Mach Learn (2017) 106:951–992
DOI 10.1007/s10994-016-5623-3

Feature-weighted clustering with inner product induced norm based dissimilarity measures: an optimization perspective

Arkajyoti Saha[1] · **Swagatam Das**[2]

Received: 1 October 2015 / Accepted: 16 December 2016 / Published online: 15 February 2017

Abstract The performance of a clustering algorithm can be improved by assigning appropriate weights to different features of the data. Such feature weighting is likely to reduce the effect of noise and irrelevant features while enhancing the effect of the discriminative features simultaneously. For the clustering purpose, feature-weighted dissimilarity measures are so far limited to Euclidean, Mahalanobis, and exponential distances. In this article, we introduce a novel feature weighting scheme for the general class of inner product induced norm (IPIN) based weighted dissimilarity measures. This class has a wide range and includes the three above-mentioned distances as special cases. We develop the general algorithms to solve the hard (k-means) and fuzzy (fuzzy c-means) partitional clustering problems and undertake in-depth analyses of the convergence of the algorithms as well. In addition, we address issues like feasibility and uniqueness of the solutions of these problems in sufficient details. The novelty of the article lies in the introduction of a general feature weighting scheme for the generalized class of IPIN-based dissimilarity measures and a complete convergence analysis for the automated feature-weighted clustering algorithms using such measures.

Keywords Dissimilarity measures · Inner product induced norm · Hard and fuzzy clustering · Generalized feature weighting · Feature selection · Convergence analysis

Editor: Eyke Hüllermeier.

✉ Swagatam Das
swagatam.das@isical.ac.in

Arkajyoti Saha
arkajyotisaha93@gmail.com

[1] Stat-Math Unit, Indian Statistical Institute, Kolkata, India

[2] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India

1 Introduction

Clustering is the method of finding meaningful groups within a collection of patterns or data points based on some prefixed similarity/dissimilarity measures. Although there does not exist a universally accepted criterion for such grouping, most often it is done in such a manner that the patterns in the same group (called a cluster) are as homogenous as possible, while patterns from different clusters share maximum heterogeneity. Clustering plays a pivotal role in efficient data analysis procedure by extracting essential and granular information from raw datasets. Unlike supervised learning or discriminant analysis, it involves no labeled data or any *training set*. Clustering techniques find huge applications in diverse areas of science and technology like financial data analysis, web mining, spatial data processing, land cover study, medical image understanding, social network analysis etc. (Anderberg 2014).

In a broad sense, data clustering algorithms can be of two types—hierarchical or partitional (Jain et al. 1999). Hierarchical clustering seeks to build a hierarchy of clusters (using a tree-like structure, called the dendrogram) following the agglomerative or the divisive approach. Partitional clustering algorithms, on the other hand, attempt to partition the dataset directly into a given number of clusters, where each cluster is characterized by a vector prototype or cluster representative. These algorithms try to optimize certain objective function involving the data points and the cluster representative (e.g., a squared-error function based on the intra-cluster spread). These algorithms typically come in two variations. One is hard clustering, where we assign each pattern to a single cluster only. The other variation is fuzzy clustering, where each pattern can belong to all the clusters with a certain membership degree (in [0, 1]) for each of them.

In order to detect the extent of homogeneity and/or heterogeneity between two data points, a dissimilarity measure is required. Dissimilarity measures are a generalized version of the distance functions in the sense that the former need not obey the triangle inequality. However, the terms dissimilarity and distance are used almost synonymously in the literature. For sound clustering, it is important to choose a dissimilarity measure, which is able to explore the inherent structure of the input data. It is also equally important to put different degrees of emphasis on different features (or variables describing each data point) while computing the dissimilarity, since it is a well-known fact that all features do not have equal contribution to the task of data clustering, and some even bear a negative influence on the task. Feature weighting is a technique for approximating the optimal degree of influence of the individual features. Each feature may be considered to have a relative degree of importance called the feature weight, with value lying in the interval [0, 1]. The corresponding algorithm should have a learning mechanism to adapt the weights in such a manner that a noisy or derogatory feature should finally have a very small weight value, thus contributing insignificantly to the dissimilarity computation. If the weight values are confined to either 0 or 1 (i.e. *bad* features are completely eliminated), then feature weighting reduces to the process of feature selection. This can be done very simply by thresholding the numerical weights. Feature selection can significantly reduce the computational complexity of the learning process with negligible loss of vital information.

Due to the use of Euclidean distance, both k-means and FCM work best when the clusters present in the data are nearly hyper-spherical in shape. FCM was formulated with other dissimilarity measures like the Mahalanobis distance (Gustafson and Kessel 1978; Gath and Geva 1989), exponential distance (D'Urso et al. 2016) etc. The same is true for the k-means algorithm, see for example works like (Hung et al. 2011; Mao and Jain 1996). The general class of inner product induced norm based Consistent Dissimilarity (IPINCD)

measures (Saha and Das 2016b) has a wide range and it includes Euclidean, Mahalanobis, and exponential distances as special cases. Several other dissimilarity measures, which can adapt themselves by estimating some kind of covariance measure of the clusters, can be derived by using the definition of this class. In a recent work (Saha and Das 2016b), we have discussed the convergence properties of the (hard) k-means and Fuzzy C-Means (FCM) clustering algorithms with the un-weighted IPINCD measures. In this article, we take one step forward and introduce the concept of a generalized feature weighting scheme for the IPINCD measures. We then develop the generalized k-means and FCM clustering algorithms with automated feature weight learning and present a complete convergence analysis of the algorithms, thus derived. Below we briefly summarize the main contributions of the paper as follows:

1. We introduce a general feature weighting scheme for clustering methods with the generalized IPIN-based dissimilarity measures (IPINCD). We treat the feature weights as optimizable parameters of the clustering problem. We develop a Lloyd heuristic and an Alternative Optimization (AO) algorithm to solve the automated feature-weighted k-means and FCM clustering problems respectively.
2. We perform an in-depth analysis of the characteristics of the optimization problems in the newly developed algorithms. We address the issues of existence and uniqueness of solutions of the sub-optimization problems, that form the basic structure of the general clustering algorithms.
3. We theoretically prove the convergence properties of the newly developed feature-weighted k-means and FCM algorithms to a stationary point. We explore the nature of the stationary points under all possible situations.
4. With a special choice of the proper dissimilarity measure, an automated weighted version of the classical fuzzy covariance matrix based clustering algorithm (Gustafson and Kessel 1978) is derived as a special case of the proposed general clustering algorithm.

Organization of the paper is in order. Section 2 provides a brief overview of the background of clustering, use of dissimilarity measures for clustering and feature weighting in the perspective of unsupervised learning. In Sect. 3, we define a general class of the IPIN-based weighted dissimilarity measures along with a novel generalized feature weighting scheme. We also present a Lloyd type and an AO algorithm to solve the weighted k-means and the FCM clustering problems respectively. Mathematical analysis of the convergence properties of the algorithms, as well as the characteristics of the underlying optimization problems, are provided in Sect. 4. In Sect. 5, the relationship of the proposed algorithm with the existing feature weighting schemes and clustering algorithms are discussed. Section 6 presents illustrative experimental results to highlight the effectiveness of the proposed feature-weighted dissimilarity measures. In Sect. 7, we present a theoretical discussion on the asymptotic runtime of the proposed algorithm. Finally, Sect. 8 concludes the proceedings and unearths a few interesting future avenues of research.

2 Background

2.1 Partitional clustering

Partitional clustering algorithms learn to divide a dataset directly into a predefined number of clusters. They either move the data points iteratively between possible subsets or try to detect areas of high concentration of data as clusters (Berkhin 2006). This paper addresses

two very popular partitional algorithms of the first kind. These are the k-means algorithm for hard clustering and FCM for fuzzy clustering. Both the algorithms, in their classical forms, attempt to fit the data points most appropriately into their clusters and are likely to yield convex clusters.

2.1.1 k-means clustering algorithm

The k-means algorithm iteratively assigns each data point to the cluster, whose representative point or centroid is nearest to the data point with respect to some distance function. An optimal k-means clustering can be identified with a Voronoi diagram whose seeds are the centrality measures of the elements of the cluster. The classical k-means algorithm (MacQueen et al. 1967) minimizes the intra-cluster spread (measured by considering the squared Euclidean distance as a dissimilarity measure) by using some heuristic to converge quickly to a local optimum. Lloyd's heuristic (Lloyd 1982) is a popular choice among the practitioners for optimizing the k-means objective function and recently a performance guarantee of this method for well-clusterable situations has been presented (Ostrovsky et al. 2012). There have been several attempts (Modha and Spangler 2003; Teboulle et al. 2006) to extend the conventional hard k-means algorithm by considering objective functions involving dissimilarity measures other than the usual squared Euclidean distance. The general Bregman divergence (Banerjee et al. 2005) unified the set of all divergence measures, for which using arithmetic mean as cluster representative guarantees a progressive decrease in the objective function with iterations.

2.1.2 Fuzzy clustering algorithms

In fuzzy clustering, each cluster is treated as a fuzzy set and all data points can belong to it with a varying degree of membership. Perhaps, the most popular algorithm in this area is the fuzzy ISODATA or Fuzzy C-means (FCM) (Dunn 1973) and its generalized version (Bezdek 1981). In FCM, each data point is assigned with a membership degree to each cluster (quantifying how truly the data point belongs to that cluster). The numerical membership degree is inversely related to the relative distance of that data point from the corresponding cluster representative. FCM uses Euclidean distance as the dissimilarity measure and an AO heuristic (Bezdek and Hathaway 2003) to locally minimize a cost function involving the cluster representatives and the membership values. Since the Euclidean distance has a bias towards forming hyper-spherical clusters, FCM has undergone significant changes in terms of the dissimilarity measures used. Gustafson and Kessel (1978) modified FCM by using the Mahalanobis distance resulting into the well-known GK (Gustafson Kessel) algorithm, which can capture hyper-ellipsoidal clusters of equal volume (Krishnapuram and Kim 1999) by estimating a fuzzy cluster covariance matrix. This algorithm has found several applications in pattern recognition and computer vision and is still a subject of active research (Chaomurilige et al. 2015). Gath and Geva (1989) extended the GK algorithm by considering the size and density of the clusters while computing the distance function.

A series of modification for the Mahalanobis distance-based clustering algorithms were proposed by adding restrictions to the covariance matrix (Liu et al. 2007a, b) or replacing the cluster specific covariance matrix by a single common covariance matrix (Liu et al. 2009a, b). Wu et al. (2012) showed that any distance function that preserves the local convergence of FCM (when the cluster representative is derived as an arithmetic mean) can be obtained from a class of continuously differentiable convex functions, called Point-to-Centroid Distance

(P2C-D) by the authors. This class comprises of the Bregman divergence and a few other divergences. Teboulle (2007) presented a generic algorithm for the center-based soft and hard clustering methods with a broad class of the *distance like functions*. Recently Saha and Das (2016a) designed an FCM algorithm with the separable geometric distance and demonstrated its robustness towards the noise feature perturbations (Saha and Das 2016a).

2.2 Feature weighting

Representing data with a minimal number of truly discriminative features is a fundamental challenge in machine learning and it greatly alleviates the computational overhead of the learning process. We can project the data to a lower dimensional space by selecting only the relevant features from the entire set of features available (feature selection) or by generating a new set of features using a combination of all the existing ones. Feature weighting may be seen as a generalization of the feature selection process. Here the relative degree of importance of each feature is quantized as the feature weight with value lying in the interval [0, 1]. Preliminary approaches of feature weighting for clustering can be found in the works like Sneath et al. (1973) and Lumelsky (1982). In the SYNCLUS (SYNthesized CLUStering) (DeSarbo et al. 1984) algorithm, first a k-means algorithm partitions the data and then a group of new weights for various features is determined by optimizing a weighted mean-squared cost function. The algorithm executes these two steps iteratively until convergence to a set of optimal weights is achieved. De Soete (1988) proposed a feature weighting method for hierarchical clustering by using two objective functions to determine the weights for trees in ultrametric and additive forms. Makarenkov and Legendre (2001) adapted De Soete's algorithm for k-means clustering and they reduced the computation time by using the Polak–Ribiere optimization procedure to minimize the objective function involving the weights.

A few well-known fuzzy clustering algorithms have also been modified to accommodate the feature weighting strategy. Keller and Klawonn (2000) adapted the Euclidean distance metric of FCM by using cluster-specific weights for each feature. Modha and Spangler (2003) presented the convex k-means algorithm where the feature weights are determined by minimizing a ratio of the average within-cluster distortion to the average between-cluster distortion. Huang et al. (2005) introduced a new step in the k-means algorithm to refine the feature weights iteratively based on the current clustering state. This particular notion of automated weighting was later integrated with FCM (Nazari et al. 2013). Hung et al. (2011) proposed an exponential distance-based clustering algorithm with similar automated feature weight learning and spatial constraints for image segmentation. Recently Saha and Das (2015b) extended the weight learning strategy to fuzzy k-modes clustering of categorical data.

Optimization with respect to a symmetric, positive definite matrix, ensures scaling with respect to the variance of different variables. The main drawback of Mahalanobis clustering is the summing up of variance normalized squared distance with equal weight (Wölfel and Ekenel 2005). In absence of noise variable, i.e. where each variable contributes in determining the underlying cluster structure, clustering with Mahalanobis distance provides perfect results. But in presence of a noise feature with extremely high values, the equal weighting method subdues the importance of the other variables, which leads to undesirable results (Wölfel and Ekenel 2005). For a more detailed discussion on these issues, see the "Appendix". In order to find a clustering method robust enough to the noise variables, even after variance normalization, we do need to find a distance measure which gives less weight to the noise variable and more weight to the features, thus contributing to determine the cluster structure.

2.3 Notations

A few words about the notation used: bold faced letters, e.g., $\mathbf{x}$, $\mathbf{y}$ represent vectors. Calligraphic upper-case alphabets denote sets, e.g., $\mathcal{X}$, $\mathcal{Y}$. Matrices are expressed with upper-case bold faced letters, e.g., $\mathbf{X}$, $\mathbf{Y}$. The symbols $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{R}^d$ denote the set of real numbers, the set of natural numbers, and the d-dimensional real vector space respectively. Further, $\mathbb{R}_+$ denotes the set of non-negative real numbers.

3 Clustering with the IPIN-based weighted dissimilarity measures

In this section, following the philosophy of Klawonn and Höppner (2003), we introduce the general class of IPIN-based weighted dissimilarity measures in an axiomatic approach. It is a fairly general and large class of weighted dissimilarity measures and deals with a general weight function.

3.1 The IPIN-based weighted dissimilarity measures

Let $\mathcal{M}^d$ denote the class of all symmetric, positive definite matrices $\mathbf{A}$ with finite Frobenius norm (Hilbert–Schmidt norm) (Golub and Van Loan 2012) i.e. $||\mathbf{A}||_2^2 < \infty$. The standard $d-1$-dimensional simplex $\mathcal{H}^d$ is defined as follows:

$$\mathcal{H}^d = \left\{ (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d \,\middle|\, \sum_{l=1}^{d} w_l = 1,\ w_l \geqslant 0,\ 1 \leqslant l \leqslant d \right\} \tag{1}$$

Let,

$$\mathbf{M} \in \mathcal{M}^d, \quad g : [0,1] \to \mathbb{R}_+,\ \mathbf{w} \in \mathcal{H}^d.$$

Then we define $\mathbf{M}_{\mathbf{w},g}$ as follows:

$$\mathbf{M}_{\mathbf{w},g} = \begin{pmatrix} g(w_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(w_d) \end{pmatrix} \mathbf{M} \begin{pmatrix} g(w_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(w_d) \end{pmatrix}, \tag{2}$$

Definition 1 For any $g : [0,1] \to \mathbb{R}_+$, a function $dist_g : \mathcal{M}^d \times \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{H}^d \to \mathbb{R}_+$ is called an IPIN-based Consistent Weighted Dissimilarity (IPINCWD) measure with respect to some convex set $\mathcal{C}_1 \subseteq \mathbb{R}_d$ and $\mathcal{C}_2 \subseteq \mathcal{M}^d$ if for some function $h : \mathbb{R} \to \mathbb{R}_+$, $dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w}) = d_{\mathbf{M}_{\mathbf{w},g}}(\mathbf{y}, \mathbf{x}) = h((\mathbf{x}-\mathbf{y})^T \mathbf{M}_{\mathbf{w},g}(\mathbf{x}-\mathbf{y}))$, where the following assumptions hold:

1. h is differentiable on $\mathbb{R}_+$.
2. $g : [0,1] \to \mathbb{R}_+$ is strictly increasing, differentiable function.
3. $\mathbf{M} \to dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$ is a strictly convex or linear function on $\mathcal{C}_2$, $\forall \mathbf{y} \in \mathcal{C}_1$, $\forall \mathbf{w} \in \mathcal{H}^d$, $\mathbf{y} \to dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$ is strictly convex function on $\mathcal{C}_1$, $\forall \mathbf{M} \in \mathcal{C}_2$, $\forall \mathbf{w} \in \mathcal{H}^d$, and $\mathbf{w} \to dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$ is strictly convex function on $\mathcal{H}_d$, $\forall \mathbf{M} \in \mathcal{C}_2$, $\forall \mathbf{y} \in \mathcal{C}_1$.

We denote the family of functions $dist_g$ satisfying the premises of Definition 1 by $\mathcal{D}_g(\mathcal{C}_1, \mathcal{C}_2)$. The very definition of $dist_g$ ensures that it is symmetric. Note that the definition of $dist$ does not require the triangle inequality to hold and hence this particular class of dissimilarity measures need not be a metric or distance function in the true sense. The

motivation behind the technical assumptions in Definition 1 will be evident from the mathematical development in Sect. 4. It should be noted that a sufficient condition for assumption 3 to hold is the increasing and convex nature of h and g at every point on their respective domains.

3.2 Examples of IPINCWD measures

We provide examples of two IPINCWD measures. The unweighted version of the first one of these has been extensively used for clustering (Saha and Das 2016b), while the other one has been observed to yield robust clustering being resistant to the presence of noise (Hung et al. 2011).

3.2.1 Weighted IPIN

The common IPIN is a popular distance measure for clustering (Gustafson and Kessel 1978). It is defined (corresponding to $\mathbf{M} \in \mathcal{M}^d$) as follows,

$$(\mathbf{x}-\mathbf{y})^T\mathbf{M}(\mathbf{x}-\mathbf{y}), \quad \forall \mathbf{M} \in \mathcal{M}^d, \mathbf{y}, \mathbf{x} \in \mathbb{R}^d.$$

Hence, the weighted IPIN (corresponding to $\mathbf{M} \in \mathcal{M}^d$) can be defined as follows ($g(x) = x^m;\ m > 1$):

$$dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w}) = (\mathbf{x}-\mathbf{y})^T\mathbf{M}_{\mathbf{w},g}(\mathbf{x}-\mathbf{y}), \quad \forall \mathbf{M} \in \mathcal{M}^d, \forall \mathbf{y}, \mathbf{x} \in \mathbb{R}^d, \forall \mathbf{w} \in \mathcal{H}^d.$$

Hence, in this case, h is the identity function on non-negative real line, i.e. $h(x) = x, \forall x \in \mathbb{R}_+$.

Now, h is trivially differentiable everywhere. $(\mathbf{x}-\mathbf{y})^T\mathbf{M}_{\mathbf{w},g}(\mathbf{x}-\mathbf{y})$ is strictly convex with respect to $\mathbf{y}$, hence so is $dist(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$. Moreover, $dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$ is a linear function with respect to $\mathbf{M}$. From the strict convexity of g it follows that $dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w})$ is a strictly convex function of $\mathbf{w}$. Hence this dissimilarity measure is a valid member of the IPINCWD class.

3.2.2 Weighted exponential IPIN

The exponential IPIN-based dissimilarity measure can be very helpful in achieving robust clustering since it has been shown to provide natural resistance against noise (Hung et al. 2011). Under the realistic assumption that the concerned $\mathcal{C}_1$ and $\mathcal{C}_2$ in Definition 1 are bounded, it can be defined as follows ($g(x) = x^m\ m > 1$):

$$dist_g(\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{w}) = \exp\left\{(\mathbf{x}-\mathbf{y})^T\mathbf{M}_{\mathbf{w},g}(\mathbf{x}-\mathbf{y})\right\}, \quad \forall \mathbf{M} \in \mathcal{M}^d, \forall \mathbf{y}, \mathbf{x} \in \mathbb{R}^d;\ \forall \mathbf{w} \in \mathcal{H}^d.$$

Hence, in this case, h is the exponential function on non-negative real line, i.e. $h(x) = \exp(x), \forall x \in \mathbb{R}_+$. Here also h is trivially differentiable everywhere. Since, the composition of a strictly convex increasing function (exponential function in this case) with a convex function is again strictly convex, assumption 3 in Definition 1 is satisfied.

On the other hand, as far as choices of g are concerned, some common choices of g (under the simplifying assumption that $h(x) = x$), can be given as follows:

$$\begin{aligned} g(x) &= x^\beta, \beta > 1, \\ g(x) &= \exp(x), \\ g(x) &= \exp(x^\beta), \beta > 1. \end{aligned}$$

Hence this measure also fit well into the IPINCWD class.

3.3 Problem formulation and algorithm development

In this section, we present the general class of clustering problems with IPIN-based weighted dissimilarity measures. We also develop a Llyod heuristic and an AO algorithm to solve the k-means and FCM clustering problems respectively.

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d, \forall i = 1, 2, \ldots, n$ be the given set of patterns, which we want to partition into c (prefixed) clusters with $2 \leqslant c \leqslant n$. Let $\mathcal{B} \subset \mathbb{R}^d$ be the convex hull of $\mathcal{X}$. The general clustering problem with any member of $\mathcal{D}(\mathcal{B}, \mathcal{M}^d)$(both $\mathcal{B}$ and $\mathcal{M}^d$ are convex, hence this class of IPINCWD is well-defined) is defined in the following way (fuzzifier $m \geqslant 1, \rho_j > 0, \forall j = 1, 2, \ldots, c$):

$$\mathbf{P}: \quad \textit{minimize } f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{W}) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m d_{\Sigma_j \mathbf{w}_j, g}(\mathbf{z}_j, \mathbf{x}_i), \tag{3}$$

subject to

$$\sum_{j=1}^{c} u_{ij} = 1, \quad \forall i = 1, 2, \ldots, n, \tag{4a}$$

$$0 < \sum_{i=1}^{n} u_{ij} < n, \quad \forall j = 1, 2, \ldots, c, \tag{4b}$$

$$u_{ij} \in [0, 1], \quad \forall i = 1, 2, \ldots, n; \ \forall j = 1, 2, \ldots, c, \tag{4c}$$

$$\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_c\}, \quad \mathbf{z}_j \in \mathcal{B} \subseteq \mathbb{R}^d, \ \forall j = 1, 2, \ldots, c; \mathcal{Z} \in \mathcal{B}^c \subset \mathbb{R}^{d \times c}, \tag{4d}$$

$$\mathcal{S} = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_c\}, \quad \Sigma_j \in \mathcal{M}^d, \ \forall j = 1, 2, \ldots, c; \mathcal{S} \in \mathcal{M}^{d \times c}, \tag{4e}$$

$$|\Sigma_j| = \rho_j, \quad \forall j = 1, 2, \ldots, c, \tag{4f}$$

$$\mathbf{w}_j \in \mathcal{H}^d \quad \forall d \in \{1, 2, \ldots, c\}; \ \mathcal{W} \in \mathcal{H}^{d \times c}. \tag{4g}$$

To solve k-means and FCM problems, in this section, we present a Lyod's heuristic and an AO procedure respectively. The general algorithm is schematically presented as Algorithm 1.

For hard clustering, we fix $m = 1$, whereas, for fuzzy clustering, we take $m > 1$. The general algorithm for solving the automated feature-weighted IPINCWD-based clustering algorithms is provided in Algorithm 1.

4 Convergence analysis of clustering with IPIN-based consistent dissimilarity measures

We carry out a full-fledged convergence analysis of the generic IPINCWD-based clustering procedure shown in Algorithm 1. First, we address the existence and uniqueness of the partial optimization problems with respect to the cluster representatives, norm inducing matrices, and weights.

For the sake of notational simplicity, we define the following:

$$\begin{aligned}
\mathcal{U}_{c,n} &= \{\mathbf{U} \mid \mathbf{U} \text{ is a } n \times c \text{ real matrix and } \mathbf{U} \text{ satisfies (2a)–(2c)}\},\\
\mathcal{M}_{d,\rho_j} &= \{\mathbf{M} \in \mathcal{M}^d \mid |\mathbf{M}| = \rho_j\},\\
\mathcal{M}_{d,\boldsymbol{\rho}} &= \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_c \mid \mathbf{M}_j \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c\}; \quad \forall \boldsymbol{\rho} \in \mathbb{R}_+^c.
\end{aligned}$$

ALGORITHM 1: Clustering with IPINCWD

Data: Datapoints, number of clusters, exponent of membership matrix, permissible fractional error, choice of h and g.
Result: Membership matrix of objects in clusters, cluster representatives, norm-inducing matrices, weights.
Initialization:
$c \leftarrow$ number of *clusters*;
$m \leftarrow$ value of *fuzzifier*;
$\epsilon \leftarrow$ permissible fractional *error*;
$\mathcal{Z}^{(0)} \leftarrow$ initial *clusters* satisfying (4d);
$\mathcal{S}^{(0)} \leftarrow$ initial *Norm inducing Matrices* satisfying (4e)–(4f);
$\mathcal{W}^{(0)} \leftarrow$ initial *Weights* satisfying (4g);
$t \leftarrow 0$;
while $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}^{(t-1)}, \mathcal{W}^{(t-1)}) - f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t)}, \mathcal{Z}^{(t)}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)}) \geqslant \epsilon f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}^{(t-1)}, , \mathcal{W}^{(t-1)})$ **do**
$\mathbf{U}^{(t+1)} = \text{argmin}_{\mathbf{U}\in\mathcal{U}_{c,n}} \; f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}, \mathcal{Z}^{(t)}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)})$;
$\mathcal{Z}^{(t+1)} = \text{argmin}_{\mathcal{Z}_j\in\mathcal{B}^c} \; f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t+1)}, \mathcal{Z}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)})$;
$\mathcal{S}^{(t+1)} = \text{argmin}_{\mathcal{S}\in\mathcal{M}_{d,\boldsymbol{\rho}}} \; f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t+1)}, \mathcal{Z}^{(t+1)}, \mathcal{S}, \mathcal{W}^{(t)})$;
$\mathcal{W}^{(t+1)} = \text{argmin}_{\mathcal{W}\in\mathcal{H}^{d\times c}} \; f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t+1)}, \mathcal{Z}^{(t+1)}, \mathcal{S}^{(t+1)}, \mathcal{W})$;
$t = t + 1$
end

Theorem 1 *For fixed* $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{W}^* = \{\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_c^*\}, \mathbf{w}_j^* \in \mathcal{H}^c, \forall j = 1, 2, \ldots, c$; *and* $\mathcal{S}^* = \{\Sigma_1^*, \Sigma_2^*, \ldots, \Sigma_c^*\}, \Sigma_j^* \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, *the problem* $\mathbf{P}_1$: *minimize* $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}, \mathcal{S}^*, \mathcal{W}^*)$, $\mathcal{Z} \in \mathcal{B}^c$, *has a unique solution.*

Proof The function to be minimized in problem $\mathbf{P}_1$ is a convex function with respect to $\mathcal{Z}$ (from the assumption on IPINCWD measures, Definition 1) and the optimization task is carried out on a convex set. Hence, there exists at most one solution.

Now, the function under consideration is also a continuous function with respect to $\mathcal{Z}$ (from the assumption on IPINCWD measures, Definition 1). Thus it attains its maxima and minima in a closed and bounded interval, which is indeed the case here. Hence, the minimization task under consideration has at least one solution in the feasible region.

Employing the two aforementioned statements, we guarantee the existence of unique solution of the optimization task $\mathbf{P}_1$ in the feasible region. □

Theorem 2 *Let* $J_1 : \mathcal{B}^c \rightarrow \mathbb{R}, J_1(\mathcal{Z}) = f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}, \mathcal{S}^*, \mathcal{W}^*); \mathbf{z}_j \in \mathcal{B}, \forall j = 1, 2, \ldots, c$, *where* $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{W}^* \in \mathcal{H}^{d\times c}$, $\mathcal{S}^* \in \mathcal{M}_{d,\boldsymbol{\rho}}$ *are fixed. Then* $\mathcal{Z}^*$ *is a global minimum of* J_1 *if and only if* $\mathbf{z}_j^*$ *satisfies the following equation*

$$\sum_{i=1}^{n} (u_{ij}^*)^m h'((\mathbf{x}_i - \mathbf{z}_j^*)^T \Sigma_{j\,\mathbf{w}_j,g}^* (\mathbf{x}_i - \mathbf{z}_j^*)) \Sigma_{j\,\mathbf{w}_j,g}^* (\mathbf{x}_i - \mathbf{z}_j^*) = 0; \;\; j = 1, 2, \ldots, c. \quad (5)$$

Proof The condition in the theorem is derived as a first order necessary condition by setting the partial derivative of the objective function with respect to $\mathbf{z}_j$ equal to zero. From the strict convexity of the objective function with respect to $\mathbf{z}_j$, it follows that those conditions are also sufficient conditions to uniquely determine the minimizer. □

Theorem 3 *For fixed* $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{Z}^* = \{\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_c^*\}, \mathbf{z}_j^* \in \mathcal{B}^c, \forall j = 1, 2, \ldots, c, \mathcal{W}^* = \{\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_c^*\}, \mathbf{w}_j^* \in \mathcal{H}^d, \forall j = 1, 2, \ldots, c; \boldsymbol{\rho} \in \mathbb{R}_+^c$, *the problem* $\mathbf{P}_2$: *minimize*

$f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*), \mathcal{S} = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_c\}, \Sigma_j \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, *has a unique solution.*

Proof Here, from the definition, it is clear that $\mathcal{M}_{d,\rho_j}$ is the inverse image of the compact set $\{\rho_j\}$ with respect to the continuous function det (determinant function). Hence the feasible set $\mathcal{M}_{d,\rho_j}$ is a compact set. Thus from continuity of $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*)$ with respect to Σ_j, there exists at most one solution to the problem under consideration.

From the assumptions on the IPINCWD, we have that $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*)$ is a non-negative linear combination ($\mathbf{U}^*$ is not identically 0) of strictly convex functions in Σ_j and hence is strictly convex with respect to Σ_j. To prove the convexity of the feasible set, we proceed as follows: We perform the following optimization task,

$$\begin{aligned} &\mathbf{EP}_2 \;\; \textit{minimize } f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*), \\ &\mathcal{S} = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_c\}, \Sigma_j \in \mathcal{F}_{d,\rho_j}, \forall j = 1, 2, \ldots, c; \end{aligned}$$

where

$$\mathcal{F}_{d,\rho_j} = \left\{ \mathbf{M} \in \mathcal{M}^d \mid |\mathbf{M}| \geqslant \rho_j \right\}.$$

$\mathcal{F}_{d,\rho_j}$ being convex set, there can be at most one solution. Now, we observe that any minimizer of the objective function in $\mathcal{F}_{d,\rho_j}$ has to be in $\mathcal{M}_{d,\rho_j}$ (if not, we can divide by a suitable constant to get a minimizer in $\mathcal{M}_{d,\rho_j}$). Hence, the objective function under consideration can have at most one solution in $\mathcal{M}_{d,\rho_j}$.

These two facts together guarantee that the optimization task with respect to matrices inducing inner products ($\mathbf{P}_2$), has a unique solution. □

Theorem 4 *Let* $J_2 : \mathcal{M}_{d,\boldsymbol{\rho}} \to \mathbb{R}$, $J_2(\mathcal{S}) = f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*)$; $\Sigma_j \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, *where* $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{Z}^* \in \mathcal{B}^c$, $\mathcal{W}^* \in \mathcal{H}^{d\times c}$ *are fixed. Then* $\mathcal{S}^*$ *is a global minimum of* J_2 *if and only if* Σ_j^* *satisfies the following equation:*

$$\mathbf{M}_j^{*-1}(\rho_j|\mathbf{M}_j^*|)^{\frac{1}{d}} = \Sigma_j^*; \;\; j = 1, 2, \ldots, c. \tag{6}$$

where,

$$\mathbf{M}_j^* = \sum_{i=1}^{n} (u_{ij}^*)^m h'((\mathbf{x}_i - \mathbf{z}_j^*)^T \Sigma_{j\,\mathbf{w}_j^*,g}^* (\mathbf{x}_i - \mathbf{z}_j^*)) \left[(\mathbf{x}_i - \mathbf{z}_j^*)(\mathbf{x}_i - \mathbf{z}_j^*)^T \right]_{\mathbf{w}_j^*,g}.$$

Proof Follows from the proof of Theorem 2. Like Theorem 2, the condition is obtained as the first order necessary condition by setting the first derivative equal to zero. The uniqueness follows from the strict convexity. □

Theorem 5 *For fixed* $\mathcal{Z}^* = \{\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_c\}, \mathbf{z}_j^* \in \mathcal{B}^c$, $\mathcal{W}^* = \{\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_c^*\}, \mathbf{w}_j^* \in \mathcal{H}^d, \forall j = 1, 2, \ldots, c$ *and* $\mathcal{S}^* = \{\Sigma_1^*, \Sigma_2^*, \ldots, \Sigma_c^*\}, \Sigma_j^* \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, *the problem* $\mathbf{P}_3$: *minimize* $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*), \mathbf{U} \in \mathcal{U}_{c,n}$, *has the solution* $\mathbf{U}^*$ *given by the following:*

For $m = 1$ *(hard clustering)*

$$\begin{aligned} \zeta_i^* &= \left\{ j \mid d_{\Sigma_{j\,\mathbf{w}_j^*,g}^*}(\mathbf{z}_j^*, \mathbf{x}_i) = \min_{1\leqslant k\leqslant c} d_{\Sigma_{k\,\mathbf{w}_k^*,g}^*}(\mathbf{z}_k^*, \mathbf{x}_i) \right\}, \\ u_{ij}^* &= \begin{cases} 1 \textit{ or } 0 \textit{ with } \sum_{k\in\zeta_i^*} u_{ik} = 1, & \textit{if } j \in \zeta_i^* \\ 0, & \textit{otherwise.} \end{cases} \end{aligned} \tag{7a}$$

For $m > 1$ (fuzzy clustering)

$$\psi_i^* = \left\{ j \mid d_{\Sigma_{j\,\mathbf{w}_j^*,g}^*}(\mathbf{z}_j^*, \mathbf{x}_i) = 0 \right\},$$

$$u_{ij}^* = \begin{cases} \left[\sum_{k=1}^{c} \left[\frac{d_{\Sigma_{j\,\mathbf{w}_j^*,g}^*}(\mathbf{z}_j^*, \mathbf{x}_i)}{d_{\Sigma_{k\,\mathbf{w}_k^*,g}^*}(\mathbf{z}_k^*, \mathbf{x}_i)} \right]^{\frac{1}{m-1}} \right]^{-1}, & \text{if } \psi_i^* = \phi \\ \geqslant 0 \text{ with } \sum_{\mathbf{z}_k^* = \mathbf{x}_i} u_{ik}^* = 1, & \text{if } j \in \psi_i^* \\ 0. & \text{if } j \notin \psi_i^* \text{ and } \psi_i^* \neq \phi. \end{cases} \tag{7b}$$

Proof As far as the k-means algorithm is concerned, the proof follows from the concave nature of the problem with respect to membership matrix which ensures that the minimum is realized at a boundary point of $\mathbf{U}_{c,n}$ (Selim and Ismail 1984).

In the case of FCM, the membership matrix updating rule follows from the techniques employed to obtain the membership updating rule corresponding to conventional FCM with squared Euclidean distance (Bezdek 1981). □

Theorem 6 *For fixed $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{Z}^* = \{\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_c^*\}, \mathbf{z}_j^* \in \mathcal{B}^c, \forall j = 1, 2, \ldots, c, \mathcal{S}^* = \{\Sigma_1^*, \Sigma_2^*, \ldots, \Sigma_c^*\}, \Sigma_j^* \in \mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, the problem $\mathbf{P}_4$ minimize $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}), \mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c\}, \mathbf{w}_j \in \mathcal{H}^d, \forall j = 1, 2, \ldots, c$, has a unique solution.*

Proof Follows from the proof of Theorem 1. Alike the case in Theorem 1, the problem in hand is a convex optimization on a convex set, hence have at most one solution. Moreover, it being a continuous function optimized on a compact set, does attend its extreme values. Hence, the uniqueness and existence of the solution follows. □

Theorem 7 *Let $J_3 : \mathcal{H}^{d\times c} \to \mathbb{R}, J_3(\mathcal{W}) = f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}); \mathbf{w}_j \in \mathcal{H}^d, \forall j = 1, 2, \ldots, c$, where $\mathbf{U}^* \in \mathcal{U}_{c,n}$, $\mathcal{Z}^* \in \mathcal{B}^c$, $\mathcal{S}^* \in \mathcal{M}_{d,\boldsymbol{\rho}}$ are fixed. Then $\mathcal{W}^*$ is a global minimum of J_3 if and only if $\mathbf{w}_j^*$ satisfies the following equation*

$$\mathbf{w}_j^* = \mathbf{v}_j^{*2} = \left({v_{j1}^*}^2, {v_{j2}^*}^2, \ldots, {v_{jd}^*}^2 \right) \quad \forall j = 1, 2, \ldots, c. \tag{8a}$$

$$\sum_{i=1}^{n} u_{ij}^m \frac{\partial}{\partial \mathbf{y}_j} d_{\Sigma_{j\,\mathbf{y}_j^2,g}^*}(\mathbf{z}_j^*, \mathbf{x}_i)|_{\mathbf{y}_j = \mathbf{v}_j^*} = -L \frac{\partial}{\partial \mathbf{y}_j} \mathbf{y}_j^T \mathbf{y}_j |_{\mathbf{y}_j = \mathbf{v}_j^*}, \quad \forall j = 1, 2, \ldots, c. \tag{8b}$$

$$\mathbf{v}_j^{*T} \mathbf{v}_j^* = \rho_j \quad \forall j = 1, 2, \ldots, c. \tag{8c}$$

where L is any constant.

Proof Follows from the proofs of Theorems 2 and 5 by replacing the non-negative weights with squares of unconstrained real numbers and using the Lagrange multiplier technique with the linear constraints on the sum of the weights. □

Our proof of convergence for the proposed feature-weighted clustering algorithms are based on Zangwill's global convergence theorem (Zangwill 1969). From the aforementioned theorems, the updating rules corresponding to the membership matrix, cluster representatives, matrices of inner product inducing norms, and the feature weights corresponding to each of the clusters can be interpreted with help of the following operators.

Definition 2 $T_{memb} : \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c} \to \mathcal{U}_{c,n}$; $T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}) = \mathbf{U} = [u_{ij}]$, which is given by the following rule: For $m = 1$ (hard clustering)

$$\zeta_i = \left\{ j \mid d_{\Sigma_{j\mathbf{w}_j,g}}(\mathbf{z}_j, \mathbf{x}_i) = \min_{1\leqslant k\leqslant c} d_{\Sigma_{k\mathbf{w}_k,g}}(\mathbf{z}_k, \mathbf{x}_i) \right\},$$
$$u_{ij} = \begin{cases} 1 \text{ or } 0 \text{ with } \sum_{k\in\zeta_i} u_{ik} = 1, & \text{if } j \in \zeta_i \\ 0, & \text{otherwise.} \end{cases} \tag{9a}$$

For $m > 1$ (fuzzy clustering)

$$\psi_i = \left\{ j \mid d_{\Sigma_{j\mathbf{w}_j,g}}(\mathbf{z}_j, \mathbf{x}_i) = 0 \right\},$$
$$u_{ij} = \begin{cases} \left[\sum_{k=1}^{c} \left[\frac{d_{\Sigma_{j\mathbf{w}_j,g}}(\mathbf{z}_j,\mathbf{x}_i)}{d_{\Sigma_{k\mathbf{w}_k,g}}(\mathbf{z}_k,\mathbf{x}_i)} \right]^{\frac{1}{m-1}} \right]^{-1}, & \text{if } \psi_i = \phi \\ \geqslant 0 \text{ with } \sum\limits_{\mathbf{z}_k=\mathbf{x}_i} u_{ik} = 1, & \text{if } j \in \psi_i \\ 0. & \text{if } j \notin \psi_i \text{ and } \psi_i \neq \phi. \end{cases} \tag{9b}$$

Definition 3

$$T_{cent} : \mathcal{U}_{c,n} \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c} \to \mathcal{B}^c; \;\; T_{cent}(\mathbf{U}, \mathcal{S}, \mathcal{W}) = \mathcal{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_c),$$

where the vectors $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_c), \mathbf{z}_j \in \mathcal{B}, j = 1, 2, \ldots, c$ are calculated such that they maintain the following condition

$$\sum_{i=1}^{n} (u_{ij})^m h' \left[(\mathbf{x}_i - \mathbf{z}_j)^T \Sigma_{j\mathbf{w}_j,g} (\mathbf{x}_i - \mathbf{z}_j) \right] \Sigma_{j\mathbf{w}_j,g} (\mathbf{x}_i - \mathbf{z}_j) = 0; \quad \forall j = 1, 2, \ldots, c. \tag{10}$$

Definition 4

$$T_{matrix} : \mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{H}^{d\times c} \to \mathcal{M}_{d,\boldsymbol{\rho}}; \;\; T_{matrix}(\mathbf{U}, \mathcal{Z}, \mathcal{W}) = \mathcal{S} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_c),$$

where the matrices $(\Sigma_1, \Sigma_2, \ldots, \Sigma_c), \Sigma_j \in \mathcal{M}_{d,\rho_j}, j = 1, 2, \ldots, c$ are calculated such that they maintain the following condition:

$$\mathbf{M}_j^{-1} (\rho_j |\mathbf{M}_j|)^{\frac{1}{d}} = \Sigma_j; \quad j = 1, 2, \ldots, c; \tag{11}$$

where,

$$\mathbf{M}_j = \sum_{i=1}^{n} (u_{ij})^m h' ((\mathbf{x}_i - \mathbf{z}_j)^T \Sigma_{j\mathbf{w}_j,g} (\mathbf{x}_i - \mathbf{z}_j)) \left[(\mathbf{x}_i - \mathbf{z}_j)(\mathbf{x}_i - \mathbf{z}_j)^T \right]_{\mathbf{w}_j,g}.$$

Definition 5

$$T_{weight} : \mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \to \mathcal{H}^{d\times c}; \;\; T_{weight}(\mathbf{U}, \mathcal{Z}, \mathcal{S}) = \mathcal{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c),$$

where the vectors $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c), \mathbf{w}_j \in \mathcal{H}^d, j = 1, 2, \ldots, c$ are calculated such that they maintain the following condition:

$$\mathbf{w}_j = \mathbf{v}_j^2 = \left(v_{j1}^2, v_{j2}^2, \ldots, v_{jd}^2 \right) \quad \forall j = 1, 2, \ldots, c. \tag{12a}$$

$$\sum_{i=1}^{n} u_{ij}^m \frac{\partial}{\partial \mathbf{y}_j} d_{\Sigma_{j\mathbf{y}_j^2,g}}(\mathbf{z}_j, \mathbf{x}_i)|_{\mathbf{y}_j=\mathbf{v}_j} = -L \frac{\partial}{\partial \mathbf{y}_j} \mathbf{y}_j^T \mathbf{y}_j |_{\mathbf{y}_j=\mathbf{v}_j}, \quad \forall j = 1, 2, \ldots, c. \tag{12b}$$

$$\mathbf{v}_j^T \mathbf{v}_j = \rho_j \quad \forall j = 1, 2, \ldots, c. \tag{12c}$$

where L is any constant.

With the help of these newly defined operators, that provide updating rules corresponding to the membership matrix, cluster representatives, matrices of IPINs and the feature weights corresponding to each of the clusters, the automated feature-weighted clustering algorithm with IPINCWD (Algorithm 1) is restated in Algorithm 2.

With the help of these newly defined operators, the clustering operator can be presented as follows:

Definition 6

$$J : (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}) \to (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c});$$
$$J = O_{weight} \circ O_{matrix} \circ O_{cent} \circ O_{memb},$$

where

$$O_{memb} : (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}) \to (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}),$$
$$O_{memb}(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{Z}) = (T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}).$$
$$O_{cent} : (\mathcal{U}_{c,n} \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}) \to (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{H}^{d\times c}),$$
$$O_{cent}(\mathbf{U}, \mathcal{S}, \mathcal{W}) = (\mathbf{U}, T_{cent}(\mathbf{U}, \mathcal{S}, \mathcal{W}), \mathcal{W}).$$
$$O_{matrix} : (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{H}^{d\times c}) \to (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}}),$$
$$O_{matrix}(\mathbf{U}, \mathcal{Z}, \mathcal{W}) = (\mathbf{U}, \mathcal{Z}, T_{matrix}(\mathbf{U}, \mathcal{Z}, \mathcal{W})).$$
$$O_{weight} : (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}}) \to (\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}),$$
$$O_{weight}(\mathbf{U}, \mathcal{Z}, \mathcal{S}) = (\mathbf{U}, \mathcal{Z}, \mathcal{S}, T_{weight}(\mathbf{U}, \mathcal{Z}, \mathcal{S})).$$

$$\begin{aligned}
J(\mathbf{U}, \mathcal{Z}, \mathcal{S}.\mathcal{W}) &= O_{weight} \circ O_{matrix} \circ O_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}) \\
&= O_{weight} \circ O_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W}) \\
&= O_{weight}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\quad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W})) \\
&= (T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\quad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W}), \\
&\quad T_{weight}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\quad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W})))
\end{aligned}$$

In what follows we prove the convergence of the iterative algorithm (Algorithm 2) using Zangwill's global convergence theorem (Zangwill 1969).

Lemma 1 *$\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$ is compact.*

Proof $\mathcal{U}_{c,n}$ being closure of a set, is a closed set (Bezdek 1981), which accompanied with the boundedness of the set implies that it is a compact set. $\mathcal{B}$ being convex hull of finitely many points, is compact, ensuring the compactness of $\mathcal{B}^c$. In Theorem 3 the compactness of $\mathcal{M}_{d,\rho_j}, \forall j = 1, 2, \ldots, c$, was proved, which in turn implies the compactness of $\mathcal{M}_{d,\boldsymbol{\rho}}$. The simplex is a compact set by definition, hence, the set under consideration being a product of four compact sets, is again compact. □

ALGORITHM 2: Clustering with IPINCWD

Data: Data points, number of clusters, exponent of membership matrix, permissible fractional error, choice of h and g.
Result: Membership matrix of objects in clusters, cluster representatives, norm-inducing matrices, weights.
Initialization:
$c \leftarrow$ number of *clusters*;
$m \leftarrow$ value of *fuzzifier*;
$\epsilon \leftarrow$ permissible fractional *error*;
$\mathcal{Z}^{(0)} \leftarrow$ initial *clusters* satisfying (4d);
$\mathcal{S}^{(0)} \leftarrow$ initial *Norm inducing Matrices* satisfying (4e)-(4f);
$\mathcal{W}^{(0)} \leftarrow$ initial *Weights* satisfying (4g);
$t \leftarrow 0$;
while $f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}^{(t-1)}, \mathcal{W}^{(t-1)}) - f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t)}, \mathcal{Z}^{(t)}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)}) \geqslant \epsilon f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^{(t-1)}, \mathcal{Z}^{(t-1)}, \mathcal{S}^{(t-1)}, , \mathcal{W}^{(t-1)})$ **do**
  for $1 \leqslant i \leqslant n$ **do**
    for $1 \leqslant j \leqslant c$ **do**
      Calculate $d_{\Sigma_j^{(t)}\mathbf{w}_j^{(t)},g}(\mathbf{z}_j^{(t)}, \mathbf{x}_i) = (\mathbf{x}_i - \mathbf{z}_j^{(t)})^T \Sigma_{j\mathbf{w}_j^{(t)},g}(\mathbf{x}_i - \mathbf{z}_j^{(t)})$;
    end
  end
  if $m > 1$ **then**
    for $1 \leqslant i \leqslant n$ **do**
      for $1 \leqslant j \leqslant c$ **do**
        if $d_{\Sigma_j^{(t)}\mathbf{w}_j^{(t)},g}(\mathbf{z}_j^{(t)}, \mathbf{x}_i) = 0$, *for some* $k = 1, 2, \ldots, c$; **then**
          $u_{ik}^{(t+1)} = 1$;
          $u_{ij}^{(t+1)} = 0;\ \forall j = 1, 2, \ldots, c;\ j \neq k$;
        else
          $u_{ij}^{(t+1)} = \left[\sum_{k=1}^{c} \left[d_{\Sigma_j^{(t)}\mathbf{w}_j^{(t)},g}(\mathbf{z}_j^{(t)}, \mathbf{x}_i) / d_{\Sigma_k^{(t)}\mathbf{w}_k^{(t)},g}(\mathbf{z}_k^{(t)}, \mathbf{x}_i) \right]^{\frac{1}{m-1}} \right]^{-1}$;
        end
      end
    end
  end
  if $m = 1$ **then**
    for $1 \leqslant i \leqslant n$ **do**
      Calculate $\zeta_i^{(t)} = \{j \mid d_{\Sigma_j^{(t)}\mathbf{w}_j^{(t)},g}(\mathbf{z}_j^{(t)}, \mathbf{x}_i) = \min_{1 \leqslant k \leqslant c} d_{\Sigma_k^{(t)}\mathbf{w}_k^{(t)},g}(\mathbf{z}_k^{(t)}, \mathbf{x}_i)\}$;
      for $1 \leqslant j \leqslant c$ **do**
        if $j = \min \zeta_i^{(t)}$ **then**
          $u_{ij}^{(t+1)} = 1$;
        else
          $u_{ij}^{(t+1)} = 0$;
        end
      end
    end
  end
  $\mathcal{Z}^{(t+1)} = T_{cent}(\mathbf{U}^{(t+1)}, \mathcal{Z}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)})$;
  $\mathcal{S}^{(t+1)} = T_{matrix}(\mathbf{U}^{(t+1)}, \mathcal{Z}^{(t+1)}, \mathcal{S}, \mathcal{W}^{(t)})$;
  $\mathcal{W}^{(t+1)} = T_{weight}(\mathbf{U}^{(t+1)}, \mathcal{Z}^{(t+1)}, \mathcal{S}^{(t+1)}, \mathcal{W})$;
  $t = t + 1$
end

We define the set of optimal points in the perspective of IPINCWD-based automated feature-weighted clustering algorithm in the following way.

Definition 7 $\mathcal{T}$ is a subset of $\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$, where

$$\left.\begin{aligned}\mathcal{T} = \big\{ &(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*) \in \mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c} \mid \\ &f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*) \leqslant f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*), \forall \mathbf{U} \in \mathcal{U}_{c,n}, \mathbf{U} \neq \mathbf{U}^*, \\ &f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*) < f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}, \mathcal{S}^*, \mathcal{W}^*), \forall \mathcal{Z} \in \mathcal{B}^c, \mathcal{Z} \neq \mathcal{Z}^*, \\ &f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*) < f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}, \mathcal{W}^*), \forall \mathcal{S} \in \mathcal{M}_{d,\boldsymbol{\rho}}, \mathcal{S} \neq \mathcal{S}^*, \\ &f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}^*, \mathcal{S}^*, \mathcal{W}^*) < f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}^*, \mathcal{Z}, \mathcal{S}^*, \mathcal{W}), \forall \mathcal{W} \in \mathcal{H}^{d\times c}, \mathcal{W} \neq \mathcal{W}^*. \end{aligned}\right\} \tag{13}$$

Lemma 2 *The set defined in* (13) *satisfies the following two conditions:*

1. *If* $\mathbf{g} \notin \mathcal{T}$, *then* $f_{m,\boldsymbol{\rho}}(\mathbf{g}^*) < f_{m,\boldsymbol{\rho},h,g}(\mathbf{g}), \forall \mathbf{g}^* \in J(\mathbf{g}), \mathbf{g}, \mathbf{g}^* \in \mathcal{U}_{c,n} \times \mathcal{B}^c \times M_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$.
2. *If* $\mathbf{g} \in \mathcal{T}$, *then* $f_{m,\boldsymbol{\rho},h,g}(\mathbf{g}^*) \leqslant f_{m,\boldsymbol{\rho},h,g}(\mathbf{g}), \forall \mathbf{g}^* \in J(\mathbf{g}), \mathbf{g}, \mathbf{g}^* \in \mathcal{U}_{c,n} \times \mathcal{B}^c \times M_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$.

Proof For any point $(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{W}) \in \mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$, the following relation holds in general:

$$\begin{aligned}
&f_{m,\boldsymbol{\rho},h,g}(J(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{W})) \\
&\quad = f_{m,\boldsymbol{\rho},h,g}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\qquad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W}), \\
&\qquad T_{weight}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\qquad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W}))) \\
&\quad \leqslant f_{m,\boldsymbol{\rho},h,g}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \\
&\qquad T_{matrix}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{W}), \mathcal{W}) \\
&\quad \leqslant f_{m,\boldsymbol{\rho},h,g}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), T_{cent}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}), \mathcal{S}, \mathcal{W}) \\
&\quad \leqslant f_{m,\boldsymbol{\rho},h,g}(T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \mathcal{Z}, \mathcal{S}, \mathcal{W}) \\
&\quad \leqslant f_{m,\boldsymbol{\rho},h,g}(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{W}).
\end{aligned}$$

Equality holds if and only if the following conditions are satisfied:

$$\begin{aligned}
\mathbf{U} &\in T_{memb}(\mathcal{Z}, \mathcal{S}, \mathcal{W}), \\
\mathcal{Z} &= T_{cent}(\mathbf{U}, \mathcal{S}, \mathcal{W}), \\
\mathcal{S} &= T_{matrix}\mathbf{U}, \mathcal{Z}, \mathcal{W}, \\
\mathcal{W} &= T_{weight}(\mathbf{U}, \mathcal{Z}, \mathcal{S}).
\end{aligned}$$

which implies that, $(\mathbf{U}, \mathcal{Z}, \mathcal{S}, \mathcal{W}) \in \mathcal{T}$. Hence, the lemma follows. □

Lemma 3 *The map* T_{memb} *is closed at* $(\mathcal{Z}^{R_1^*}, \mathcal{S}^{R_1^*}, \mathcal{W}^{R_1^*})$ *if* $(\mathbf{U}, \mathcal{Z}^{R_1^*}, \mathcal{S}^{R_1^*}, \mathcal{W}^{R_1^*} \notin \mathcal{T}$ *for some* $\mathbf{U} \in \mathcal{U}_{c,n}$.

Proof We have to prove the following: for all sequence $\{(\mathcal{Z}^{R_1(t)}, \mathcal{S}^{R_1(t)}, \mathcal{W}^{R_1(t)})\}_{t=0}^{\infty}\big\{ \in \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}\big\}$ converging to $(\mathcal{Z}^{R_1^*}, \mathcal{S}^{R_1^*}, \mathcal{W}^{R_1^*})$ and $\{\mathbf{U}^{R_2(t)}\}_{t=0}^{\infty}\Big[\in T_{memb}(\mathcal{Z}^{R_1^{(t)}}, \mathcal{S}^{R_1(t)}, \mathcal{W}^{R_1^{(t)}})\Big]$ converging to $\mathbf{U}^{R_2^*}$; we have that, $\mathbf{U}^{R_2^*} \in T_{memb}(\mathcal{Z}^{R_1^*}, \mathcal{S}^{R_1^*}, \mathcal{W}^{R_1^*})$.

We shall show that the closedness property holds true individually for membership vector corresponding to each of the patterns $\mathbf{x}_i, \forall i = 1, 2, \ldots, n$.

For hard clustering i.e. $m = 1$, we define the following:

$$E_i^{(t)} = \left\{ j \left| dist_g \left(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}} \right) = \min_{1 \leqslant k \leqslant c} dist_g \left(\Sigma_k^{R_1^{(t)}}, \mathbf{z}_k^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_k^{R_1^{(t)}} \right) \right. \right\},$$

$$E_i^{*} = \left\{ j \left| dist_g \left(\Sigma_j^{R_1^{*}}, \mathbf{z}_j^{R_1^{*}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{*}} \right) = \min_{1 \leqslant k \leqslant c} dist_g \left(\Sigma_k^{R_1^{*}}, \mathbf{z}_k^{R_1^{*}}, \mathbf{x}_i, \mathbf{w}_k^{R_1^{*}} \right) \right. \right\}.$$

Using convergence of $\{(\mathbf{z}_j^{R_1^{(t)}}, \Sigma_j^{R_1^{(t)}}, \mathbf{w}_j^{R_1^{(t)}})\}_{t=0}^{\infty}$ to $(\mathbf{z}_j^{R_1^{*}}, \Sigma_j^{R_1^{*}}, \mathbf{w}_j^{R_1^{*}})$ and the continuity of $dist_g$, we can find M_{hard} such that, $\forall t > M_{hard}$, $\max_{j \in E_i^*} dist_g(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}}) < \min_{j \notin E_i^*} dist_g(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}})$, which completes the proof of the lemma.

For fuzzy clustering, i.e. $m > 1$, we define the following:

$$\Psi_i^{(t)} = \left\{ j \left| dist_g \left(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}} \right) = 0 \right. \right\},$$

$$\Psi_i^{*} = \left\{ j \left| dist_g \left(\Sigma_j^{R_1^{*}}, \mathbf{z}_j^{R_1^{*}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{*}} \right) = 0 \right. \right\}.$$

If $|\Psi_i^*| = 0$, using the convergence of $\{(\mathbf{z}_j^{R_1^{(t)}}, \Sigma_j^{R_1^{(t)}}, \mathbf{w}_j^{R_1^{(t)}})\}_{t=0}^{\infty}$ to $(\mathbf{z}_j^{R_1^{*}}, \Sigma_j^{R_1 *}, \mathbf{z}_j^{R_1^{*}})$ and continuity of the dissimilarity measure, we can find M_{fuz1} such that, $\forall t > M_{fuz1}$, $|\Psi_i^{(t)}| = 0$, implying the Lemma. If $|\Psi_i^*| > 0$, $\forall c > 1$ we can find $c(> 0)$, M_{fuz2} such that, $\forall t > M_{fuz2}$, $\max_{j \in \Psi_i^*} dist_g(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}}) < c \min_{j \notin \Psi_i^*} dist_g(\Sigma_j^{R_1^{(t)}}, \mathbf{z}_j^{R_1^{(t)}}, \mathbf{x}_i, \mathbf{w}_j^{R_1^{(t)}})$, which implies the lemma. □

Lemma 4 *O_{cent} is a continuous function on $\mathbf{U}_{c,n} \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d \times c}$.*

Proof

$$T_{cent} = \left(T_{cent}^1, T_{cent}^2, \ldots, T_{cent}^c\right) : \mathbf{U}_{c,n} \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d \times c} \to \mathcal{B}^c,$$

where, $T_{cent}^j(\mathbf{U}, \mathcal{S}, \mathcal{W}) = \mathbf{z}_j, \forall j = 1, 2, \ldots, c$, such that

$$\sum_{i=1}^{n} (u_{ij})^m \frac{\partial}{\partial \mathbf{z}_j} d_{\Sigma_j \mathbf{w}_j, g}(\mathbf{z}_j, \mathbf{x}_i) = 0.$$

In order to show the continuity of T_{cent}^j, we proceed as follows. We define a function B_j in the following way:

$$B_j : \mathcal{U}_{c,n} \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d \times c} \times \mathcal{B} \to \mathbb{R}, \quad j = 1, 2, \ldots, c;$$

$$B_j(\mathbf{U}, \mathcal{S}, \mathcal{W}, \mathbf{z}_j) = \sum_{i=1}^{n} (u_{ij})^m \frac{\partial}{\partial \mathbf{z}_j} d_{\Sigma_j \mathbf{w}_j, g}(\mathbf{z}_j, \mathbf{x}_i).$$

From the very definition of T_{cent}^j, the set of zeroes of B_j can be written in the following form:

$$\mathcal{L}_j = \left\{ (\mathbf{U}, \mathcal{S}, \mathcal{W}, T_{cent}^j(\mathbf{U}, \mathcal{S}, \mathcal{W})) \right\}.$$

As the function B_j is a continuous real valued function, the set of zeroes of B_j is a closed set. Now, from the very form of $\mathcal{L}_j$, we see that it is also the graph of the function T_{cent}^j.

Now, we apply the closed graph theorem (Fitzpatrick 2006) to prove the continuity of T^{j}_{cent}.

Closed graph theorem (Munkres 2000, p. 171) Define the graph of a function $T : \mathcal{P} \rightarrow \mathcal{Y}$ to be the set $\{((x, y) \in \mathcal{P} \times \mathcal{Y} \mid T(x) = y)\}$. If $\mathcal{P}$ is a topological space and $\mathcal{Y}$ is a compact Hausdorff space, then the graph of T is closed if and only if T is continuous.

Using the fact that $\mathcal{B}$ is a compact set, from closed graph theorem it follows that T_{cent} is continuous. □

Lemma 5 *O_{matrix} is a continuous function on $\mathbf{U}_{c,n} \times \mathcal{B}_c \times \mathcal{H}^{d \times c}$.*

Proof Follows from the proof of Lemma 4. □

Lemma 6 *O_{weight} is a continuous function on $\mathbf{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}}$.*

Proof Follows from the proof of Lemma 4. □

Lemma 7 *The map J is closed at $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2})$ if $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2}) \notin \mathcal{T}$.*

Proof From the very definition of J, we have the following:

$$J = O_{weight} \circ O_{matrix} \circ O_{cent} \circ O_{memb}.$$

Now, O_{memb} is closed at $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2})$ if $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2}) \notin \mathcal{T}$ (Lemma 3), O_{cent}, O_{matrix}, and O_{weight} are continuous in their respective domain (from Lemma 4, 5, and 6 respectively). Now, composition of a closed map (at a particular point) with a continuous map, is again continuous. Hence, we have that J is closed at $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2})$ if $(\mathbf{U}^{R_2}, \mathcal{Z}^{R_2}, \mathcal{S}^{R_2}, \mathcal{W}^{R_2}) \notin \mathcal{T}$. □

Theorem 8 *$\forall (\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}) \in \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d \times c}$, the sequence, $\{J(T_{memb}(\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}), \mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)})\}_{t=1}^{\infty}$ either terminates at a point in $\mathcal{T}$ [as defined in (13)] or has a subsequence that converges to a point in $\mathcal{T}$.*

Proof We begin by restating the Zangwill's global convergence theorem from which the current proof is derived.

Zangwill's global convergence theorem (Zangwill 1969) Let $\mathcal{R}$ be a set of minimizers of a continuous objective function Ω on $\mathcal{E}$. Let $A : \mathcal{E} \rightarrow \mathcal{E}$ be a point-to-set map which determines an algorithm that given a point $\mathbf{s}_0 \in \mathcal{E}$, generates a sequence $\{\mathbf{s}_t\}_{t=0}^{\infty}$ through the iteration $\mathbf{s}_{t+1} \in A(\mathbf{s}_t)$. We further assume

1. The sequence $\{\mathbf{s}_t\}_{t=0}^{\infty} \in \mathcal{C} \subseteq \mathcal{E}$, where $\mathcal{C}$ is a compact set.
2. The continuous objective function Ω on $\mathcal{E}$ satisfies the following:
   (a) If $\mathbf{s} \notin \mathcal{R}$, then $\Omega(\mathbf{s}^{'}) < \Omega(\mathbf{s}), \forall \mathbf{s}^{'} \in A(\mathbf{s})$,
   (b) If $\mathbf{s} \in \mathcal{R}$, then $\Omega(\mathbf{s}^{'}) \leqslant \Omega(\mathbf{s}), \forall \mathbf{s}^{'} \in A(\mathbf{s})$.
3. The map A is closed at $\mathbf{s}$ if $\mathbf{s} \notin \mathcal{R}$ (if A is actually a point-to-point map instead of a point-to-set map, condition (3) of the theorem turns out to be simply the continuity of A.)

Then the limit of any convergent subsequence of $\{\mathbf{s}_t\}_{t=0}^{\infty}$ is in $\mathcal{R}$.
We take A to be J, $\mathcal{E}$ to be $\mathcal{U}_{c,n} \times \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d \times c}$; $\mathbf{s}_0$ to be $(\mathbf{U}^{(0)}, \mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)})$; $\mathcal{C}$ to be the whole of $\mathcal{H}$ (compactness of $\mathcal{H}$ is guaranteed by Lemma 1); Ω to be $f_{m,\boldsymbol{\rho},h,g}$ (being sum of continuous functions, $f_{m,\boldsymbol{\rho},h,g}$ is a continuous function), R to be $\mathcal{T}$ (13)

(Lemma 2 justifies the choice). By Lemma 7, J is closed on this particular choice of $\mathcal{H}$. Thus, from Zangwill's convergence theorem, the limit of any convergent subsequence of $\{\mathbf{U}^{(t)}, \mathcal{Z}^{(t)}, \mathcal{S}^{(t)}, \mathcal{W}^{(t)}\}_{t=0}^{\infty}$ has a limit in $\mathcal{T}$. Next $\forall(\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}) \in \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$, we consider the sequence $\{J^{(t)}(T_{memb}(\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}), \mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)})\}_{t=0}^{\infty}$ which is contained in the compact set given by $\mathcal{H}$. Hence by Bolzano–Weierstrass Theorem (Olmsted 1961) it has a convergent subsequence. These statements imply that the sequence given by $\{J^{(t)}(T_{memb}(\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}), \mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)})\}_{t=0}^{\infty}$, $\forall(\mathcal{Z}^{(0)}, \mathcal{S}^{(0)}, \mathcal{W}^{(0)}) \in \mathcal{B}^c \times \mathcal{M}_{d,\boldsymbol{\rho}} \times \mathcal{H}^{d\times c}$ either terminates at a point in $\mathcal{T}$ given by Eq. (13) or has a subsequence that converges to a point in $\mathcal{T}$. □

Theorem 8 provides us with a complete convergence result of the newly developed IPINCWD-based automated feature-weighted clustering algorithms. This particular class of clustering algorithms and the squared Euclidean distance based clustering algorithms share the similar type of global convergence characteristics.

5 Relationship with the existing feature weighting schemes and clustering algorithms

In this section, we discuss how the proposed feature-weighted clustering algorithm with IPINCWD is related to the existing clustering algorithms and feature weighting schemes. We start by presenting a comparative discussion on convergence analysis of IPINCWD-based clustering algorithms and that of the classical FCM with squared Euclidean distance.

5.1 A comparative discussion on the convergence analysis of IPINCWD-based clustering algorithm

The fundamental difference between the convergence analysis of FCM with squared Euclidean distance, with the general IPINCWD-based automated feature-weighted clustering, hinges on the following fact. In case of the feature-weighted clustering algorithm with IPINCWD measures, as far as the updating rules for the cluster representatives, the norm inducing matrices, and the feature weights corresponding to each cluster are concerned, in spite of knowing the existence of a unique upgradation rule, we do not know the continuity of the upgradation rule, which plays a key role in proving the convergence result of FCM with squared Euclidean distance. Hence, given the convergence analysis in Sect. 3, the convergence analysis corresponding to clustering algorithm with IPINCWD boils down to proving the closedness of the clustering operator J which (by Definition 6) is essentially equivalent to proving the continuity of O_{matrix}, O_{cent}, and O_{weight}. In this article, we develop a novel proof of the fact that even in the absence of a closed form upgradation rule of the cluster representative, inner product inducing norms, feature weights corresponding to each of the clusters, a unique updating rule exists which is also continuous. This enables us to perform the convergence analysis for a much broader class of IPINCWD-based automated feature-weighted clustering algorithms.

5.2 Relation with Gustafson–Kessel like algorithms

The conventional fuzzy covariance matrix based GK algorithm (Gustafson and Kessel 1978) can be obtained as a special case of the proposed algorithm, with specific choices of h and g (Definition 1). This is possible if we take h to be identity and consider g to be identically 1.

This choice of g removes the importance of feature weighting. In that case, weight updating becomes meaningless. If we choose h to be identity and consider a non-constant g, a feature-weighted version of the conventional Gustafson–Kessel algorithm is obtained as a special case of the proposed general class of automated feature-weighted IPINCWD-based clustering algorithm. This particular generalized and novel feature weighting scheme introduced in the framework of the GK algorithm can also be extended for various similar kind of clustering algorithms (Liu et al. 2007a, b, 2009a, b). Hence, a general feature weighting scheme can thus be derived to match the dissimilarity measures inspired by IPIN. It can also be integrated with the IPIN-based clustering, with a fixed non-singular matrix (Bezdek 1981) as the inner product inducing matrix.

5.3 Relation with existing feature weighting scheme

If we choose h to be identity and consider $g(x) = x^m$, a special feature-weighted version of the conventional GK algorithm (with the conventional weight of the form w_{ij}^m) is obtained. This feature weighting scheme coincides with that corresponding to the weighting scheme introduced in Saha and Das (2015a) for IPIN. Hence, the feature weighting scheme corresponding to IPIN introduced in Saha and Das (2015a) is a special case of the generalized feature weighting scheme introduced in this article.

5.4 Extension in general divergence setup

To the best of our knowledge, the novel feature weighting scheme introduced in this article is the first of its kind. The earlier existing literature in feature weighting (mentioned in Sect. 2), generally deals with a specific choice of the form $w_{ij}^m \; \forall i = i, 2, \ldots, n; \; j = 1, 2, \ldots, c,$, which is just a special case of the weight function g introduced in the article. The theoretical study on automated feature weighting presented in Saha and Das (2015a) corresponding to clustering with separable distance functions, can also be generalized for this broad class of feature weighting scheme presented in this article.

6 Experimental results

In this section, we present a sample performance comparison (on several simulated and real-life datasets) of the proposed algorithm with 5 other pertinent clustering algorithms, just to highlight the usefulness of our proposal.

6.1 Benchmark dataset

Here, we consider a total of 10 datasets of which 8 are synthetic and 2 from the real world. Table 1 provides a brief description of the datasets.

6.2 Performance measures

The performance of the clustering algorithms is evaluated by using fuzzy and hard, both kind of partition-based validity functions. In order to achieve hard partition from the soft partition (where it is required), we assign the point to the cluster with the highest membership degree. In the case of a tie, it is randomly assigned to any of the clusters with equal probability.

Table 1 Summary of used datasets

Data	n	d	c
2elps_1gauss	300	2	3
Face	230	2	4
Spherical 5_2 (Bandyopadhyay and Maulik 2002)	250	2	5
Spherical 6_2 (Bandyopadhyay and Maulik 2002)	300	2	6
st900 (Bandyopadhyay and Pal 2007)	900	2	9
elliptical_10_2 (Bandyopadhyay and Pal 2007)	500	2	10
Step3_blocks	300	3	3
Step60_blocks	600	60	3
Iris Data (Lichman 2013)	150	4	3
Seed Data (Lichman 2013)	210	8	3

Here, n, d, and c stand for the number of data points, features, and actual clusters, respectively

Table 2 Description of the cluster validity functions

	Functional form	Measured property	Optimal partition
$V_{MPC}(\mathbf{U})$	$1-\frac{c}{c-1}\left[1-\left(\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}^2\right)/n\right]$	Fuzziness of the partition	max V_{MPC}
$V_{PE}(\mathbf{U})$	$\left[\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}\ln u_{ij}\right]/n$	Fuzziness of the partition	min V_{PE}
$V_{XB}(\mathbf{U},\mathcal{Z},\mathcal{X})$	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}^2\|\mathbf{x}_i-\mathbf{z}_j\|^2}{n\min_{j\neq k}\|\mathbf{z}_k-\mathbf{z}_j\|^2}$	Geometric compactness	min V_{XB}
$ARI(\mathcal{T},\mathcal{S})$	$\frac{\sum_{i,j}\binom{n_{ij}}{2}-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i\cdot}}{2}+\sum_j\binom{n_{\cdot j}}{2}\right]-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}$	Crisp partition matching	max ARI

Let $\mathcal{T}=\{t_1,t_2,\ldots,t_R\}$ and $\mathcal{S}=\{s_1,s_2,\ldots,s_c\}$ be two valid partitions of the given data. Let $\mathcal{T}$ be the actual partition and $\mathcal{S}$ be the obtained partition in some clustering algorithm, Now, we wish to evaluate the goodness of $\mathcal{S}$. n_{ij} is the number of objects present in both cluster t_i and s_j; $n_{i.}$ is the number of objects present in cluster t_i; $n_{.j}$ is the number of objects present in cluster s_j.

In order to compare the clustering performance of the different algorithms, we use the Modified Partition Coefficient (V_{MPC}) (Dave 1996), Partition Entropy (V_{PE}) (Bezdek 1973), Xie–Beni Function (V_{XB}) (Xie and Beni 1991), and Adjusted Rand Index (ARI) (Yeung and Ruzzo 2001; Hubert and Arabie 1985). Table 2 provides a brief description of the cluster validity functions under consideration.

6.3 Simulation procedures

The simulation results were obtained by using the following computational protocols over all the datasets.

Choice of h and g From the very definition of h and g, it is obvious that mathematically, there are infinitely many candidates for them. Since it is not possible to extensively compare all the possible weighted dissimilarity measures generated by considering various choices of

h and g, we use very simple forms of these functions for the demonstration purpose only. In particular,we choose and fix h to be the identity function ($h(x) = x$) and g to be the function given by $g(x) = x^{\beta}$, $\beta > 1$.

Algorithms under consideration For a comparative analysis, we choose the following standard algorithms: FCM with fuzzy covariance matrix (IPINCD)(A0) (Gustafson and Kessel 1978; Saha and Das 2016a), Proposed FCM with specific choices of h and g mentioned earlier (IPINCWD) (A1), FCM with squared Euclidean Distance (FCM) (A2), FCM based on automated feature variable weighting (WFCM) (A3) (Nazari et al. 2013), k-means type algorithm with squared Euclidean distance (k-means)(A4) (MacQueen et al. 1967), k-means clustering with automated feature weights (w-k-means) (A5) (Huang et al. 2005).

Choice of the exponent of weights in weighted FCM weighted k-means, and β We take integer values of the exponent, vary it from 2 to 10, and consider the one with the best average value of ARI.

In the comparative study regarding different β values, for $\beta = 0$ (if we remove the effect of weights), for the specific choice of h, A1 boils down to conventional FCM with fuzzy covariance matrix (Gustafson and Kessel 1978) i.e. A0.

6.4 Results and discussions

In what follows, we discuss the comparative performance of the algorithms compared over each of the datasets.

2elps_1gauss For this dataset, A1 achieves perfect clustering in all the 30 runs for $\beta > 2$ as is evident from the plot of the points after clustering (Fig. 1a) and the ARI values (Fig. 1e). Algorithms A4 and A5 were far from being grossly accurate, though A2 and A3 managed to maintain good performance in terms of ARI, but on average, the new algorithm outperforms them for most of the values of β. Considering fuzziness of the partitions, the best performance was shown by A3 with $\beta = 2$ (Fig. 1b, c), but A3 with $\beta = 2$ actually performed really poor, in terms of ARI (Fig. 1e). As far as the other values of β are concerned, the values of MPC and PE are almost same (Fig. 1b, c) for A1, A3, and A2. A0 performs the worst (Fig. 1b ,c) in terms of MPC and PE. If the geometric structure of clustering is concerned, the proposed algorithm performs better than A2 and A3 in terms of XB Index for all values of β (refer to Fig. 1d).

Face For this dataset, A1 yields perfect clustering on all the 30 runs for all non-zero values of β under consideration. As far as the ARI values (Fig. 2e) are concerned, A1 outperforms other algorithms by a considerable margin. Among the other algorithms, A3 showed the worst performance in the class for $\beta = 2$. If we take fuzziness in consideration, according to MPC and PE (Fig. 2b, c) A1 shows considerable improvement from A2, A0, and A3 for all $\beta \neq 0$. Though A2 and A1 with $\beta = 0$ algorithm performs slightly better than A1, in terms of the XB Index (Fig. 2d), our algorithm improves over A3 for $\beta \neq 0$.

Spherical 5_2 In this dataset, the proposed algorithm A1 yields almost perfect partitioning in all the 30 runs for all $\beta \neq 0$.The slight deviation from the perfect clustering [according to ARI (Fig. 3e)] is due to the inability to handle overlapping cluster structures. However, A1 performs much better than A0 [MPC and PE (Fig. 3b, c)], A4, and A5 [Minkowski Score and ARI (Fig. 3e, f)]. Due to the presence of perfectly spherical clusters, A2 and A3 performed

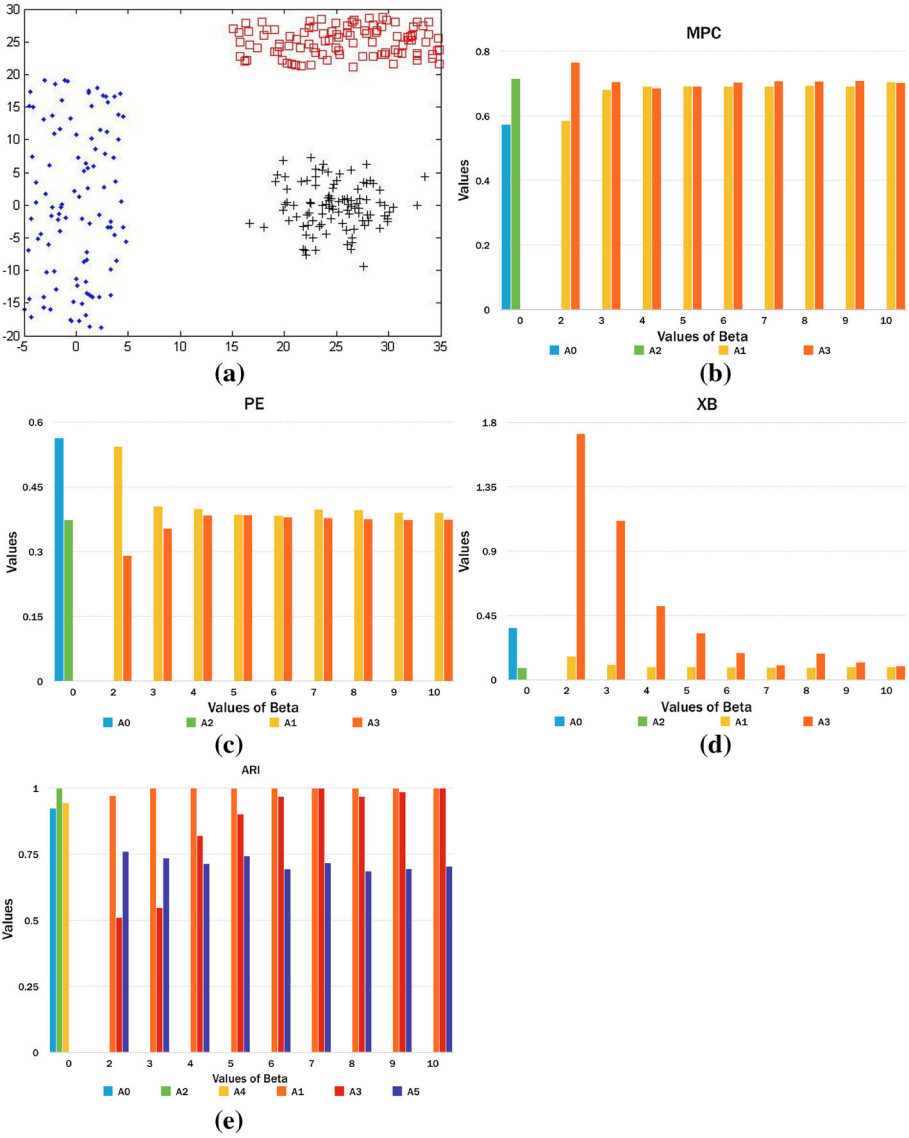

Fig. 1 **a** The best clustering performance of our algorithm on 2elps_1gauss, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

well in this dataset, (in terms of XB Index too) but unlike A0, our algorithm (in spite of using the Mahalanobis distance) is adaptive enough to perform as good as them.

Spherical 6_2 Here we find that A1 is able to provide perfect clustering for all the 30 runs and for all the non-zero values of β under consideration. A2 and A3, due to the presence of well-separated spherical clusters, performs well. The overall performance of A4 and A5 is not satisfactory according to the ARI values (Fig. 4e). As A0 uses the Mahalanobis distance, it fails to recognize the spherical cluster pattern and its performance is not very impressive.

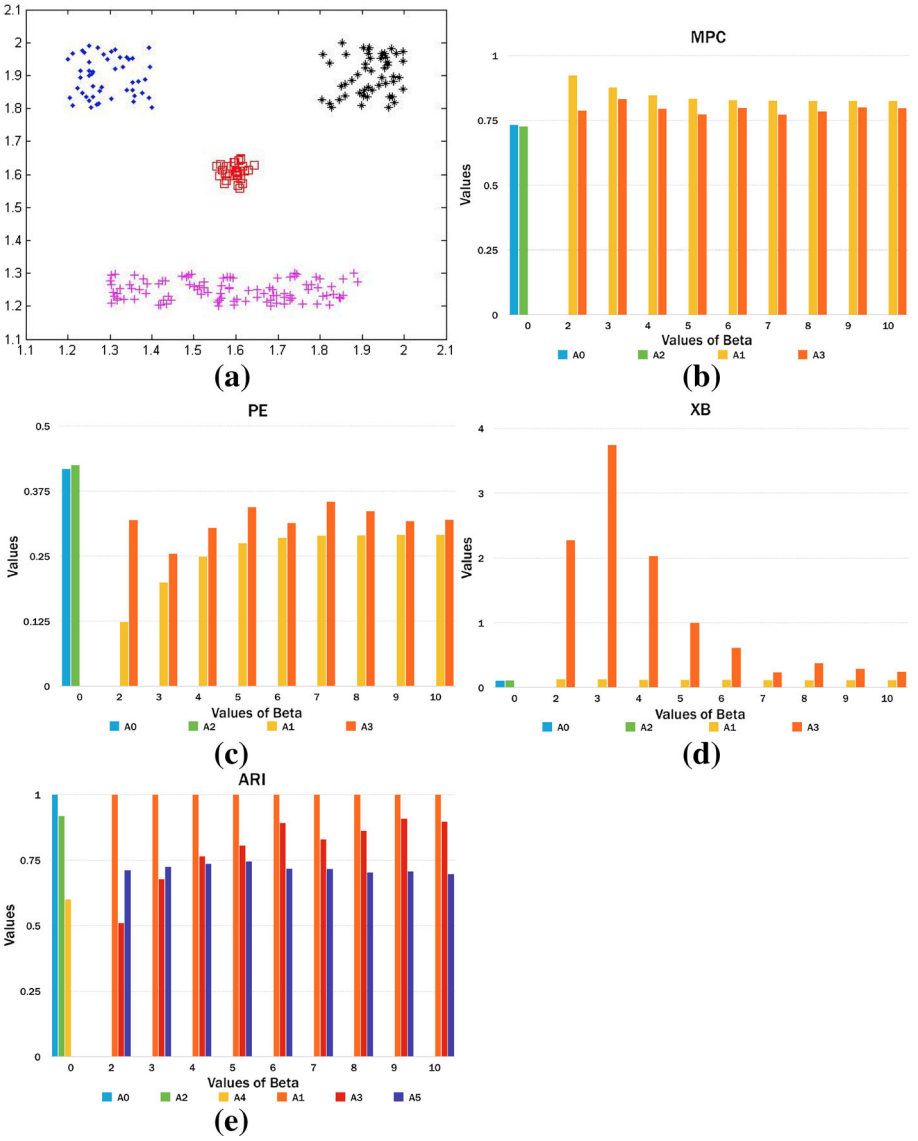

Fig. 2 **a** The best clustering performance of our algorithm on the Face dataset, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

On the other hand, despite using Mahalanobis distance, A1 remains adaptive enough to produce the best overall performance according to all the performance measures (Fig. 4b–e).

st900 In spite of the presence of overlapping clusters, this dataset is partitioned almost perfectly by A1 (Fig. 5a). However, such clusters make the job harder for A2 and A3 as can be observed from Fig. 5e. The proposed algorithm outperforms both A2 and A3 $\forall \beta \neq 0$, by a significant margin. Though MPC and PE for A3 corresponding to $\beta = 2, 3$ appear high (Fig. 5b, c), their clustering performance remain fairly poor (Fig. 5e, f). A0, A4, and A5 did

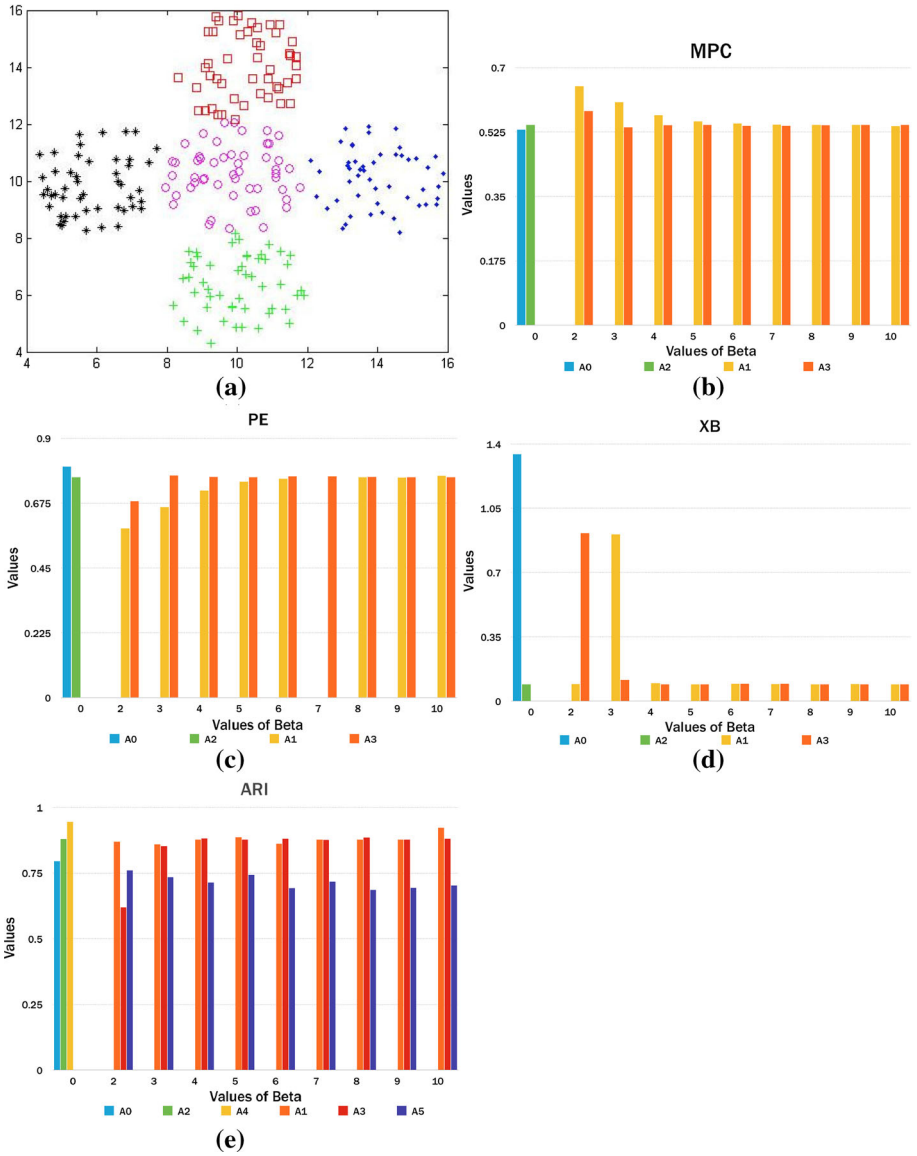

Fig. 3 **a** The best clustering performance of our algorithm on Spherical 5_2, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

not perform satisfactorily, according to the ARI values (Fig. 5e). The best performance of our algorithm with respect to the XB index is better than that of the other algorithms under consideration.

elliptical_10_2 For this dataset, A1 performs perfect clustering in almost all runs for $\beta = 5$. The presence of a single elliptical cluster is enough to affect the performance of A3 and A2 as even their best average performance according to the ARI value remains significantly lower than that of A1 (Fig. 6e). The overall performances of A4 and A5 are not satisfactory

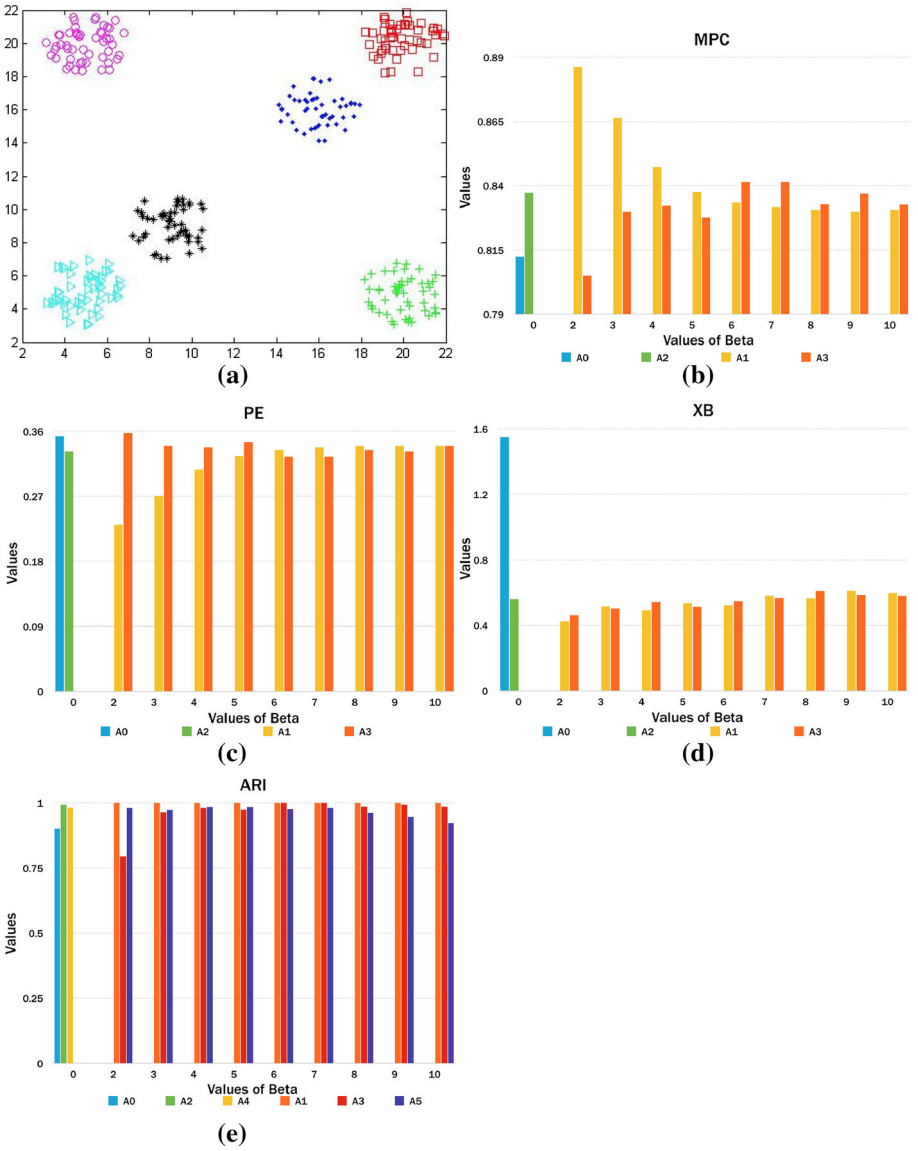

Fig. 4 **a** The best clustering performance of our algorithm on Spherical 6_2, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

according to the ARI values(Fig. 6e). A0 was unable to perform well in all the measures under consideration. Coming to the fuzziness of the partitions, MPC and PE in A3 for $\beta = 2, 3$ are high (Fig. 6b, c), but their clustering performance remains considerably poorer than that of A1 (Fig. 6e). Here also, the best performance of our algorithm in XB Index is better than that of the other algorithms under consideration.

Step3_blocks This dataset has two actual features and one noise feature. Our algorithm is able to detect the noise feature nicely and gives us a near perfect clustering for $\beta > 4$. Accord-

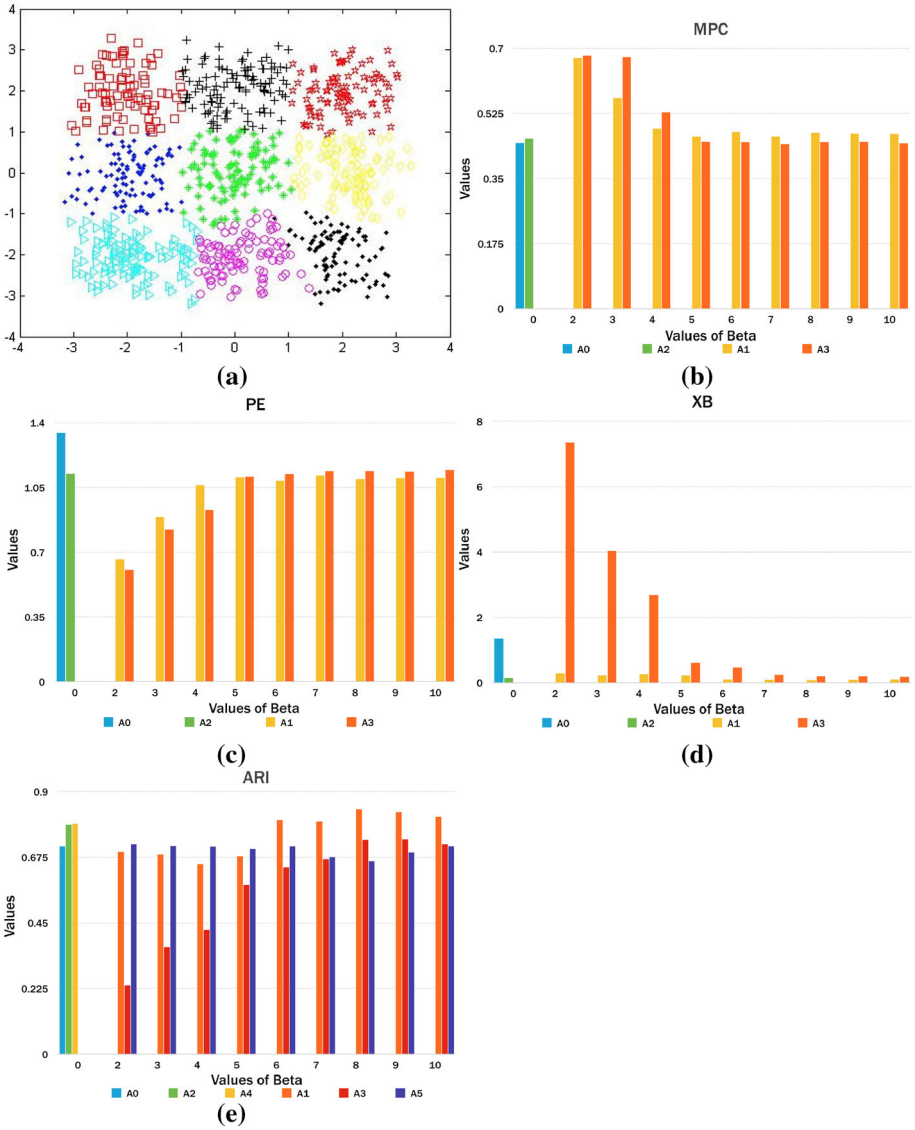

Fig. 5 **a** The best clustering performance of our algorithm on st900, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

ing to the ARI values, our algorithm performs really well for $\beta > 4$ (Fig. 7e). The weight of the noise feature is set to zero. On the other hand, the noise feature, as expected, deteriorates the performances of A2, A0, and A4 significantly. A3 and A5 perform much better than their unweighted counterparts. As far as fuzziness is concerned, A1 with $\beta = 3, 4, 5, 6$ performs much better than the rest of the algorithms (Fig. 7b, c). Amongst them, A1 with $\beta = 5, 6$ also shows greater ARI values (Fig. 7e).

Step60_blocks This 60 dimensional dataset has 30 noise features each following an independent $N(0, 1)$ distribution. There are three clusters, each consisting of 200 points. The

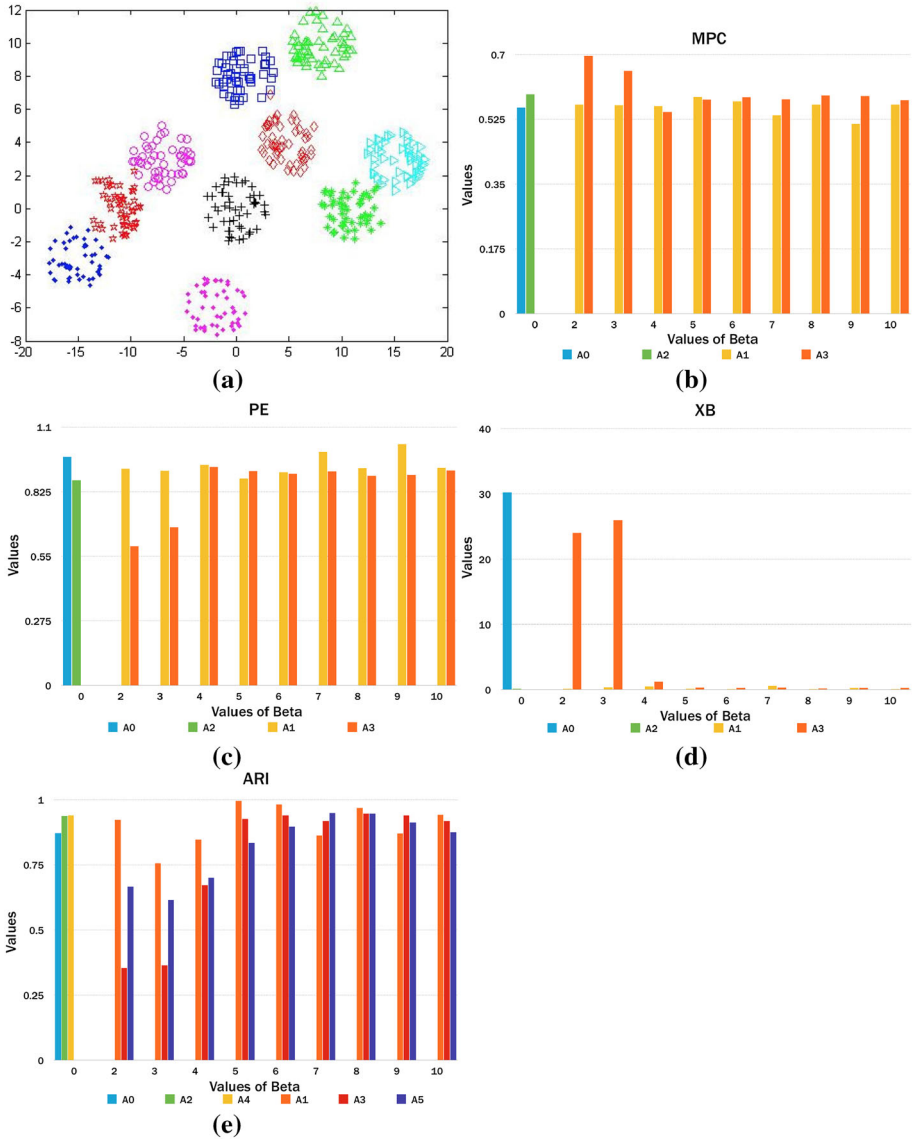

Fig. 6 **a** The best clustering performance of our algorithm on elliptical_10_2, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

30 features, which are instrumental in determining the clustering are simulated from $N(\mathbf{m}_i, \Sigma_i), i = 1, 2, 3$, where $\mathbf{m}_1 = \mathbf{1}$, $\mathbf{m}_2 = 2\mathbf{m}_1$, and $\mathbf{m}_3 = 3\mathbf{m}_1$, $\Sigma_i = \frac{1}{4}\mathbf{I}$. On this dataset, we observe similar clustering performance, as in Step3_blocks. This provides us with an example of a case, where our algorithm works in higher dimension also.

Iris Dataset The parallel coordinate plot of the best clustering performance by A1 shows that in this real world data set also, it is as effective as the synthetic dataset. A1 performed much better than A2, A0, A4, and A5 and was almost as good as A3, which was the top performer

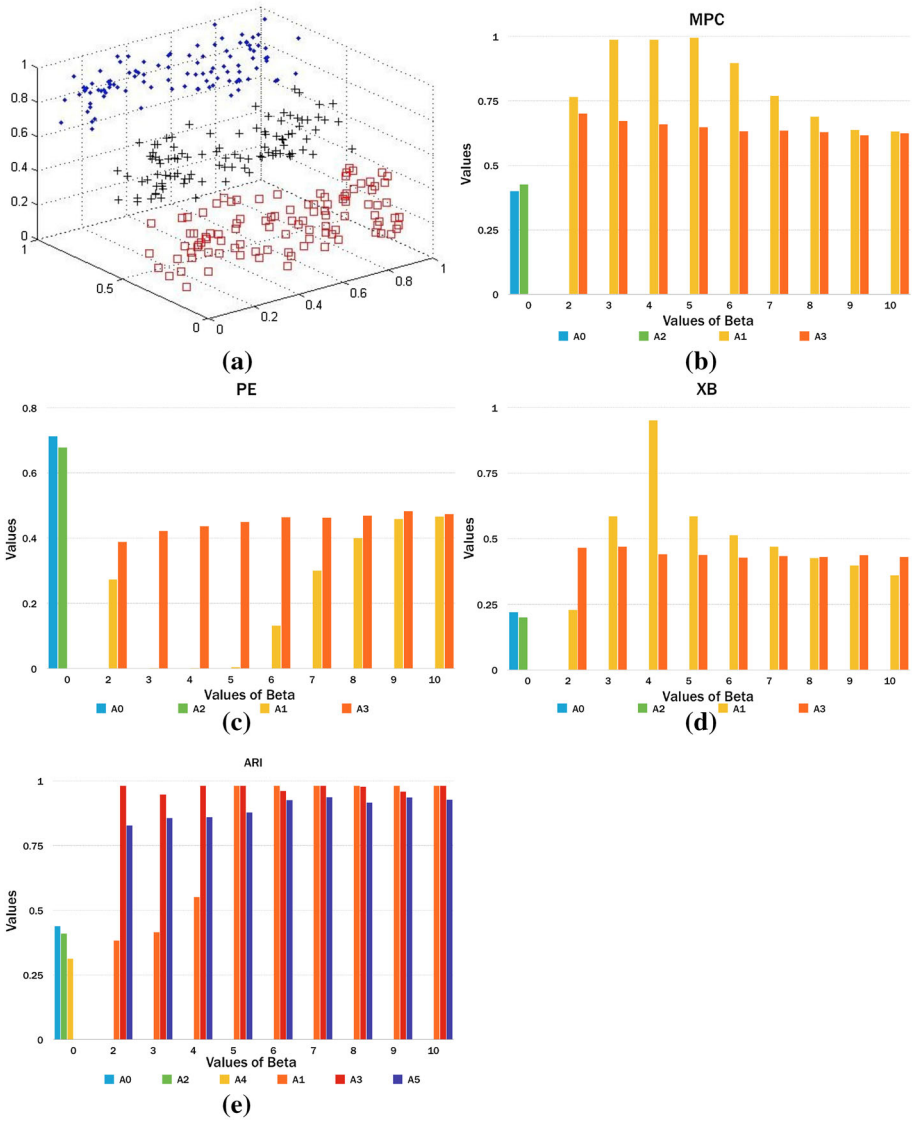

Fig. 7 **a** The best clustering performance of our algorithm on Step3_blocks, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

among the 5 other algorithms, in terms of ARI (Fig. 8e). A1 also performed almost as good as the other 3 algorithms in terms of XB Index, except for a few value of $\beta(2, 3, 4)$. As far as fuzziness (MPC and PE) is concerned, A1 performed better than the rest of the algorithms, for $\beta = 2, 3, 4, 5, 6$ (Fig. 8b, c).

Seed Dataset The parallel coordinate plot of the best clustering performance by A1 shows that it is able to give a more or less accurate clustering on this real world dataset. Here, we see that, in terms of ARI (Fig. 9e), A2 and A0 algorithms, perform the best, but even the best

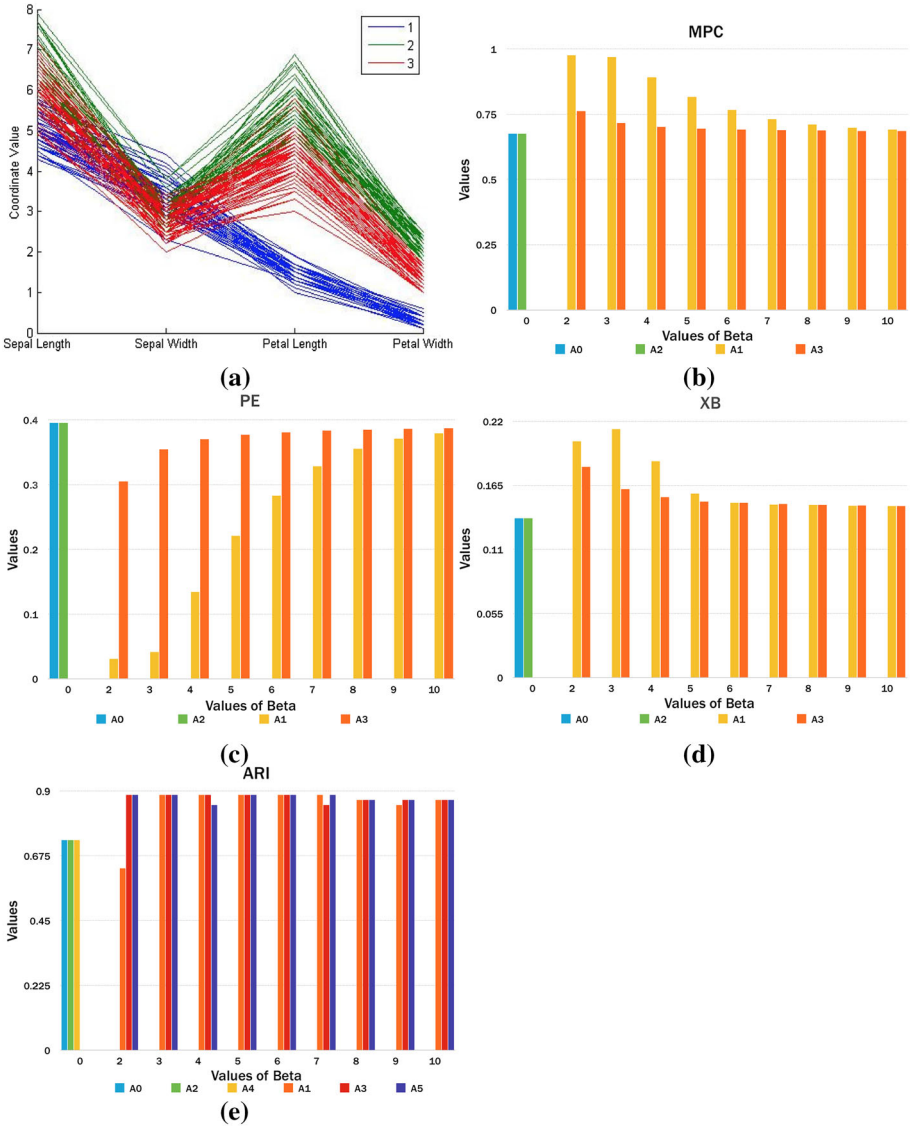

Fig. 8 **a** The best clustering performance of our algorithm on the iris dataset, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

average performance by A3, A4, and A5 are far from the best values of the earlier ones. In the same context, our algorithm for $\beta = 2$ is not only able to perform well but also it outperforms them in terms of the mean value. With respect to the XB index also, A1 is better than A3 for all the β s; and for $\beta = 8, 9, 10$, it is actually pretty close to the best performances shown jointly by A2 and A1 with $\beta = 0$. As far as fuzziness of the obtained partition is concerned (in terms of MPC and PE), A1 $\forall \beta > 1$, performs much better than the other three fuzzy clustering algorithms (Fig. 9b, c).

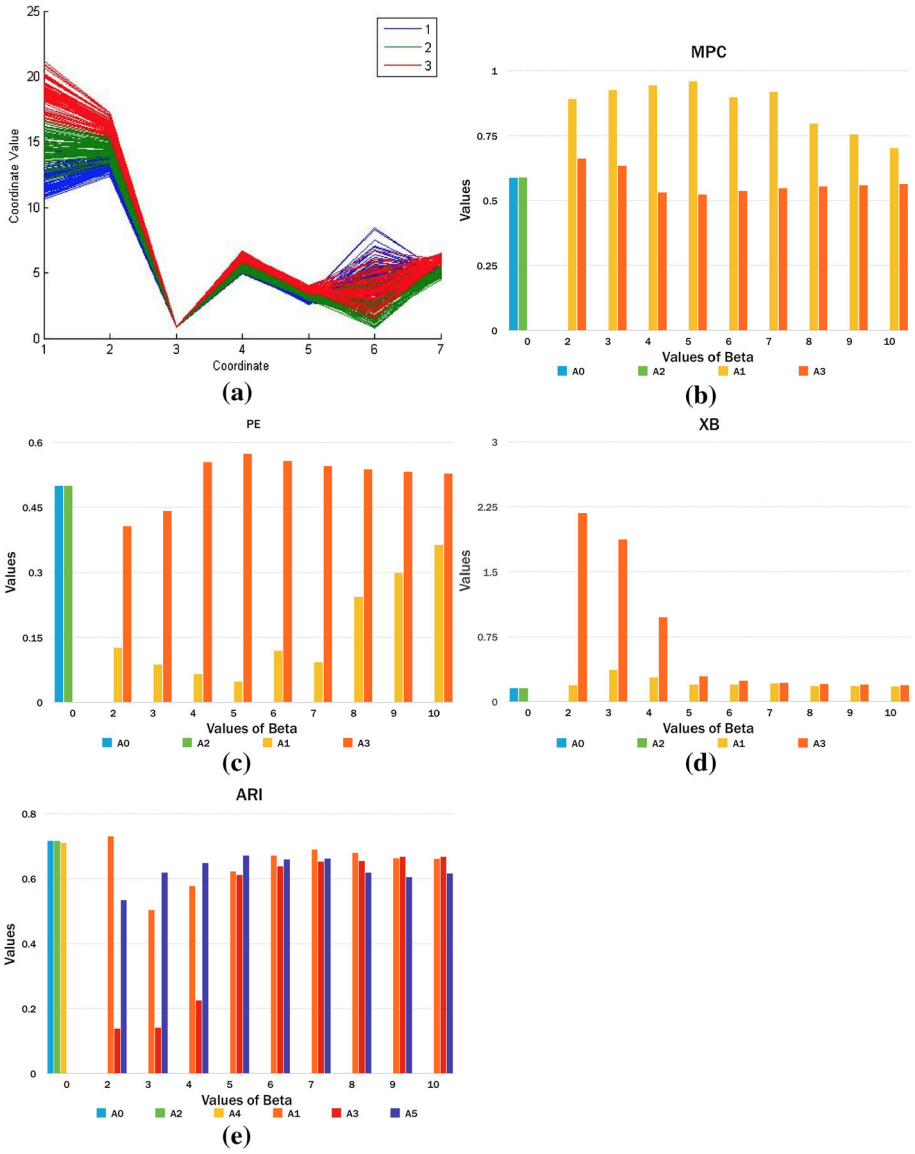

Fig. 9 **a** The best clustering performance of our algorithm on the seed dataset, **b** average value of V_{MPC}, **c** average value of V_{PE}, **d** average value of V_{XB}, **e** average value of Adjusted Rand Index

6.5 Statistical comparison

For each of the 5 performance measures and the 10 datasets under consideration, we carry out the Wilcoxon's rank-sum test (paired) between the best average performances by the algorithms under consideration, to see if we have a statistically significant improvement in our algorithm over the others. Here for A1, A3, and A5, the β producing the best average over the set of 30 runs is considered. In Tables 3, 4, 5 and 6, the P values obtained from the rank-sum test are reported in each column, beneath the main value, and within the parentheses.

Table 3 Average V_{MPC} along with the respective P values within parentheses

Data	WFCM	FCM	IPINCD	IPINCWD
2elps_1gauss	0.764 (1)	0.7138 (1)	0.573 (9E−10)	0.703 (1)
Face	0.8322 (9E−10)	0.7256 (9E−10)	0.7324 (9E−10)	0.9237 (1)
Spherical 5_2	0.5821 (1.9E−06)	0.5444 (9E−10)	0.531 (9E−10)	0.65 (1)
Spherical 6_2	0.8414 (1.6E − 07)	0.8372 (1.6E − 07)	0.8124 (9E−10)	0.8863 (1)
st900	0.681 (0.9787)	0.4573 (9E−10)	0.4451 (9E−10)	0.6732 (1)
elliptical_10_2	0.69 (1)	0.59 (0.87)	0.5571 (9E−10)	0.59 (1)
Step3_blocks	0.701 (9E−10)	0.426 (9E−10)	0.4 (9E−10)	0.9957 (1)
Step60_blocks	0.695 (9E−10)	0.372 (9E−10)	0.321 (9E−10)	0.9857 (1)
Iris Data	0.761 (9E−10)	0.675 (9E−10)	0.675 (9E−10)	0.9762 (1)
Seed Data	0.6613 (9E−10)	0.5886 (9E−10)	0.5879 (9E−10)	0.9586 (1)
Total	7.2	5.93	5.64	8.34
Mean	0.72	0.593	0.564	0.834

6.5.1 Modified partition coefficient

For a specific dataset $\mathcal{R}$, let the median of V_{MPC} corresponding to algorithms A1 ($\beta > 1$), A0, A2, and A3 be denoted by m_{11}, m_{12}, m_{13}, and m_{14} respectively. We perform paired Wilcoxon's rank-sum test on the following hypothesis testing setup:

$$\mathbf{H}_{11,N} : m_{11} = m_{12} \text{ vs. } \mathbf{H}_{11,A} : m_{11} > m_{12}$$
$$\mathbf{H}_{12,N} : m_{11} = m_{13} \text{ vs. } \mathbf{H}_{12,A} : m_{11} > m_{13}$$

and,

$$\mathbf{H}_{13,N} : m_{11} = m_{14} \text{ vs. } \mathbf{H}_{13,A} : m_{11} > m_{14}$$

From the comparison of best average performances of the 3 fuzzy clustering algorithms under consideration with the best average performance of A1 ($\beta > 1$), we see (Table 3) that for dataset 2–4, 7–10 our algorithm shows statistically significant improvement for the considered βs. For datasets 1 and 6, we have a significant improvement from A0. It outperforms both A2 and A0 on dataset 5. The mean of the best average performances over the 9 datasets shows that our algorithm is the best among all the fuzzy clustering methods under consideration according to V_{MPC}.

6.5.2 Partition entropy

For a specific dataset $\mathcal{R}$, let the median of the V_{PE} corresponding to algorithms A1 ($\beta > 1$), A0, A2, and A3 be denoted by m_{21}, m_{22}, m_{23}, and m_{24} respectively. We perform paired Wilcoxon's rank-sum test on the following hypothesis testing setup:

$$\mathbf{H}_{21,N} : m_{21} = m_{22} \text{ versus } \mathbf{H}_{21,A} : m_{21} < m_{22}$$
$$\mathbf{H}_{22,N} : m_{21} = m_{23} \text{ versus } \mathbf{H}_{22,A} : m_{21} < m_{23}$$

Table 4 Average V_{PE} along with the respective P values within parentheses

Data	WFCM	FCM	IPINCD	IPINCWD
2elps_1gauss	0.2905 (1)	0.3733 (1)	0.5623 (9E−10)	0.383 (1)
Face	0.2546 (9E−10)	0.4248(9E-10)	0.4179 (9E−10)	0.1229 (1)
Spherical 5_2	0.681 (1.9E−06)	0.765 (9E−10)	0.8024 (9E−10)	0.587 (1)
Spherical 6_2	0.3253 (1.6E−07)	0.3319 (1.6E−07)	0.3532 (9E−10)	0.231 (1)
st900	0.604 (0.9787)	1.124 (9E−10)	1.345 (9E−10)	0.661 (1)
elliptical_10_2	0.593 (1)	0.874 (0.87)	0.9734 (9E−10)	0.882 (1)
Step3_blocks	0.388 (9E−10)	0.678 (9E−10)	0.712 (9E−10)	0.002 (1)
Step60_blocks	0.396 (9E−10)	0.67 (9E−10)	0.705 (9E−10)	0.002 (1)
Iris Data	0.305 (9E-10)	0.395 (9E−10)	0.395 (9E−10)	0.0308 (1)
Seed Data	0.407 (9E−10)	0.4997 (9E−10)	0.5001 (9E−10)	0.048 (1)
Total	4.25	6.14	6.77	2.95
Mean	0.425	0.614	0.677	0.295

and,

$$\mathbf{H}_{23,N} : m_{21} = m_{24} \text{ versus } \mathbf{H}_{23,A} : m_{21} < m_{24}$$

Here also, we observe the same pattern as that for the case of V_{MPC} (Table 4). A1($\beta > 1$) shows statistically significant amount of improvement from the rest of the fuzzy clustering algorithm and in most of the cases. It outperforms A0, A2, and A3 in 10,7, and 6 datasets respectively. In mean of the best average performances, it appears best among the algorithms considered.

6.5.3 Xie–Beni Index

For a specific dataset $\mathcal{R}$, let the median of the V_{XB} corresponding to algorithms A1 ($\beta > 1$), A1($\beta = 0$), A2, and A3 be denoted by m_{31}, m_{32}, m_{33}, and m_{34} respectively. We perform paired Wilcoxon's rank-sum test on the following hypothesis testing setup:

$$\mathbf{H}_{31,N} : m_{31} = m_{32} \text{ versus } \mathbf{H}_{31,A} : m_{31} < m_{32}$$
$$\mathbf{H}_{32,N} : m_{31} = m_{33} \text{ versus } \mathbf{H}_{32,A} : m_{31} < m_{33}$$

and,

$$\mathbf{H}_{33,N} : m_{31} = m_{34} \text{ versus } \mathbf{H}_{33,A} : m_{31} < m_{34}$$

Here we see (Table 5) that the proposed algorithm appears statistically superior to the rest of the algorithms in several cases. It performs better than the 3 other algorithms on dataset 5, 6; better than A3 and A0 on dataset 1; better than A3 on dataset 2,7, and 10 and better than A2 and A0 on dataset 3, 4. As far as the mean value of the best average performance over all the 9 datasets is concerned, our algorithm proved to be the best one.

6.5.4 Adjusted rand index

For a specific dataset $\mathcal{R}$, let the median of the ARI corresponding to algorithms A1 ($\beta > 1$), A0, A2, A3, A4, and A5 be denoted by $m_{41}, m_{42}, m_{43}, m_{44}, m_{45}$, and m_{46} respectively. We perform paired Wilcoxon's rank-sum test on the following hypothesis testing setup:

Table 5 Average V_{XB} along with the respective P values within parentheses

Data	WFCM	FCM	IPINCD	IPINCWD
2elps_1gauss	0.094 (1)	0.083 (1)	0.3623 (9E−10)	0.084 (1)
Face	0.233 (9E−10)	0.109 (1)	0.1056 (1)	0.1156 (1)
Spherical 5_2	0.0911 (1)	0.0914 (1)	1.345 (9E−10)	0.091 (1)
Spherical 6_2	0.046 (1)	0.561 (1.6E−07)	1.551 (9E−10)	0.043 (1)
st900	0.1977 (0.001)	0.142 (0.0005)	1.345 (9E−10)	0.075 (1)
elliptical_10_2	0.2022 (0.004)	0.15 (0.01)	30.24 (9E−10)	0.0948 (1)
Step3_blocks	0.427 (9E−10)	0.201 (1)	0.22 (1)	0.22 (1)
Step60_blocks	0.457 (9E−10)	0.213 (1)	0.235 (0.11)	0.21 (1)
Iris Data	0.1473 (1)	0.1369 (1)	0.1369 (1)	0.147 (1)
Seed Data	0.1938 (0.15)	0.1514 (1)	0.154 (1)	0.17 (1)
Total	2.09	1.84	5.695	1.2504
Mean	0.209	0.184	0.5695	0.12504

$$\mathbf{H}_{41,N} : m_{41} = m_{42} \text{ versus } \mathbf{H}_{41,A} : m_{41} > m_{42}$$
$$\mathbf{H}_{42,N} : m_{41} = m_{43} \text{ versus } \mathbf{H}_{42,A} : m_{41} > m_{43}$$
$$\mathbf{H}_{43,N} : m_{41} = m_{44} \text{ versus } \mathbf{H}_{43,A} : m_{41} > m_{44}$$
$$\mathbf{H}_{44,N} : m_{41} = m_{45} \text{ versus } \mathbf{H}_{44,A} : m_{41} > m_{45}$$

and,

$$\mathbf{H}_{45,N} : m_{41} = m_{46} \text{ versus } \mathbf{H}_{45,A} : m_{41} > m_{46}$$

From Table 6, we observe that our algorithm achieves best average ARI value in all the datasets, along with an average value of 0.935.

7 An estimation of the runtime complexity

The computational complexity of the proposed algorithm depends on the choice of g and h. Hence, it is impossible to find a general expression for the asymptotic complexity of the proposed algorithm. Here we develop a theoretical expression of the computational overhead of the proposed clustering algorithm with the specific choice of h and g mentioned in Sect. 6.3.

Membership matrix upgradation Here we observe that independent of the choices of h and g, assuming that the function evaluation is of O(1), the computational complexity of the membership matrix update is $\mathrm{O}(nc^3d^2)$. The complexity of the membership updating rule can be obtained from the fact that, there are cd many coordinates of the cluster representatives and corresponding weights. Under the assumption, that the weights corresponding to each of the coordinates are equal for each of the clusters, the computational cost will be $O(nc^2d^2)$.

Cluster representative upgradation With this specific choice of h, we have a closed form upgradation rule of the cluster representatives (which is nothing but the weighted average of the points, obtained by setting the derivative to zero and utilizing the inner product structure).

Table 6 Average ARI along with the respective P values (within parentheses)

Data	FCM	WFCM	k-means	w-k-means	IPINCD	IPINCWD
2elps_1gauss	1 (1)	1 (1)	0.944 (1.6E−10)	0.759 (9E−10)	0.9231 (1.6E−10)	1 (1)
Face	0.918 (0.001)	0.9076 (1.6E−10)	0.6 (9E−10)	0.744 (9E−10)	1 (1)	1 (1)
Spherical 5_2	0.8793 (0.07)	0.8814 (0.1)	0.944 (1)	0.759 (1.6E−10)	0.7956 (1.6E−10)	0.9225 (1)
Spherical 6_2	0.9924 (0.01)	1 (1)	0.982 (0.05)	0.984 (0.05)	0.9004 (1.6E−5)	1 (1)
st900	0.7864 (1.6E−10)	0.7343 (9E−10)	0.79 (1.6E−10)	0.72 (9E−10)	0.713 (9E−10)	0.84 (1)
elliptical_10_2	0.938 (0.005)	0.947 (0.005)	0.94 (0.005)	0.95 (0.005)	0.872 (9E−10)	0.9955 (1)
Step3_blocks	0.41 (1)	0.98 (1)	0.312 (9E−10)	0.937 (0.15)	0.438 (9E−10)	0.98 (1)
Step60_blocks	0.367 (9E−10)	0.996 (1)	0.284 (9E−10)	0.942 (0.15)	0.412 (9E−10)	0.996 (1)
Iris Data	0.729 (9E−10)	0.8857 (1)	0.729 (9E−10)	0.8857 (1)	0.729 (9E−10)	0.8857 (1)
Seed Data	0.7167 (0.15)	0.668 (0.005)	0.71 (0.15)	0.671 (0.005)	0.7167 (0.15)	0.7299 (1)
Total	7.74	9	7.24	8.35	6.7	9.35
Mean	0.774	0.9	0.724	0.835	0.67	0.935

Table 7 Asymptotic computational overhead of the algorithms compared

FCM	WFCM	k-means	w-k-means	IPINCD	IPINCWD
$O(nc^2d)$	$O(nc^2d^2)$	$O(ncd)$	$O(ncd^2)$	$O(nc^2d)$	$O(nc^2d^2)$

Hence, in this case, the complexity of the cluster representative updating rule will be exactly same as that of the other algorithms under consideration and will be given by $O(ncd)$.

Inner product inducing matrix upgradation In this scenario, due to the choice of h, we have a closed form upgradation rule corresponding to the inner product inducing matrix. This is obtained by just multiplying the weight matrix (say corresponding to jth cluster) with the matrix $(\mathbf{x}_i - \mathbf{z}_j)$, $i = 1, 2, \ldots, n$ and using the updating rule used in clustering with the fuzzy covariance matrix. The computational overhead of the updating rule, in this case, will be of $O(nc^2d^2)$.

Weight vector upgradation Here the optimization problem with respect to weights is convex and the objective function is differentiable. Hence we use the gradient descent algorithm to solve the problem. Here, for the sake of computational simplicity, we run a fixed number of any iterative algorithm and then use the result as a close approximation of the actual optimizer. If we choose to use say r runs for approximating the solution, the complexity of the process will be $O(ncdr)$. Note that we can use sophisticated optimization tools like stochastic gradient descent in order to get a better bound on the computational complexity (based on the user determined error), but that is beyond the purview of the present article.

Hence the computational complexity of the process turns out to be (under the assumption, that the weights corresponding to different vectors do not change over clusters, which is logical as the weighted FCM and k-means also do not change the weights for different clusters), $O(nc^2d^2)$ (r is a constant, not of the order of n, c or d). Table 7 provides a comparative view of the asymptotic complexities of the algorithms under consideration.

Hence, the asymptotic complexity of the proposed algorithm is similar to that of WFCM.

8 Conclusion

In this article, we start with a general class of dissimilarity measures based on IPINs. We introduce a novel feature weighting scheme and define a new class of the IPINCWD measures. Next, we develop the automated feature-weighted versions of the hard and fuzzy clustering algorithms with this class of IPINCWD measures in terms of the Lloyd heuristic and alternative optimization procedure respectively. We undertake a detailed discussion regarding the issue of existence and uniqueness corresponding to the sub-optimization problems that form the basic structure of the proposed clustering algorithms. We demonstrate that for any set of initial points, the sequence generated by the clustering operator converges (or has a converging subsequence) to a point in the set of optimal points.

The theoretical development of the generalized IPINCWD-based clustering algorithms provides us with a flexible class of dissimilarity measures which can be used to design data-specific clustering algorithms. This can perform better clustering in situations like non-spherical clusters with background noise, overlapping clusters, clusters with unequal size and density etc. Some spatial constraints may also be introduced, which may lead to a general class of image segmentation algorithms with a better take on the outliers. Given some dataset, what are the bounds on the expected convergence time of this algorithm? How

can we approximate the updating rule corresponding to the cluster representatives, inner product inducing matrices, and feature weights corresponding to each of the clusters, when a closed form expression is not readily available? Even partial answers to such theoretical questions would have a significant practical impact and deserve further investigations. We wish to continue our future research in this direction.

Acknowledgements We sincerely thank the E-i-C, the handling editor, and the anonymous referees for their constructive comments and suggestions which greatly helped in improving the quality of this article.

Appendix

In this appendix, we further illustrate the problems of the fuzzy covariance matrix based clustering without explicit feature weighting. In order to provide a theoretical closure to the issues, we proceed as follows. To maintain the generality of the whole setup, we treat the data points as random variables and show that the expected value of the "weight" corresponding to the noise variable is more than that corresponding to the predominant variables.

We start with the following framework. Let $Y^{(1)}$, $Y^{(2)}$, $X_j^{(1)}$, $X_j^{(2)}$, for $j = 1, 2, \ldots, p-1$ be independent random variables such that, $X_j^{(1)}$'s are identically distributed random variables with

$$E(X_j^{(1)}) = \frac{1}{\sqrt{2}}, \quad Var(X_j^{(1)}) = \frac{1}{2}, \quad \forall j = 1, 2, \ldots, p-1,$$

$X_j^{(2)}$'s are identically distributed random variables with

$$E(X_j^{(2)}) = -\frac{1}{\sqrt{2}}, \quad Var(X_j^{(2)}) = \frac{1}{2} \quad \forall j = 1, 2, \ldots, p-1,$$

and the distributions of $Y^{(1)}$ and $Y^{(2)}$ follow

$$E(Y^{(1)}) = 0; \;\; Var(Y^{(1)}) = \sigma_1^2; \;\; E(Y^{(2)}) = 0, \;\; Var(Y^{(2)}) = \sigma_2^2 = 2 - \sigma_1^2,$$

where σ_1 is a real constant. We construct a dataset of size $2n$, with the first n being i.i.d. observations from $X^{(1)}$ and the last n being i.i.d. observations from $X^{(2)}$. The random variables $X^{(1)}$, $X^{(2)}$ can be expressed as:

$$X^{(1)} = \left(X_1^{(1)}, X_2^{(1)}, \ldots, X_{p-1}^{(1)}, Y^{(1)}\right)^T,$$

and

$$X^{(2)} = \left(X_1^{(2)}, X_2^{(2)}, \ldots, X_{p-1}^{(2)}, Y^{(2)}\right)^T.$$

Evidently, there are two clusters for $n' = 2n$ points, with the first n points belonging to one cluster and the remaining points belonging to the other. The data is normalized for each column (mean 0, variance 1). Moreover, the pth variable is not helpful in detecting the cluster structure and hence works as a noise variable. The remaining $p-1$ variables are the deterministic variables. Now, we demonstrate that, under any cluster membership assignment, the expected weight corresponding to the first $p-1$ variables is less than that corresponding to the noise variable.

At any stage of the iterative algorithm, let the membership assignment be denoted by U_i (corresponding to the ith data point). Here, we note that, as there are only two clusters, the

membership allocation for each point can be done just by finding the allocation in one of the clusters (as the membership in the other cluster can be obtained by subtracting the membership in this cluster from 1). Hence, U_i can be denoted by the membership corresponding to one fixed cluster. As the first n data points are identically distributed, so is their membership vector. The U_i corresponding to the first n data points and the last n data points are identically distributed (let them be denoted by $U^{(1)}$ and $U^{(2)}$ respectively). Now, we compute the expected value of the weights corresponding to the several variables.

In the aforementioned setting, in a cluster (for the sake of notational simplicity, we just use u_i instead of the more familiar u_{ij} as we are only working with one cluster, i.e. fixed j) under consideration, the cluster representative is given by (Gustafson and Kessel 1978, Eq. (20))

$$\mathbf{z} = \frac{\sum_{i=1}^{n'} u_i^m \mathbf{x}_i}{\sum_{i=1}^{n'} u_i^m}.$$

Next, in order to find out the weight corresponding to the variables, we observe that the covariance matrix will essentially be a diagonal matrix (as we have assumed the variables to be independent). Hence, the weight corresponding to kth variable is inversely proportional to the kth diagonal entry of the following matrix (Gustafson and Kessel 1978, Eq. (32)) (in order to make it a member of the class **M** (refer to Sect. 3.1 of the main article), we have to scale it by the determinant of the matrix (Gustafson and Kessel 1978, Eq. (33)), but that would not change the ordering of the values, as the determinant is a positive number due to the positive definiteness of the matrix):

$$\mathbf{P} = \frac{\sum_{i=1}^{n'} u_i^m \left(\mathbf{x}_i - \mathbf{z}\right)\left(\mathbf{x}_i - \mathbf{z}\right)^T}{\sum_{i=1}^{n'} u_i^m}.$$

Now, we calculate the expected value of the matrix, to show that the expected value of the pth diagonal is higher than that of the remaining diagonal entries. Thus we obtain:

$$\begin{aligned} E(\mathbf{P}) &= \sum_{i=1}^{n} E\left(\frac{{U_i^{(1)}}^m \left(X^{(1)} - Z\right)\left(X^{(1)} - Z\right)^T}{\sum_{i=1}^{2n} {U_i}^m}\right) \\ &\quad + \sum_{i=1}^{n} E\left(\frac{{U_i^{(2)}}^m \left(X^{(2)} - Z\right)\left(X^{(2)} - Z\right)^T}{\sum_{i=1}^{2n} {U_i}^m}\right), \\ \implies E(\mathbf{P}) &= n e_1^m E\left(\left(X^{(1)} - Z\right)\left(X^{(1)} - Z\right)^T\right) \\ &\quad + n e_2^m E\left(\left(X^{(2)} - Z\right)\left(X^{(2)} - Z\right)^T\right), \end{aligned}$$

where,

$$e_1^m = E\left(\frac{{U^{(1)}}^m}{\sum_{i=1}^{2n} {U_i}^m}\right); \quad e_2^m = E\left(\frac{{U^{(2)}}^m}{\sum_{i=1}^{2n} {U_i}^m}\right) = \frac{1}{n} - e_1^m.$$

Now, in order to compute $E\left(\left(X^{(l)} - Z\right)\left(X^{(l)} - Z\right)^T\right), l = 1, 2$, we argue as follows. The variables of $X^{(l)}$ are independent, hence their covariance is zero. Here, we observe that, all the first $p - 1$ variables are i.i.d and are deterministic in nature. The pth variable is the

noise variable. Hence, for comparison between the effective weights of the deterministic and noise variables, we can assume $p = 2$. Under this scenario,

$$E\left(\left(X^{(l)} - Z\right)\left(X^{(l)} - Z\right)^T\right) = E\left(X^{(l)}X^{(l)^T}\right) + E\left(ZZ^T\right) + E\left(X^{(l)}Z^T\right) + E\left(ZX^{(l)^T}\right).$$

For notational simplicity, let the distribution of the ith data point be denoted by $X_i = X^{(1)}$ or $X^{(2)}$ depending on whether $1 \leqslant i \leqslant n$ or $n < i \leqslant 2n$.

Hence,

$$\begin{aligned}
E\left(X^{(l)}X^{(l)^T}\right) &= Var\left(X^{(l)}\right) + E\left(X^{(l)}\right)E\left(X^{(l)}\right)^T \\
&= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \sigma_l^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sigma_l^2 \end{bmatrix}, \\
E\left(ZZ^T\right) &= \sum_{i=1}^{2n}\sum_{j=1}^{2n} E\left(\frac{{U_i}^m}{\sum_{i=1}^{2n}{U_i}^m}\right)E\left(\frac{{U_j}^m}{\sum_{i=1}^{2n}{U_i}^m}\right)E\left(X_i X_j^T\right) \\
&= n^2\left(e_1^m - e_2^m\right)^2\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}, \\
E\left(ZX^{(l)^T}\right)^T &= E\left(X^{(l)}Z^T\right) = \sum_{i=1}^{2n} E\left(\frac{{U_i}^m}{\sum_{i=1}^{2n}{U_i}^m}\right)E\left(X^{(l)}X_i^T\right) \\
&= e_l^m\begin{bmatrix} 1 & 0 \\ 0 & \sigma_l^2 \end{bmatrix} + e_{3-l}^m\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix},
\end{aligned}$$

and

$$E\left(\left(X^{(l)} - Z\right)\left(X^{(l)} - Z\right)^T\right) = \begin{bmatrix} 1 & 0 \\ 0 & \sigma_l^2 \end{bmatrix} + n^2\left(e_1^m - e_2^m\right)^2\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} + 2e_l^m\begin{bmatrix} 1 & 0 \\ 0 & \sigma_l^2 \end{bmatrix} + 2e_{3-l}^m\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}.$$

Now, using the fact that whatever be the value of e_1^m, we can choose σ_1 and σ_2 in such a way that the element in the first diagonal in the expected value of $\mathbf{P}$ will be greater than the element in the pth diagonal.

This indicates that under even a generalized framework, though the use of $\mathbf{M}$ scales the data with respect to the variance, it does not ensure that the noise variable will receive a smaller weight. Even in a normalized setup, the presence of noise can be detrimental to the clustering performance, as it does not assign less weight to the noise variable than that of the deterministic variables.

In order to provide support to our aforementioned claim, we present a simulated example. The setup is as follows:

The data set has 200 points, with first 100 points belonging to one cluster and the remaining 100 points belonging to another cluster. We present the experiment with 3 variables, where the first two variables are deterministic variables and are simulated such that for both of them the first 100 values are i.i.d observations sampled from the distribution with mean $\frac{1}{\sqrt{2}}$ and variance $\frac{1}{2}$; and the last 100 values are i.i.d observations sampled from the distribution with mean $-\frac{1}{\sqrt{2}}$ and variance $\frac{1}{2}$. As far as the noise variable is concerned, the first 100 values are

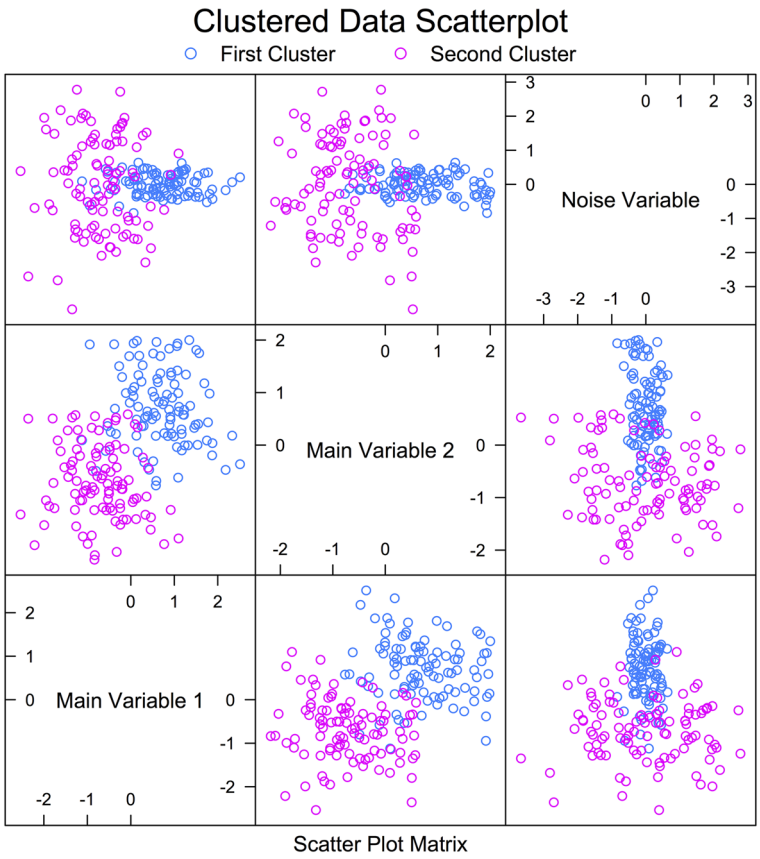

Fig. 10 (Colour figure online) The pairwise scatter plot of the simulated data, with data belonging to separate clusters marked in *different color*

i.i.d observations from a distribution with mean 0 and variance 0.1 and the last 100 values are i.i.d observations from a distribution with mean 0 and variance 1.9. Here, we use Gaussian distribution for all of the aforementioned simulations. We note that the data is normalized by construction. We also get rid of the assumption of the independence of the variables and demonstrate that, in the simulated example, our claim holds even for correlated variables. In Fig. 10, we present a pairwise scattered plot of the clustered data to demonstrate that the third variable is indeed a noise variable.

We use a shifted uniform distribution (with equal variance) to simulate the membership assignment. Then we proceed to calculate the mean and the corresponding **M**. Let the bound on the determinant of **M** be 1 (this is nothing special, it can be any positive real number, as all the terms on the diagonal will be equally divided by the constant, their ordering would not change). The **P** matrix (corresponding to the cluster under consideration) can be expressed as:

$$\mathbf{P} = \begin{bmatrix} 58.28588 & 22.629363 & -1.013130 \\ 22.62936 & 64.189931 & -2.784149 \\ -1.01313 & -2.784149 & 37.443018 \end{bmatrix}.$$

Thus, the corresponding weighting matrix ($\mathbf{M}$) is given by:

$$\mathbf{M} = \left(\frac{1}{1.|\mathbf{P}|}\right)^{\frac{1}{3}} \mathbf{P}^{-1} = \begin{bmatrix} 1.571340513 & -0.55389886 & 0.001330983 \\ -0.553898862 & 1.43075521 & 0.091399243 \\ 0.001330983 & 0.09139924 & 2.118073100 \end{bmatrix}.$$

In the case of $\mathbf{M}$, the last diagonal entry, namely the weight corresponding to the noise variable is greater than that of all the other deterministic variable. Hence, this simulation result experimentally establishes that even in the normalized setup, the noise variable does not necessarily get the smallest weight, rather it may very well get the biggest weight. Hence, the presence of noise variable may significantly deteriorate the clustering performance, even in the normalized setup.

References

Anderberg, M. R. (2014). *Cluster analysis for applications: Probability and mathematical statistics: A series of monographs and textbooks* (Vol. 19). London: Academic Press.

Bandyopadhyay, S., & Maulik, U. (2002). Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern Recognition*, *35*(6), 1197–1208.

Bandyopadhyay, S., & Pal, S. K. (2007). *Classification and learning using genetic algorithms: Applications in bioinformatics and web intelligence*. Berlin: Springer.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman divergences. *The Journal of Machine Learning Research*, *6*, 1705–1749.

Berkhin, P. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping multidimensional data* (pp. 25–71). Berlin: Springer.

Bezdek, J. C. (1973). Cluster validity with fuzzy sets. *Journal of Cybernetics*, *3*(3), 58–73.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Norwell, MA: Kluwer.

Bezdek, J. C., & Hathaway, R. J. (2003). Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, *11*(4), 351–368.

Chaomurilige, Y. J., & Yang, M. S. (2015). Analysis of parameter selection for Gustafson-Kessel fuzzy clustering using Jacobian matrix. *IEEE Transactions on Fuzzy Systems*, *23*(6), 2329–2342.

Dave, R. N. (1996). Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognition Letters*, *17*(6), 613–623.

De Soete, G. (1988). OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting. *Journal of Classification*, *5*(1), 101–104.

DeSarbo, W. S., Carroll, J. D., Clark, L. A., & Green, P. E. (1984). Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *Psychometrika*, *49*(1), 57–78.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, *3*(3), 32–57.

D'Urso, P., Massari, R., De Giovanni, L., & Cappelli, C. (2016). Exponential distance-based fuzzy clustering for interval-valued data. *Fuzzy Optimization and Decision Making*. doi:10.1007/s10700-016-9238-8.

Fitzpatrick, P. (2006). *Advanced calculus* (Vol. 5). Providence: American Mathematical Society.

Gath, I., & Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(7), 773–780.

Golub, G. H., & Van Loan, C. F. (2012). *Matrix computations* (Vol. 3). Baltimore: JHU Press.

Gustafson, D., & Kessel, W. (1978). Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE conference on decision and control including the 17th symposium on adaptive processes* (No. 17, pp. 761–766).

Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(5), 657–668.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Hung, W.-L., Yang, M.-S., & Hwang, C.-M. (2011). Exponential-distance weighted k-means algorithm with spatial constraints for color image segmentation. In *2011 international conference on multimedia and signal processing (CMSP)* (Vol. 1, pp. 131–135). IEEE.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, *31*(3), 264–323.

Keller, A., & Klawonn, F. (2000). Fuzzy clustering with weighting of data variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *8*(06), 735–746.

Klawonn, F., & Höppner, F. (2003). What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In *International symposium on intelligent data analysis* (pp. 254–264). Springer.

Krishnapuram, R., & Kim, J. (1999). A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms. *IEEE Transactions on Fuzzy Systems*, *7*(4), 453–461.

Lichman, M. (2013). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.

Liu, H.-C., Jeng, B.-C., Yih, J.-M., & Yu, Y.-K. (2009a). Fuzzy c-means algorithm based on standard Mahalanobis distances. In *Proceedings of the 2009 international symposium on information processing* (pp. 422–427).

Liu, H.-C., Yih, J.-M., Lin, W.-C., & Wu, D.-B. (2009b). Fuzzy c-means algorithm based on common Mahalanobis distances. *Journal of Multiple-Valued Logic & Soft Computing, 15*, 581–595.

Liu, H.-C., Yih, J.-M., & Liu, S.-W. (2007a). Fuzzy c-means algorithm based on Mahalanobis distances and better initial values. In *Proceedings of the 10th joint conference and 12th international conference on fuzzy theory and technology* (Vol. 1, pp. 1398–1404). Singapore: World Scientific.

Liu, H.-C., Yih, J.-M., Sheu, T.-W., & Liu, S.-W. (2007b). A new fuzzy possibility clustering algorithms based on unsupervised Mahalanobis distances. In *2007 international conference on machine learning and cybernetics* (Vol. 7, pp. 3939–3944). IEEE.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

Lumelsky, V. J. (1982). A combined algorithm for weighting the variables and clustering in the clustering problem. *Pattern Recognition*, *15*(2), 53–60.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297), Oakland, CA, USA.

Makarenkov, V., & Legendre, P. (2001). Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. *Journal of Classification*, *18*(2), 245–271.

Mao, J., & Jain, A. K. (1996). A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks*, *7*(1), 16–29.

Modha, D. S., & Spangler, W. S. (2003). Feature weighting in k-means clustering. *Machine Learning*, *52*(3), 217–237.

Munkres, J. R. (2000). *Topology* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Nazari, M., Shanbehzadeh, J., & Sarrafzadeh, A. (2013). Fuzzy c-means based on automated variable feature weighting. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 25–29).

Olmsted, J. M. H. (1961). *Advanced calculus*. Upper Saddle River, NJ: Prentice Hall.

Ostrovsky, R., Rabani, Y., Schulman, L. J., & Swamy, C. (2012). The effectiveness of Lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, *59*(6), 28.

Saha, A., & Das, S. (2015a). Automated feature weighting in clustering with separable distances and inner product induced norms–A theoretical generalization. *Pattern Recognition Letters*, *63*, 50–58.

Saha, A., & Das, S. (2015b). Categorical fuzzy k-modes clustering with automated feature weight learning. *Neurocomputing*, *166*, 422–435.

Saha, A., & Das, S. (2016a). Geometric divergence based fuzzy clustering with strong resilience to noise features. *Pattern Recognition Letters*, *79*, 60–67.

Saha, A., & Das, S. (2016b). Optimizing cluster structures with inner product induced norm based dissimilarity measures: Theoretical development and convergence analysis. *Information Sciences*, *372*, 796–814.

Selim, S. Z., & Ismail, M. A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(1), 81–87.

Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical taxonomy. The principles and practice of numerical classification*. San Francisco, CA: WH Freeman.

Teboulle, M. (2007). A unified continuous optimization framework for center-based clustering methods. *The Journal of Machine Learning Research*, *8*, 65–102.

Teboulle, M., Berkhin, P., Dhillon, I., Guan, Y., & Kogan, J. (2006). Clustering with entropy-like k-means algorithms. In M. Teboulle, P. Berkhin, I. Dhillon, Y. Guan, & J. Kogan (Eds.), *Grouping multidimensional data* (pp. 127–160). Berlin: Springer.

Wölfel, M., & Ekenel, H. K. (2005). Feature weighted Mahalanobis distance: Improved robustness for Gaussian classifiers. In *2005 13th European signal processing conference* (pp. 1–4). IEEE.

Wu, J., Xiong, H., Liu, C., & Chen, J. (2012). A generalization of distance functions for fuzzy c-means clustering with centroids of arithmetic means. *IEEE Transactions on Fuzzy Systems*, *20*(3), 557–571.

Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(8), 841–847.

Yeung, K. Y., & Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper "An empirical study on principal component analysis for clustering gene expression data". *Bioinformatics*, *17*(9), 763–774.

Zangwill, W. I. (1969). *Nonlinear programming: A unified approach* (Vol. 196). Englewood Cliffs, NJ: Prentice-Hall.