic comparison and evaluation of biclustering methods for gene expression data, Bioinformatics 22 (2006) 1122–1129.
[23] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM SIGKDD Explor. Newsl. 6 (2004) 90–105.

[24] K. Kailing, H.-P. Kriegel, P. Kröger, Density-connected subspace clustering for high-dimensional data, in: Proc. SIAM International Conference on Data Mining (SDM'04), vol. 4, 2004, pp. 246–256.
[25] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, SIGMOD Rec. 27 (1998) 94–105.
[26] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, A weighted adaptive mean shift clustering algorithm, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, 2014, pp. 794–802.
[27] A.K.H. Tung, X. Xu, B.C. Ooi, Curler: finding and visualizing nonlinear correlation clusters, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD '05, ACM, New York, NY, USA, 2005, pp. 467–478.
[28] Y. Cheng, G.M. Church, Biclustering of expression data, in: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 2000, pp. 93–103.
[29] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. Van Sanden, D. Lin, W. Talloen, L. Bijnens, H.W.H. Göhlmann, Z. Shkedy, D.-A. Clevert, FABIA: factor analysis for bicluster acquisition, Bioinformatics 26 (2010) 1520–1527.
[30] X. Liu, T. Yu, X. Zhao, C. Long, R. Han, Z. Su, G. Li, ARBic: an all-round biclustering algorithm for analyzing gene expression data, NAR Genomics Bioinform. 5 (2023).
[31] G. Li, Q. Ma, H. Tang, A.H. Paterson, Y. Xu, Qubic: a qualitative biclustering algorithm for analyses of gene expression data, Nucleic Acids Res. 37 (2009), Page e101.
[32] J. Xie, A. Ma, Y. Zhang, B. Liu, S. Cao, C. Wang, J. Xu, C. Zhang, Q. Ma, QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data, Bioinformatics 36 (2019) 1143–1149.
[33] N. Jain, S. Ghosh, C.A. Murthy, Reldenclu: a relative density based biclustering method for identifying non-linear feature relations, arXiv preprint, arXiv: 1811.04661, 2021.
[34] N. Jain, S. Ghosh, Machine learning method to discover the novel association of vaccine and vitamin a supplement on covid 19 infection: a biclustering approach, in: 2021 IEEE 18th India Council International Conference (INDICON), 2021, pp. 1–6.
[35] X. He, L. Moreira-Matias, Robust continuous co-clustering, arXiv preprint, arXiv:1802.05036, 2018.
[36] F. Alqadah, C. Reddy, J. Hu, H. Alqadah, Biclustering neighborhood-based collaborative filtering method for top-n recommender systems, Knowl. Inf. Syst. 44 (2014).
[37] D. Ienco, R.G. Pensa, R. Meo, Parameter-free hierarchical co-clustering by n-ary splits, in: W. Buntine, M. Grobelnik, D. Mladenić, J. Shawe-Taylor (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, 2009, pp. 580–595.
[38] F. Liu, Y. Yang, X.S. Xu, M. Yuan, Mesbc: a novel mutually exclusive spectral biclustering method for cancer subtyping, Comput. Biol. Chem. 109 (2024) 108009.
[39] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (1962) 1065–1076.
[40] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science 16 (2011) 1518–1524.
[41] T.O. Kvålseth, On normalized mutual information: measure derivations and properties, Entropy 19 (2017).
[42] Y. Horibe, Entropy and correlation, IEEE Trans. Syst. Man Cybern. (1985) 641–642.
[43] A. Dey, S. Ghosh, A. Ghosh, Band elimination for dimensionality reduction of hyperspectral images using mutual information, in: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 2055–2058.
[44] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, J.S. Park, Fast algorithms for projected clustering, SIGMOD Rec. 28 (1999) 61–72.
[45] I.S. Dhillon, S. Mallela, D.S. Modha, Information-theoretic co-clustering, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 89–98.
[46] G. Moise, J. Sander, M. Ester, Robust projected clustering, Knowl. Inf. Syst. 14 (2008) 273–298.
[47] P. Orzechowski, A. Pańszczyk, X. Huang, J.H. Moore, runibic: a bioconductor package for parallel row-based biclustering of gene expression data, Bioinformatics (2018) bty512.
[48] M. Hassani, M. Hansen, E. Müller, I. Assent, S. Günnemann, T. Jansen, T. Seidl, Subspace: interface to opensubspace, University of Waikato, 2015, https://cran.r-project.org/web/packages/subspace/index.html.
[49] J.K. Gupta, S. Singh, N.K. Verma, MTBA: MATLAB toolbox for biclustering analysis, in: IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, IIT Kanpur, India, IEEE, 2013, pp. 94–97.
[50] L. Fisher, J.W.V. Ness, Admissible clustering procedures, Biometrika 58 (1971) 91–104.
[51] C. Xu, W. Xu, K. Jing, Fast algorithms for singular value decomposition and the inverse of nearly low-rank matrices, Nat. Sci. Rev. 10 (2023) nwad083.
[52] D. Dua, C. Graff, UCI machine learning repository, 2017.
[53] I.-C. Yeh, Default of credit card clients, UCI Machine Learning Repository, 2016, https://doi.org/10.24432/C55S3H, 2016.
[54] V. Realinho, M.V. Martins, J. Machado, L. Baptista, Predict Students' Dropout and Academic Success, UCI Machine Learning Repository, 2021, https://doi.org/10.24432/C5MC89.
[55] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[56] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2006) 861–874.
[57] J.E. Chacón, A.I. Rastrojo, Minimum adjusted rand index for two clusterings of a given size, Adv. Data Anal. Classif. 17 (2022) 125–133.
[58] R. Bock, MAGIC Gamma Telescope, UCI Machine Learning Repository, 2007, https://doi.org/10.24432/C52C8B.
[59] W. Wolberg, Breast Cancer Wisconsin (Original), UCI Machine Learning Repository, 1992, https://doi.org/10.24432/C5HP4Z.
[60] V. Arzamasov, Electrical Grid Stability Simulated Data, UCI Machine Learning Repository, 2018, https://doi.org/10.24432/C5PG66.
[61] The World Bank, World development indicators, https://datacatalog.worldbank.org/dataset/world-development-indicators, 2015.
[62] Sambit Mukherjee, Covid-19 data: adding world development indicators, https://www.kaggle.com/code/sambitmukherjee/covid-19-data-adding-world-development-indicators/report, 2020.
[63] E. Dong, H. Du, L. Gardner, An interactive web-based dashboard to track covid-19 in real time, Lancet Infect. Dis. 20 (2020) 533–534.
[64] E. Mathieu, H. Ritchie, L. Rodés-Guirao, C. Appel, C. Giattino, J. Hasell, B. Macdonald, S. Dattani, D. Beltekian, E. Ortiz-Ospina, M. Roser, Coronavirus pandemic (covid-19), Our World in Data, 2020, https://ourworldindata.org/coronavirus.
[65] Centers for Disease Control and Prevention, People who are at higher risk for severe illness, cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-at-higher-risk.html, 2020.
[66] N. Curtis, A. Sparrow, T.A. Ghebreyesus, M.G. Netea, Considering BCG vaccination to reduce the impact of COVID-19, Lancet 395 (2020) 1545–1546.
[67] New Scientist, Will the spread of covid-19 be affected by changing seasons?, https://www.newscientist.com/article/2239380-will-the-spread-of-covid-19-be-affected-by-changing-seasons/, 2020.