Applied Soft Computing 113 (2021) 107922

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy clustering algorithms with distance metric learning and entropy regularization

Sara I.R. Rodríguez, Francisco de A.T. de Carvalho *

Centro de Informática, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária, CEP 50740-560, Recife (PE), Brazil

ARTICLE INFO

Article history:
Received 12 July 2019
Received in revised form 26 August 2021
Accepted 19 September 2021
Available online 29 September 2021

Keywords:
Fuzzy clustering
Prototype-based clustering
Distance metric learning
Maximum-entropy regularization

ABSTRACT

Clustering has been used in various fields, such as image processing, data mining, pattern recognition, and statistical analysis. Generally, clustering algorithms consider all variables equally relevant or not correlated. Nevertheless, the pattern of data samples in the multidimensional space can be geometrically complicated, e.g., clusters may exist in different subsets of features. In this regard, new soft subspace clustering algorithms have been proposed, in which the correlation and relevance of variables are considered to improve their performance. Since regularization-based methods are robust for initializations, the approaches proposed introduce an entropy regularization term for controlling the membership degree of the objects. Such regularizations are popular due to high performance in large-scale data clustering and low computational complexity. These three-step iterative algorithms provide a fuzzy partition, a representative for each cluster, and the relevance weight of the variables or their correlation by minimizing a suitable objective function. Several experiments on synthetic and real datasets, including their application to the segmentation of noisy image textures, demonstrate the usefulness of the proposed clustering methods.

1. Introduction

Clustering refers to a procedure that groups similar objects together while separating dissimilar ones apart [1,2]. Clustering algorithms are an efficient tool for image processing, data mining, pattern recognition, and statistical analysis [3–5]. The most popular clustering algorithms provide hierarchical and partitioning structures. Hierarchical methods deliver an output represented by a hierarchical structure of groups known as dendrogram, i.e., a nested sequence of partitions of the input data. However, partitioning approaches divide the input data into a fixed number of clusters based on either distance or density criteria computed on the dataset, typically by optimizing an objective function. An advantage of partitioning methods is their ability to manipulate large datasets, since the construction of the dendrogram by the hierarchical approach may be computationally impractical in some applications.

Partitioning methods perform mainly in two different ways: hard and fuzzy. In hard or crisp approaches, clusters are disjoint and non-overlapping [6,7]. In this case, any object may belong to one and only one group. On the other hand, in fuzzy clustering, an object belongs to all clusters with a specific membership degree, a characteristic that can be essential for discovering intricate relations between a given data object and all groups [8]. Fuzzy clustering methods are peculiarly effective when data points belong to multiple clusters. Ref. [9] used the Fuzzy C-Means (FCM) clustering based on an integration of psychological and physiological data for therapeutic music design. Mahmoudi et al. [10] compare the spread rate of Covid-19 in high-risk countries using the fuzzy clustering method. Refs. [11,12] include a review of fuzzy clustering and its applications.

More recently in the scope of fuzzy techniques, much attention has been directed to maximum entropy clustering algorithms for complex data, due to their low sensitivity to initialization and high clustering performance on large-scale data compared with Fuzzy C-Mean models [13,14]. The maximum entropy approaches belong to an optimization problem with conditional constraints, in which the maximum entropy model is identified as an optimal solution among all models that meet the restrictions. Such models offer a distinct physical meaning and possess well-defined mathematical characteristics, making them easy to understand [15–17]. In this case, fuzziness is represented by a weighting factor that multiplies the regularization term added to the clustering criterion. In this framework, the regularization function measures the overall fuzziness of the obtained clustering pattern.

Traditional methods of clustering assume that the variables are uncorrelated and equally relevant to the clustering process.

* Corresponding author.
E-mail addresses: sirr@cin.ufpe.br (S.I.R. Rodríguez), fatc@cin.ufpe.br (F.d.A.T. de Carvalho).

https://doi.org/10.1016/j.asoc.2021.107922
1568-4946/

The Euclidean distance as a similarity measure restricts conventional algorithms to datasets with hyper-spherical clusters and linearly separable characteristics. However, in real problems, mainly in high-dimensional ones, some variables can be correlated. Therefore, for any given pair of neighboring items within the same cluster, the objects may be separated from each other in a few dimensions of the high-dimensional space. Soft subspace clustering (SSC) techniques perform clustering in high-dimensional spaces by assigning a weight to each dimension to measure the contribution of individual dimensions when forming a particular cluster [18]. The advantage of this approach is that the clustering methods can recognize groups of different shapes and sizes.

Several SSC methods have been proposed [19–21], for example, Wang et al. [22] introduced a technique applying a weighted Euclidean distance in the FCM formulation to improve the clustering performance. Later, Deng et al. [23] developed an enhanced entropy-weighting subspace clustering algorithm for high dimensional gene expression data clustering by simultaneously integrating the within-cluster and between-cluster information. Ref. [24] shows a fuzzy co-clustering approach using a multi-dimensional distance function as the dissimilarity measure and entropy as the regularization term for image segmentation. Rodríguez and de Carvalho [25] also presented a soft subspace clustering algorithm based on the Euclidean distance and entropy regularization. Aiming to simplify the presentation and discussion of the experimental results of Section 4, hereafter, we adopt the notation EFCM-LP2 for the clustering algorithm of Ref. [25].

In prototype-based clustering algorithms, the squared Euclidean distance is generally used to compare objects and prototypes. However, theoretical studies indicate that City-Block distance-based methods are more robust to cope with outliers in the dataset. Addressing such, Rodríguez and de Carvalho [26] introduced a soft subspace clustering algorithm based on the City-block distance and entropy regularization, in which each cluster has a different set of relevant variables. Despite its usefulness, the method does not consider the correlation between variables, an issue that can be overcome with the use of the Mahalanobis distance metric as dissimilarity measure by considering the covariance of the variables. Besides, in some situations, local methods such as Ref. [26] can be improved by introducing global adaptive distances. In such approaches, the set of relevant variables is the same for all clusters and can be helpful when the internal dispersion of the groups is almost the same.

This paper extends Refs. [25,26] by proposing new soft subspace clustering methods with entropy regularization. Unlike previous works, new global and local dissimilarity functions based on the Euclidean, Mahalanobis, and City-Block distances are introduced. These new approaches consider the relevance and correlation of the variables during the clustering process. The proposed algorithms optimize an objective function to obtain each cluster prototype, the relevance weights of the variables, and the best fuzzy partition until a stopping criterion is satisfied.

The following main contributions were achieved:

- Proposal of new soft subspace clustering algorithms with entropy regularization, recognizing clusters of different shapes and sizes. The relevance of each dimension is calculated by automatically assigning different weights to the dimensions of the clusters embedded in subspaces. Such approaches allow locating groups in different subspaces of the same dataset.
- Introduction of several adaptive distances that change at each algorithm iteration and may differ from one cluster to another. They are based on the Euclidean distance, one of the most used in the literature, and the Mahalanobis distance to consider covariance between the variables. This last choice is defined by a positive definite symmetric matrix that may differ from one cluster to another. Additionally, methods based on City-Block distance were proposed because of their better perform in a noisy data environment.
- Local (a different set of relevant variables per cluster) and global (the same set of relevant variables for all groups) adaptive distances were considered, since local approaches may not be appropriate to some situations, e.g., when the internal dispersion of the clusters are almost the same. The relevance weights of the variables are computed by considering two types of restrictions. In the first, the sum of the weights of the variables must be equal to one, whereas in the second, the product of the relevance of the variables is equal to one. The latter constraint has the advantage of requiring the adjustment of fewer hyperparameters, such as the one that controls the membership degree of objects.
- Our proposals add an entropy term that regularizes the clustering results during the optimization process to satisfy all the constraint conditions. In this case, the method with the maximum entropy will be identified as the optimal solution among all the methods meeting the restrictions. Due to their simple implementation and low computational complexity, regularization-based methods can be used in large and high-dimension data clustering. Furthermore, such approaches are less sensitive to initialization and offer a distinct physical meaning and well-defined mathematical characteristics, making them easy to understand.
- Finally, this paper provides an algebraic solution to compute the minimizers of the objective functions and a detailed derivation for all constraints and metrics. Considering the absence of algebraic solution to obtain the prototype minimizer in City-Block distance-based approaches, we present an algorithmic solution, in addition to analyzing the convergence properties of the proposed algorithms.

The paper is organized as follows. Section 2 reviews two works closely related to the proposed approaches. Section 3 presents the proposed fuzzy clustering algorithms based on the adaptive Euclidean, Mahalanobis, City-block distances, and entropy regularization. Experiments and results are reported in Section 4. Finally, conclusions are drawn in Section 5.

2. Related work

In this section, we briefly review two algorithms closely related to our work. Let $E = \{e_1, \ldots, e_P\}$ be a set of P objects. Each object $e_i\,(1 \leq i \leq P)$ is described by the vector of variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{iV})$, with $x_{ij} \in \mathbb{R}\,(1 \leq \mathrm{j} \leq \mathrm{V})$. Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_P\}$ be the dataset. We assume that the fuzzy clustering algorithms considered in this paper provide:

- A fuzzy partition represented by the matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_P) = (u_{ik})_{\substack{1\leq i\leq P\\ 1\leq k\leq C}}$, where u_{ik} is the membership degree of the object e_i into the fuzzy cluster k and $\mathbf{u}_i = (u_{i1}, \ldots, u_{iC})$;
- A matrix $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_C) = (g_{kj})_{\substack{1\leq k\leq C\\ 1\leq j\leq V}}$, where the component $\mathbf{g}_k = (g_{k1}, \ldots, g_{kV})$ is the representative (prototype) of the fuzzy cluster k, where $g_{kj} \in \mathbb{R}$.

The literature holds several maximum entropy clustering algorithms to search for global regularity and obtain the smoothest reconstructions from the available data. Ref. [16] proposes one of the best-known works, hereafter named EFCM, using a method that is a variant of the Fuzzy C-Means algorithm considering entropy regularization. The minimization of the following objective function is implicated:

$$J_{EFCM} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{C}(u_{ik}) = 1 \quad \text{and} \quad u_{ik} \in [0,1] \tag{2}$$

where d is a dissimilarity function comparing the object e_i and the cluster prototype $\mathbf{g}_k$. In this work, the Euclidean and City-Block distances are used to measure dissimilarity. For a better understanding, we consider two variants of EFCM in the experimental section, named EFCM-2 and EFCM-1, with $d(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V}(x_{ij} - g_{kj})^2$ and $d(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V}|x_{ij} - g_{kj}|$, respectively. The first term in Eq. (1) denotes the total heterogeneity of the fuzzy partition as the sum of the heterogeneity of the fuzzy clusters. The second term is related to the entropy that serves as a regulating factor during the minimization process. The parameter T_u is a weight factor in the entropy term.

Despite the usefulness of the previous method, neither the correlation nor relevance of the variables are considered. The SSC has been proposed to overcome this challenge [27]. For instance, Rodríguez and de Carvalho [25,26] introduced a variant of EFCM with adaptive distances (EFCM-LP). In addition to the matrix $\mathbf{U}$ of membership degrees and the matrix $\mathbf{G}$ of prototypes, the method also provides:

- A matrix of relevance weights $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_C) = (v_{kj})_{\substack{1\le k\le C \\ 1\le j\le V}}$, where v_{kj} defines the relevance weight of the jth variable of the kth fuzzy cluster and $\mathbf{v}_k = (v_{k1}, \ldots, v_{kV})$.

The corresponding algorithm minimizes the following objective function:

$$\begin{aligned} J_{EFCM-LP} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{v}_k}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_{kj}) d(x_{ij}, g_{kj}) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{3}$$

with $u_{ik} \in [0,1]$, $v_{kj} > 0$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $\prod_{j=1}^{V}(v_{kj}) = 1$. The method EFCM-LP is defined as EFCM-LP2 when $d_{\mathbf{v}_k}$ $(1 \le k \le C)$ is based on the local adaptive Euclidean distance such that $d_{\mathbf{v}_k}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V} v_{kj}\, d(x_{ij}, g_{kj})$, with $d(x_{ij}, g_{kj}) = (x_{ij} - g_{kj})^2$. Additionally, EFCM-LP is named EFCM-LP1 when $d_{\mathbf{v}_k}$ is the local adaptive City-Block distance such that $d_{\mathbf{v}_k}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V} v_{kj}\, d(x_{ij}, g_{kj})$, with $d(x_{ij}, g_{kj}) = |x_{ij} - g_{kj}|$. The first term defines the shape and size of the clusters and encourages agglomeration, while the second term is the negative entropy and is used to control the membership degree. The weighting parameter T_u specifies the fuzziness degree. As T_u increases, the fuzziness of the clusters also increases. Although the method considers the relevance weights of the variables, it assumes that they are uncorrelated. However, the variables can be correlated in many problems. Thus, it may be more appropriate to use a Mahalanobis distance in such circumstances. Furthermore, the use of global dissimilarity measures can be effective when the internal dispersion of the groups is almost the same.

3. The proposed clustering algorithms with automatic variable selection and entropy regularization

This section presents new SSC algorithms that measure fuzzy partition heterogeneity as the sum of the heterogeneity in each fuzzy cluster. The distance-based term defines the shape and size of the groups and encourages agglomeration. Additionally, an entropy term is employed functioning as a regulating factor during the minimization process. The algorithms can be divided into two main categories: fuzzy weighting subspace clustering and entropy weighting subspace clustering [27].

Initially, a fuzzy weighting subspace clustering algorithm is defined by a local covariance matrix introduced by Gustafson and Kessel [28] that changes in each iteration of the algorithm and differs from one cluster to another. In this case, the algorithm is named as EFCM-Mk and minimizes Eq. (4), such that $u_{ik} \in [0,1]$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $det(\mathbf{M}_k) = 1$.

$$\begin{aligned} J_{EFCM-Mk} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{M}_k}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k) \\ &\quad + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{4}$$

We also consider that the adaptive distance is defined by a global covariance matrix that changes in each algorithm iteration and is the same for all clusters. The algorithm EFCM-M minimizes the following objective function:

$$\begin{aligned} J_{EFCM-M} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) \\ &\quad + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{5}$$

such that $u_{ik} \in [0,1]$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $det(\mathbf{M}) = 1$. For both cases, in addition to the membership degree matrix and the prototype vector, the global covariance matrix $\mathbf{M}$ and the local covariance matrix $\mathbf{M}_k$ are also computed from EFCM-M and EFCM-Mk, respectively.

Another alternative is when the product of the weights of the variables is equal to one. This dissimilarity function is parameterized by the vector of relevance weights $\mathbf{v} = (v_1, \ldots, v_V)$, in which $v_j > 0$ and $\prod_{j=1}^{V}(v_j) = 1$. This approach is named EFCM-GP and its objective function is defined as in Eq. (6), such that $u_{ik} \in [0,1]$, $v_j > 0$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $\prod_{j=1}^{V}(v_j) = 1$. T_u is a weighting parameter specifying the fuzziness degree.

$$\begin{aligned} J_{EFCM-GP} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j) d(x_{ij}, g_{kj}) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{6}$$

The method EFCM-GP is named EFCM-GP2 when $d_{\mathbf{v}}$ is based on the global adaptive Euclidean distance. In this case, the objective function becomes:

$$J_{EFCM-GP2} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j)(x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \tag{7}$$

Furthermore, EFCM-GP is defined as EFCM-GP1 when $d_{\mathbf{v}}$ is the global adaptive City-Block distance. In this case, the objective function becomes:

$$J_{EFCM-GP1} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j)|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) \tag{8}$$

Unlike fuzzy-weighted subspace clustering, the weights of the variables can also be controlled by entropy. In this regard, we propose a clustering method in which the set of relevant variables is the same for all clusters, and the sum of the weights of the variables is equal to one ($v_j \geq 0$ and $\sum_{j=1}^{V}(v_j) = 1$). This variant is named EFCM-GS, and the corresponding algorithm involves the minimization of Eq. (9), with $\mathbf{v} = (v_1, \ldots, v_V)$ and subject to: $u_{ik} \in [0, 1]$, $v_j \in [0, 1]$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $\sum_{j=1}^{V}(v_j) = 1$. T_u and T_v are weighting parameters, the former specify the fuzziness degree and the latter controls the relevance of the variables. Increasing the value of T_u increases the fuzziness of the clusters. Additionally, when T_v is high, the relevance of the variables tends to be similar.

$$\begin{aligned} J_{EFCM-GS} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik}) \\ &\quad + T_v \sum_{j=1}^{V}(v_j) \ln(v_j) \qquad (9) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j) d(x_{ij}, g_{kj}) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik}) \\ &\quad + T_v \sum_{j=1}^{V}(v_j) \ln(v_j) \end{aligned}$$

EFCM-GS is named EFCM-GS2 when $d_{\mathbf{v}}$ is based on the global adaptive Euclidean distance, such that $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V}(v_j)\, d(x_{ij}, g_{kj})$, with $d(x_{ij}, g_{kj}) = (x_{ij} - g_{kj})^2$. In this case, the objective function becomes:

$$\begin{aligned} J_{EFCM-GS2} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j)(x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik}) \\ &\quad + T_v \sum_{j=1}^{V}(v_j) \ln(v_j) \qquad (10) \end{aligned}$$

Furthermore, EFCM-GS is named EFCM-GS1 when $d_{\mathbf{v}}$ is the global adaptive City-Block distance such that $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V}(v_j)\, d(x_{ij}, g_{kj})$, with $d(x_{ij}, g_{kj}) = |x_{ij} - g_{kj}|$. In this case, the objective function becomes:

$$\begin{aligned} J_{EFCM-GS1} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_j)|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik}) \\ &\quad + T_v \sum_{j=1}^{V}(v_j) \ln(v_j) \qquad (11) \end{aligned}$$

The algorithms EFCM-GS2, EFCM-GS1, EFCM-GP2 and EFCM-GP1 return the matrix of membership degrees, the prototype vector for each fuzzy cluster, and the vector of relevance weights $\mathbf{v} = (v_1, \ldots, v_V)$, where v_j is the relevance weight of the jth variable estimated globally.

Additionally, a variable-wise dissimilarity with relevance weight of the variables locally estimated is also considered. In this case, the sum of the weights is equal to one [29], and the City-Block distance compares objects and prototypes. The dissimilarity function is parameterized by the vector of relevance weights $\mathbf{v}_k = (v_{k1}, \ldots, v_{kV})$, in which $v_{kj} \geq 0$ and $\sum_{j=1}^{V}(v_{kj}) = 1$, and it is associated with the kth fuzzy cluster ($k = 1, \ldots, C$). This approach is named EFCM-LS1 and defines the objective function Eq. (12), such that $u_{ik} \in [0, 1]$, $v_{kj} \in [0, 1]$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and $\sum_{j=1}^{V}(v_{kj}) = 1$. The weighting parameters T_u and T_v control, respectively, the degree of fuzziness of the clusters and the relevance of the variables in the clusters. In addition to the matrix of membership degrees and the vector of prototypes, the method returns the matrix of relevance weights $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_C) = (v_{kj})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq V}}$, where v_{kj} is the relevance weight of the jth variable in the fuzzy cluster k and $\mathbf{v}_k = (v_{k1}, \ldots, v_{kV})$.

$$\begin{aligned} J_{EFCM-LS1} &= \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \sum_{j=1}^{V}(v_{kj})|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik}) \\ &\quad + T_v \sum_{k=1}^{C}\sum_{j=1}^{V}(v_{kj}) \ln(v_{kj}) \qquad (12) \end{aligned}$$

Note that the proposed fuzzy-weighted subspace clustering algorithms require less parameter setting than the other approaches.

3.1. Optimization steps

This section provides the optimization steps of the algorithms aiming to compute the prototypes, the fuzzy partition, and either the covariance matrix or the relevance weights of the variables. The minimization of the objective functions is performed iteratively in three steps (representation, weighting, and assignment).

3.1.1. Representation step

During the representation step, the matrix of membership degree $\mathbf{U}$, the global matrix $\mathbf{M}$ for EFCM-M or the local matrices $\mathbf{M}_k$ for EFCM-Mk and the relevance weights of the variables for the other approaches are maintained fixed. This step provides the optimal solution of the prototype vector associated with each fuzzy cluster. Then, the adequacy criterion for the algorithms is minimized concerning the prototypes. The dissimilarity function is found to play an essential role in computing the prototypes. This paper provides an exact solution for each of the three possible choices of the dissimilarity functions.

Case 1: If the dissimilarity function is based on the Mahalanobis distance $(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}(\mathbf{x}_i - \mathbf{g}_k)$, we take the partial derivative of J_{EFCM-M} (see Eq. (5)) concerning $\mathbf{g}_k$ and obtain Eq. (13). Then, by solving Eq. (13), $\mathbf{g}_k$ is defined as Eq. (14).

$$\frac{\partial J_{EFCM-M}}{\partial \mathbf{g}_k} = -2 \sum_{i=1}^{P} \mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) = 0 \qquad (13)$$

$$\mathbf{g}_k = \frac{\sum_{i=1}^{P} u_{ik}\mathbf{x}_i}{\sum_{i=1}^{P} u_{ik}} \qquad (14)$$

Similarly, if the dissimilarity function is $(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k)$ (Eq. (4)), $\mathbf{g}_k$ is computed as in Eq. (14).

Case 2: If the dissimilarity function is based on the global adaptive Euclidean distance, we take the partial derivative of $J_{EFCM-GS2}$ (see Eq. (10)) concerning g_{kj} and Eq. (15) is obtained. Then, we have Eq. (16) by solving Eq. (15).

$$\frac{\partial J_{EFCM-GS2}}{\partial g_{kj}} = -2 \sum_{i=1}^{P} (u_{ik})(x_{ij} - g_{kj}) = 0 \qquad (15)$$

$$g_{kj} = \frac{\sum_{i=1}^{P} u_{ik}x_{ij}}{\sum_{i=1}^{P} u_{ik}} \qquad (16)$$

Following a similar reasoning, the prototype g_{kj} of the kth cluster that minimizes the clustering criterion $J_{EFCM-GP2}$ (see Eq. (7)) is computed as in Eq. (16).

Case 3: If the dissimilarity function is the global adaptive City-Block distance, then the minimization problem of Eqs. (8), (11) and (12) with respect to g_{kj} leads to the minimization of $\sum_{i=1}^{P} |y_i - az_i|$, where $y_i = (u_{ik})x_{ij}$, $z_i = (u_{ik})$ and $a = g_{kj}$. Since

Algorithm 1 Algorithm to compute the prototype of each fuzzy cluster.

1: Rank (y_i, z_i) such that $\frac{y_{i1}}{z_{i1}} \leq \ldots \leq \frac{y_{iP}}{z_{iP}}$;
2: To $-\sum_{l=1}^{P}|z_{il}|$ add successive values of $2|z_{il}|$
3: Find r such that $-\sum_{l=1}^{P}|z_{il}|+2\sum_{s=1}^{r}|z_{is}|<0$ and $-\sum_{l=1}^{P}|z_{il}|+2\sum_{s=1}^{r+1}|z_{is}|>0$;
4: Then $a = \frac{y_{ir}}{z_{ir}}$;
5: **if** $-\sum_{l=1}^{P}|z_{il}|+2\sum_{s=1}^{r}|z_{is}|=0$ and $-\sum_{l=1}^{P}|z_{il}|+2\sum_{s=1}^{r+1}|z_{is}|=0$ **then**
6: $a = \frac{\frac{y_{ir}}{z_{ir}}+\frac{y_{k(r+1)}}{z_{i(r+1)}}}{2}$;
7: **end if**

there is no algebraic solution for this problem, but an algorithmic solution [30] is known, we used Algorithm 1to solve it.

Alternatively, the minimization of Eqs. (8), (11) and (12) with respect to g_{kj} can be solved by expressing [30]:

$$g_{kj} = \frac{\sum_{i=1}^{P} w_{ik}\, x_{ij}}{\sum_{i=1}^{P} w_{ik}} \quad \text{where} \quad w_{ik} = \frac{u_{ik}^{(t-1)}}{|x_{ij} - g_{kj}^{(t-1)}|} \tag{17}$$

In this paper, Algorithm 1 was used to compute the prototypes of the City-Block distance-based approaches.

3.1.2. Weighting step

This step provides an optimal solution for computing the covariance matrix for the algorithms EFCM-M and EFCM-Mk, or the relevance weight of the variables for the other proposed approaches, globally for all clusters or locally for each group. During the weighting step, the prototype vector **G** and the matrix of membership degrees **U** remain fixed.

Proposition 1. *The covariance matrix or the weights of the variables minimizing the proposed objective functions are calculated according to the adaptive distance function used.*

(a) If the distance function is the local adaptive Mahalanobis distance $d_{\mathbf{M}_k}(\mathbf{x}_i, \mathbf{g}_k) = (\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k)$*, the positive definite symmetric matrices* $\mathbf{M}_k$ *that minimizes the criterion* $J_{EFCM-Mk}$ *(Eq. (4)) under* $\det(\mathbf{M}_k) = 1$ *is updated according to the following expression:*

$$\mathbf{M}_k = [\det(C_k)]^{\frac{1}{V}} C_k^{-1} \text{ with } \quad C_k = \sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T \tag{18}$$

(b) If the distance function is the global adaptive Mahalanobis distance $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{g}_k) = (\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k)$*, the positive definite symmetric matrix* $\mathbf{M}$ *that minimizes the criterion* J_{EFCM-M} *(Eq. (5)) under* $\det(\mathbf{M}) = 1$ *is updated according to Eq. (19).*

$$\mathbf{M} = [\det(Q)]^{\frac{1}{V}} Q^{-1}, \quad Q = \sum_{k=1}^{C} C_k \text{ and } \quad C_k = \sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T \tag{19}$$

(c) If the adaptive distance function is given by $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V} v_j d(x_{ij}, g_{kj})$*, the components* $v_j (j = 1, \ldots, V)$ *of the vector of weights* $\boldsymbol{v} = (v_1, \ldots, v_V)$ *minimizing the criterion* $J_{EFCM-GS}$ *(Eq. (9)) under* $v_j \in [0, 1]\ \forall\, j$*, and* $\sum_{j=1}^{V}(v_j) = 1$ *are computed as follows:*

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{ij}, g_{kj})}{T_v}\}}{\sum_{w=1}^{V} exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{iw}, g_{kw})}{T_v}\}} \tag{20}$$

When the dissimilarity function is based on the Euclidean and City-Block distances, d is defined as Eqs. (21) and (22), respectively.

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(x_{ij} - g_{kj})^2}{T_v}\}}{\sum_{w=1}^{V} exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(x_{iw} - g_{kw})^2}{T_v}\}} \tag{21}$$

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\}}{\sum_{w=1}^{V} exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{22}$$

(d) If the adaptive distance function is given by $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{V}(v_j)d(x_{ij}, g_{kj})$*, the components* $v_j (j = 1, \ldots, V)$ *of the vector of weights* $\boldsymbol{v} = (v_1, \ldots, v_V)$ *minimizing the criterion* $J_{EFCM-GP}$ *(Eq. (6)) under* $v_j > 0\ \forall\, j$ *and* $\prod_{j=1}^{V}(v_j) = 1$ *are computed according to Eq. (23).*

$$v_j = \frac{\left\{\prod_{w=1}^{V}\left[\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{iw}, g_{kw})\right]\right\}^{\frac{1}{V}}}{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{ij}, g_{kj})} \tag{23}$$

We define Eq. (23) as Eqs. (24) and (25) when the dissimilarity function is based on the Euclidean and City-Block distances, respectively.

$$v_j = \frac{\left\{\prod_{w=1}^{V}\left[\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(x_{iw} - g_{kw})^2\right]\right\}^{\frac{1}{V}}}{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(x_{ij} - g_{kj})^2} \tag{24}$$

$$v_j = \frac{\left\{\prod_{w=1}^{V}\left[\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})|x_{iw} - g_{kw}|\right]\right\}^{\frac{1}{V}}}{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})|x_{ij} - g_{kj}|} \tag{25}$$

(e) If the adaptive distance function is given by $\sum_{j=1}^{V}(v_{kj})|x_{ij} - g_{kj}|$*, the components* $v_{kj} (k = 1, \ldots, C, j = 1, \ldots, V)$ *of the vector of weights* $\boldsymbol{v}_k = (v_{k1}, \ldots, v_{kV})$ *minimizing the criterion* $J_{EFCM-LS1}$ *(Eq. (12)) under* $v_{kj} \in [0, 1]\ \forall\, k, j$ *and* $\sum_{j=1}^{V}(v_{kj}) = 1\ \forall\, k$ *are computed as follows:*

$$v_{kj} = \frac{exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\}}{\sum_{w=1}^{V} exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{26}$$

Proof. The proof is given in Appendix A. □

3.1.3. Assignment step

This step provides the solution to compute the matrix **U** of membership degree. In the assignment step, the prototype **G** and the matrix **M** for EFCM-M, $\mathbf{M}_k$ for EFCM-Mk or the relevance weights of the variables for the other approaches remain fixed.

Proposition 2. *The fuzzy partition* $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_P)$ *is updated according to Eq. (27), with* $\mathbf{u}_i = (u_{ik}, \ldots, u_{iC})$*, such that* $u_{ik} (i = 1, \ldots, P; k = 1, \ldots, C)$ *represents the membership degree of object* e_i *in the kth fuzzy cluster, under* $u_{ik} \in [0, 1]$ *and* $\sum_{k=1}^{C}(u_{ik}) = 1$.

$$u_{ik} = \frac{\exp\left\{-\frac{\Delta(\mathbf{x}_i, \mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\Delta(\mathbf{x}_i, \mathbf{g}_w)}{T_u}\right\}} \tag{27}$$

The distance function Δ *compares the ith object and the fuzzy cluster prototype k. Table 1 specifies the assignment rules to obtain the fuzzy partition according to the different adaptive distance functions.*

Proof. The proof is given in Appendix B. □

Note that the proposed maximum entropy clustering algorithms share some similarities with the Gaussian method proposed by Rui-Ping Li et al. [31] regarding the membership degree

Table 1
Assignment rules for the fuzzy partition according to the distance functions.

Distance function Δ	Algorithms	Rules for u_{ik}
$(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}(\mathbf{x}_i - \mathbf{g}_k)$	EFCM-M	$\frac{\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\right\}}$
$(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k)$	EFCM-Mk	$\frac{\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}_k(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}_w(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\right\}}$
$\sum_{j=1}^{V}(v_j)(x_{ij} - g_{kj})^2$	EFCM-GS2 EFCM-GP2	$\frac{\exp\left\{-\frac{\sum_{j=1}^{V}(v_j)(x_{ij}-g_{kj})^2}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{V}(v_j)(x_{ij}-g_{uj})^2}{T_u}\right\}}$
$\sum_{j=1}^{V}(v_j)\lvert x_{ij} - g_{kj}\rvert$	EFCM-GS1 EFCM-GP1	$\frac{\exp\left\{-\frac{\sum_{j=1}^{V}(v_j)\lvert x_{ij}-g_{kj}\rvert}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{V}(v_j)\lvert x_{ij}-g_{wj}\rvert}{T_u}\right\}}$
$\sum_{j=1}^{V}(v_{kj})\lvert x_{ij} - g_{kj}\rvert$	EFCM-LS1	$\frac{\exp\left\{-\frac{\sum_{j=1}^{V}(v_{kj})\lvert x_{ij}-g_{kj}\rvert}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{V}(v_{uj})\lvert x_{ij}-g_{wj}\rvert}{T_u}\right\}}$

computing. Therefore, we can state that our proposals have a distinct physical meaning and well-defined mathematical characteristics [14,15,31].

3.2. The proposed clustering algorithms

Algorithm 2 summarizes the proposed clustering algorithms.

Algorithm 2 The proposed algorithms.

Input: The dataset $\mathcal{D}$, the number C of clusters, the parameters $T_u > 0$ and $T_v > 0$, the maximum number of iterations T and the threshold $\varepsilon > 0$, with $\varepsilon << 1$.

Output: The vector of prototypes $\mathbf{G}$, the matrix of membership degrees $\mathbf{U}$, the matrix $\mathbf{M}$ and the matrix $\mathbf{M}_k$ or the relevance weights globally for all clusters or locally for each group.

1: **Initialization.**

$t = 0$;

Randomly initializing the matrix of membership degrees $\mathbf{U} = (u_{ik})_{\substack{1\le i\le P\\ 1\le k\le C}}$, such that $u_{ik} \geq 0$ and $\sum_{k=1}^{C}(u_{ik}^{(t)}) = 1$;

2: **repeat**

3: $t = t + 1$;

4: **Step 1: representation**.

Computing the component g_{kj} of the prototype $\mathbf{g}_k = (g_{k1}, ..., g_{kV})$ according to Section 3.1.1;

5: **Step 2: weighting**.

Computing the weights of the variables as shown in Section 3.1.2;

6: **Step 3: assignment**.

Computing the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1\le i\le P\\ 1\le j\le C}}$ according to Eq. (27).

7: **until** $max(|u_{ij}^{(t)} - u_{ij}^{(t-1)}|) < \varepsilon$ or $t \geq T$

3.2.1. Convergence of the proposed algorithms

The algorithms EFCM-M and EFCM-Mk provide a global covariance matrix $\mathbf{M}^*$ with $det(\mathbf{M}^*) = 1$, and a local covariance matrix $\mathbf{M}_k^*$ estimated locally such that $det(\mathbf{M}_k^*) = 1$ for each cluster, a fuzzy partition $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_P^*)$ and a vector of prototypes $\mathbf{G}^* = (\mathbf{g}_1^*, \ldots, \mathbf{g}_C^*)$ where:

- $J_{EFCM-M}(\mathbf{G}^*, \mathbf{M}^*, \mathbf{U}^*) = \min\{J_{EFCM-M}(\mathbf{G}, \mathbf{M}, \mathbf{U}), \mathbf{G} \in \mathbb{L}^C, \mathbf{M} \in \mathbb{M}, \mathbf{U} \in \mathbb{U}^P\}$
- $J_{EFCM-Mk}(\mathbf{G}^*, \mathbf{M}_k^*, \mathbf{U}^*) = \min\{J_{EFCM-Mk}(\mathbf{G}, \mathbf{M}_k, \mathbf{U}), \mathbf{G} \in \mathbb{L}^C, \mathbf{M}_k \in \mathbb{M}^C, \mathbf{U} \in \mathbb{U}^P\}$

– $\mathbb{L}$ is the representation space of the prototypes such that $\mathbf{g}_k \in \mathbb{L}(k = 1, \ldots, C)$ and $\mathbf{G} \in \mathbb{L}^C = \mathbb{L} \times \ldots \times \mathbb{L}$. In this paper $\mathbb{L} = \mathbb{R}^V$.
– $\mathbb{U}$ is the space of the fuzzy partition membership degrees such that $\mathbf{u}_i \in \mathbb{U}(i = 1, \ldots, P)$. In this paper $\mathbb{U} = \{\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_C) \in [0, 1] \times \ldots \times [0, 1] = [0, 1]^C : \sum_{k=1}^{C}(u_{ik}) = 1$ and $u_{ik} \geq 0\}$ and $\mathbf{U} \in \mathbb{U}^P = \mathbb{U} \times \ldots \times \mathbb{U}$.
– $\mathbb{M}$ defines the space of positive definite symmetric matrix with determinant equal to 1, such that $\mathbf{M} \in \mathbb{M}$ and $\mathbf{M}_k \in \mathbb{M}^C = \mathbb{M} \times \ldots \times \mathbb{M}$.

Moreover, the algorithms EFCM-GS (EFCM-GS2 and EFCM-GS1) and EFCM-GP (EFCM-GP1 and EFCM-GP2) provide a fuzzy partition $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_P^*)$, a vector of prototypes $\mathbf{G}^* = (\mathbf{g}_1^*, \ldots, \mathbf{g}_C^*)$ and a relevance weight vector $\mathbf{v}^*$ such that:

- $J_{EFCM-GS}(\mathbf{G}^*, \mathbf{v}^*, \mathbf{U}^*) = \min\{J_{EFCM-GS}(\mathbf{G}, \mathbf{v}, \mathbf{U}), \mathbf{G} \in \mathbb{L}^C, \mathbf{v} \in \mathcal{Z}, \mathbf{U} \in \mathbb{U}^P\}$
- $J_{EFCM-GP}(\mathbf{G}^*, \mathbf{v}^*, \mathbf{U}^*) = \min\{J_{EFCM-GP}(\mathbf{G}, \mathbf{v}, \mathbf{U}), \mathbf{G} \in \mathbb{L}^C, \mathbf{v} \in \mathcal{Z}, \mathbf{U} \in \mathbb{U}^P\}$

– $\mathcal{Z}$ is the space of vectors of weights such that $\mathbf{v} \in \mathcal{Z}$. In this paper, $\mathcal{Z} = \{\mathbf{v} = (v_1, \ldots, v_V) \in \mathbb{R}^V : v_j > 0$ and $\prod_{j=1}^{V}(v_j) = 1\}$ or $\mathcal{Z} = \{\mathbf{v} = (v_1, \ldots, v_V) \in \mathbb{R}^V : v_j \in [0, 1]$ and $\sum_{j=1}^{V}(v_j) = 1\}$.

Additionally, EFCM-LS1 provides a fuzzy partition $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_P^*)$, a vector of prototypes $\mathbf{G}^* = (\mathbf{g}_1^*, \ldots, \mathbf{g}_C^*)$ and a vector of relevance weight vectors $\mathbf{V}^* = (\mathbf{v}_1^*, \ldots, \mathbf{v}_C^*)$ such that:

- $J_{EFCM-LS1}(\mathbf{G}^*, \mathbf{V}^*, \mathbf{U}^*) = \min\{J_{EFCM-LS1}(\mathbf{G}, \mathbf{V}, \mathbf{U}), \mathbf{G} \in \mathbb{L}^C, \mathbf{V} \in \mathcal{Z}^C, \mathbf{U} \in \mathbb{U}^P\}$

– $\mathcal{Z}$ is the space of vectors of weights such that $\mathbf{v}_k \in \mathcal{Z}, (k = 1, \ldots, C)$. In this paper, $\mathcal{Z} = \{\mathbf{v} = (v_1, \ldots, v_V) \in \mathbb{R}^V : v_j \in [0, 1]$ and $\sum_{j=1}^{V}(v_j) = 1\}$, and $\mathbf{V} \in \mathcal{Z}^C = \mathcal{Z} \times \ldots \times \mathcal{Z}$.

Similarly to Ref. [32], the convergence properties of the proposed algorithms can be studied from the series:

- $v_{EFCM-M}^{(t)}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathbb{M} \times \mathbb{U}^P$ and $u_{EFCM-M}^{(t)} = J_{EFCM-M}(v_{EFCM-M}^{(t)}) = J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v_{EFCM-Mk}^{(t)}(\mathbf{G}^{(t)}, \mathbf{M}_k^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathbb{M}^C \times \mathbb{U}^P$ and $u_{EFCM-Mk}^{(t)} = J_{EFCM-Mk}(v_{EFCM-Mk}^{(t)}) = J_{EFCM-Mk}(\mathbf{G}^{(t)}, \mathbf{M}_k^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v_{EFCM-GS}^{(t)}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathcal{Z} \times \mathbb{U}^P$ and $u_{EFCM-GS}^{(t)} = J_{EFCM-GS}(v_{EFCM-GS}^{(t)}) = J_{EFCM-GS}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v_{EFCM-GP}^{(t)}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathcal{Z} \times \mathbb{U}^P$ and $u_{EFCM-GP}^{(t)} = J_{EFCM-GP}(v_{EFCM-GP}^{(t)}) = J_{EFCM-GP}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v_{EFCM-LS1}^{(t)}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathcal{Z}^C \times \mathbb{U}^P$ and $u_{EFCM-LS1}^{(t)} = J_{EFCM-LS1}(v_{EFCM-LS1}^{(t)}) = J_{EFCM-LS1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

From the initial terms: $v_{EFCM-M}^{(0)}(\mathbf{G}^{(0)}, \mathbf{M}^{(0)}, \mathbf{U}^{(0)})$, $v_{EFCM-Mk}^{(0)}(\mathbf{G}^{(0)}, \mathbf{M}_k^{(0)}, \mathbf{U}^{(0)})$, $v_{EFCM-GS}^{(0)}(\mathbf{G}^{(0)}, \mathbf{v}^{(0)}, \mathbf{U}^{(0)})$, $v_{EFCM-GP}^{(0)}(\mathbf{G}^{(0)}, \mathbf{v}^{(0)}, \mathbf{U}^{(0)})$ and $v_{EFCM-LS1}^{(0)}(\mathbf{G}^{(0)}, \mathbf{V}^{(0)}, \mathbf{U}^{(0)})$, the algorithms EFCM-M, EFCM-Mk,

EFCM-GS, EFCM-GP and EFCM-LS1 compute the different terms of the series, $v^{(t)}_{EFCM-M}$, $v^{(t)}_{EFCM-Mk}$, $v^{(t)}_{EFCM-GS}$, $v^{(t)}_{EFCM-GP}$, and $v^{(t)}_{EFCM-LS1}$, until the respective convergence (to be demonstrated) when the objective functions J_{EFCM-M}, $J_{EFCM-Mk}$, $J_{EFCM-GS}$, $J_{EFCM-GP}$ and $J_{EFCM-LS1}$ reach stationary values.

Proposition 3.

(i) The series $u^{(t)}_{EFCM-M} = J_{EFCM-M}(v^{(t)}_{EFCM-M}) = J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *decreases at each iteration and converges;*

(ii) The series $u^{(t)}_{EFCM-Mk} = J_{EFCM-Mk}(v^{(t)}_{EFCM-Mk}) = J_{EFCM-Mk}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *decreases at each iteration and converges;*

(iii) The series $u^{(t)}_{EFCM-GS} = J_{EFCM-GS}(v^{(t)}_{EFCM-GS}) = J_{EFCM-GS}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *decreases at each iteration and converges;*

(iv) The series $u^{(t)}_{EFCM-GP} = J_{EFCM-GP}(v^{(t)}_{EFCM-GP}) = J_{EFCM-GP}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *decreases at each iteration and converges;*

(v) The series $u^{(t)}_{EFCM-LS1} = J_{EFCM-LS1}(v^{(t)}_{EFCM-LS1}) = J_{EFCM-LS1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *decreases at each iteration and converges;*

Proof. The proof is given in Appendix C. □

Proposition 4.

(i) The series $v^{(t)}_{EFCM-M} = (\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges;*

(ii) The series $v^{(t)}_{EFCM-Mk} = (\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges;*

(iii) The series $v^{(t)}_{EFCM-GS} = (\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges;*

(iv) The series $v^{(t)}_{EFCM-GP} = (\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges;*

(v) The series $v^{(t)}_{EFCM-LS1} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges.*

Proof. The proof is given in Appendix D. □

3.2.2. Complexity analysis of the proposed algorithms

To obtain prototypes for methods based on the Euclidean and Mahalanobis distances, the computational complexity is $O(P \times C \times V)$, while $O(P \times C \times V \times \log(P))$ is applied for approaches based on the City-Block distance. The number of objects, clusters, and variables is represented by P, C, and V, respectively. In the weighting step, the complexity to obtain $\mathbf{M}$ and $\mathbf{M}_k$ for EFCM-M and EFCM-Mk, respectively, depends on the matrix inversion method used to implement the clustering algorithm. In this paper, the complexity for obtaining $\mathbf{M}$ for EFCM-M is $O(\max\{P \times C \times V^2, V^3\})$, and $O(C \times \max\{P \times V^2, V^3\})$ to compute $\mathbf{M}_k$ for the EFCM-Mk algorithm. For the other methods, the complexity time to compute the relevance weights is $O(P \times C \times V)$. Finally, for computing the matrix of membership degree for EFCM-M and EFCM-Mk, the complexity time is $O(P \times C \times V^2)$. However, for other approaches, it is $O(P \times C \times V)$. Therefore, globally, assuming that the iterative function needs T iterations to converge, we would have:

- Complexity time of $O(T \times \max\{P \times C \times V^2, V^3\})$ for EFCM-M.
- Complexity time of $O(T \times C \times \max\{P \times V^2, V^3\})$ for EFCM-Mk.
- Complexity time of $O(T \times P \times C \times V)$ for EFCM-GS2 and EFCM-GP2.
- Finally, complexity time of $O(T \times C \times P \times V \times \log(P))$ for EFCM-GS1, EFCM-GP1 and EFCM-LS1.

Maximum entropy clustering has lower computational complexity than its fuzzy C-Means versions, which makes this type of method more attractive for large-scale and high-dimensional data clustering.

4. Experimental results

This section aims to evaluate the performance and illustrate the usefulness of the proposed algorithms by applying them to suitable synthetic and real datasets. All the experiments were performed on the same machine (OS: Windows 10 Professional 64-bits, Memory: 8 GiB, Processor: Intel Core i7-5500U CPU @ 2.40 GHz).

4.1. Experimental setting

The proposed algorithms were compared with the five previous most related fuzzy clustering models: EFCM-2 and EFCM-1 [16], the Fuzzy Co-Clustering algorithm for Images (hereafter, we will adopt the notation EFCM-LS2) [24], EFCM-LP2 [25] and EFCM-LP1 [26] algorithms.

Optimal values for T_u and T_v were found using a grid search strategy. Following a procedure similar to Ref. [33], for each dataset, the value of T_u varied, as shown in Table 2. Then, an optimal value for the parameter is obtained when the minimum centroid distance falls under 0.1 for the first time. For the methods with both T_u and T_v, T_v varied between 10 and 10^8, as shown in Table 2. Fixing the value of T_v, we follow the procedure abovementioned to compute T_u. Subsequently, we chose the pair (T_u, T_v) with the maximum distance. Note that the optimal values are calculated without supervision.

In contrast, before running the algorithms, each dataset was normalized so that the characteristics have a zero mean and a standard deviation of one [33]. Table 2 shows the parameter settings of all experiments. Furthermore, the number of clusters is set equal to the number of a priori classes for simplicity. Finally, the clustering results obtained by the algorithms were compared using two measures: The Hullermeier index (*HUL*) [34] and the Adjusted Rand index (*ARI*) [35].

The internal validation metric *HUL* (Eq. (28)) is a fuzzy extension of the Rand index comparing two fuzzy partitions. It considers its values on the interval [0, 1], in which the value 1 indicates perfect agreement between the fuzzy partitions. In contrast, values close to 0 correspond to the cluster agreement found by chance. $\mathbf{V}$ and $\mathbf{U}$ are two fuzzy partitions, and $\|.\|$ is a proper metric on $[0, 1]^C$. This work uses $\|.\|$ as the L_1-norm divided by 2, as in Ref. [34].

$$HUL = 1 - \left[\frac{\sum_{i=1}^{P-1} \sum_{j=i+1}^{P} | \| \mathbf{V}_i - \mathbf{V}_j \| - \| \mathbf{U}_i - \mathbf{U}_j | \|}{\binom{P}{2}} \right] \tag{28}$$

The Adjusted Rand Index is an external validation metric that measures the correspondence between two crisp partitions. It considers its values on the interval $[-1, 1]$, in which 1 indicates perfect agreement. Moreover, values near 0 or negative correspond to cluster agreement found by chance. Let $\mathcal{A} = \{A_1, \ldots, A_M\}$ be the a priori partition into M classes and $\mathcal{Q} = \{Q_1, \ldots, Q_C\}$ be the partition into C clusters provided by the clustering algorithm. The *ARI* is computed as:

$$ARI = \frac{\sum_{i=1}^{M} \sum_{j=1}^{C} \binom{n_{i,j}}{2} - \binom{P}{2}^{-1} \sum_{i=1}^{M} \binom{n_i}{2} \sum_{j=1}^{C} \binom{n_j}{2}}{\frac{1}{2}\left[\sum_{i=1}^{M} \binom{n_i}{2} + \sum_{j=1}^{C} \binom{n_j}{2} \right] - \binom{P}{2}^{-1} \sum_{i=1}^{M} \binom{n_i}{2} \sum_{j=1}^{C} \binom{n_j}{2}} \tag{29}$$

where $n_{i,j}$ is the number of agreements between class A_i and the cluster Q_j, n_i (or n_j) is the number of data points in class A_i (or cluster Q_j).

Table 2
Experimental settings.

Parameters	Values
T_u controls the membership degree of objects	0.01 to 100 (with step 0.01)
T_v controls the relevance weight of the variables	10 to 10^8 (with step 10)
T sets the maximum number of iterations	100
ε is a stop condition parameter	10^{-5}

4.2. Experimental analysis with synthetic datasets

In the first experiment, we investigate the performance aspects of the proposed algorithms based on the Mahalanobis (EFCM-M, EFCM-Mk), Euclidean (EFCM-GS2, EFCM-GP2), and City-block (EFCM-GS1, EFCM-GP1, EFCM-LS1) distances with synthetic datasets. In this regard, we created four synthetic datasets described by two-dimensional vectors generated randomly from a normal distribution. The synthetic datasets were generated having classes of different sizes and shapes, as in Ref. [36]. Each synthetic dataset has 450 points, divided into four classes of unequal sizes. The classes were drawn according to a bivariate normal distribution with vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For more details, see Table 3.

We consider four data configurations. Firstly, the class covariance matrices are diagonal and nearly equal (Fig. 1(a)). In the second configuration, the class covariance matrices are diagonal but unequal (Fig. 1(b)). The class covariance matrices are not diagonal but almost the same for the third configuration (Fig. 1(c)). Finally, we consider that in addition to not being diagonal, the class covariance matrices are also unequal (Fig. 1(d)). Table 4 shows a detailed review of each synthetic data configuration.

Fifty replications of each synthetic dataset were carried out in a framework of a Monte Carlo experiment. We performed 50 random initializations of the algorithm for each dataset. The best result of these 50 repetitions is selected according to their respective objective function. The average and standard deviation of the indices were calculated based on the 50 Monte Carlo iterations. Note that the number of clusters is assumed to be equal to four.

The Friedman test [37] is used to explore the statistical significance of the results obtained. We analyze the algorithms by ranking them on each dataset separately. The best performing algorithm is ranked as 1, the second best is ranked as 2, and so on. In the case of ties, average ranks are assigned. Subsequently, we calculate and compare the average ranks of all algorithms on the datasets. Suppose that the null hypothesis that all the algorithms are performing equivalently is rejected under the Friedman test. In that case, the Nemenyi post-hoc test [38] is used to determine which algorithms perform statistically differently. The Nemenyi test compares algorithms in a pairwise manner. According to this test, the performances of two algorithms are significantly different if the distance of the average ranks exceeds the critical distance. The objective is to determine whether there is at least one method that is significantly better than at least one other method at the $\alpha = 0.05$ level.

4.2.1. Results

Fig. 2 shows the values of the mean and the standard deviation for *HUL* and *ARI* for different methods and data configurations. The center of the line in Fig. 2 represents the mean. In addition, the width of each bar above and below the mean denotes the standard deviation. Table 5, in the supplementary material, shows the numeric values.

In data configuration 1 (cluster covariance matrices are diagonal and almost the same), the best results concerning the index *HUL* were obtained by EFCM-1, EFCM-GP1, and EFCM-M with values of 0.7946, 0.7739 and 0.7091, respectively. For *ARI*, EFCM-1 had the best performance. Moreover, EFCM-GP1 and EFCM-2 achieved, respectively, the second and third-best values. The EFCM-LS1, EFCM-GS1 and EFCM-Mk algorithms produced the worst clustering results for *HUL*, while for *ARI*, it was the algorithms EFCM-Mk, EFCM-LS1, and EFCM-LP2. As expected for this data configuration, almost all the methods with global adaptive distance outperformed their respective variants based on local adaptive distance, i.e., the methods EFCM-M, EFCM-GP1, EFCM-GP2, EFCM-GS1 outperformed, respectively, the methods EFCM-Mk, EFCM-LP1, EFCM-LP2, EFCM-LS1. EFCM-GS2 outperforms EFCM-LS2 concerning the *ARI* index, but EFCM-LS2 surpasses EFCM-GS2 regarding the *HUL* index.

Data configuration 2 presents cluster covariance matrices that are diagonal but unequal. In this case, the best results were provided by the algorithms EFCM-GP1, EFCM-LP2, and EFCM-1 for *HUL* and by EFCM-LP2, EFCM-LP1, and EFCM-Mk for *ARI*. The algorithms EFCM-M, EFCM-GP2 and EFCM-2 had the worst performances for *HUL* and EFCM-GS1, EFCM-M, and EFCM-GS2 for *ARI*. In this configuration, almost all methods with local adaptive distance presented better results than their respective variants based on global adaptive distance: EFCM-Mk, EFCM-LP2, EFCM-LS1, and EFCM-LS2 surpassed, respectively, the methods EFCM-M, EFCM-GP2, EFCM-GS1, EFCM-GS2. EFCM-LP1 surpasses EFCM-GP1 concerning *ARI*, but EFCM-GP1 outperforms EFCM-LP1 regarding the index *HUL*.

In data configuration 3 (cluster covariance matrices are not diagonal but almost the same), EFCM-GP1, EFCM-Mk, and EFCM-M presented the first, second, and third-best performances for *HUL*, respectively. For the *ARI* index, the best results were achieved by EFCM-M, EFCM-Mk, and EFCM-GP1. As expected for this data configuration, EFCM-M and EFCM-Mk had some of the best results, while the algorithms EFCM-LS1 and EFCM-GS1 showed the worst results for *HUL*, and EFCM-LP2 and EFCM-LS2 for *ARI*. Finally, for ARI in this data configuration, the methods with global adaptive distance outperformed their respective variants based on local adaptive distance, i.e., the methods EFCM-M, EFCM-GP1, EFCM-GP2, EFCM-GS1 and EFCM-GS2 outperformed, respectively, the methods EFCM-Mk, EFCM-LP1, EFCM-LP2, EFCM-LS1, and EFCM-LS2.

For data configuration 4, in which the cluster covariance matrices are not diagonal and unequal, the algorithm EFCM-Mk outperforms the other approaches for both indices. EFCM-M presented the worst performance at *HUL*, and EFCM-GS1 and EFCM-GS2 had the worst performance for *ARI*. Finally, concerning the *ARI* index, the methods with local adaptive distance outperformed their respective variants based on global adaptive distance, i.e., the methods EFCM-Mk, EFCM-LP1, EFCM-LP2, EFCM-LS1 and EFCM-LS2 outperformed, respectively, the methods EFCM-M, EFCM-GP1, EFCM-GP2, EFCM-GS1, and EFCM-GS2.

Figs. 3 and 4 show the comparison of the algorithms against each other with the Nemenyi test. For the models joined by the horizontal lines, there is no evidence of statistical significant differences. The average performance rank is also presented. In this regard, concerning HUL index, Fig. 3 shows that the EFCM-1 reached the best average performance ranking for the first configuration. However, there is no consistent evidence indicating statistical differences among EFCM-1, EFCM-GP1, and AFCM-M. In configuration 2, the best results were presented by EFCM-GP1,

Table 3
Description of the synthetic dataset.

Total objects per class				Bi-variate normal distribution definition	
Class 1	Class 2	Class 3	Class 4	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$
150	150	50	100	$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$	$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}$

Table 4
Synthetic data configurations.

Classes	Configuration 1				Configuration 2				Configuration 3				Configuration 4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
μ_1	45	70	45	42	45	70	50	42	45	70	45	42	45	70	50	42
μ_2	30	38	42	20	22	38	42	2	30	38	42	20	22	38	42	2
σ_1^2	100	81	100	81	144	81	36	9	100	81	100	81	144	81	36	9
σ_2^2	9	16	16	9	9	36	81	144	9	16	16	9	9	36	81	144
ρ	0	0	0	0	0	0	0	0	0.7	0.8	0.7	0.8	0.7	0.8	0.7	0.8

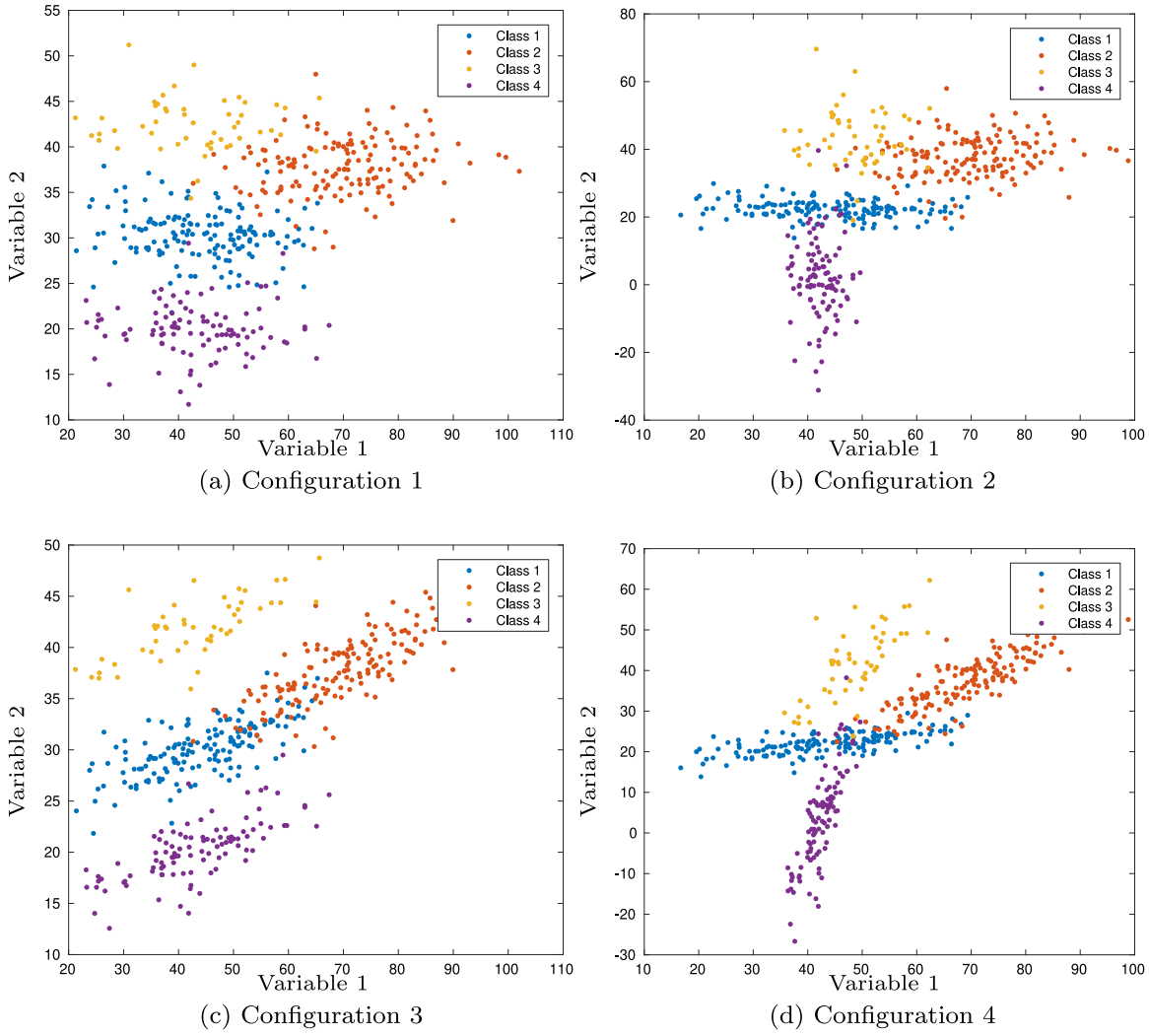
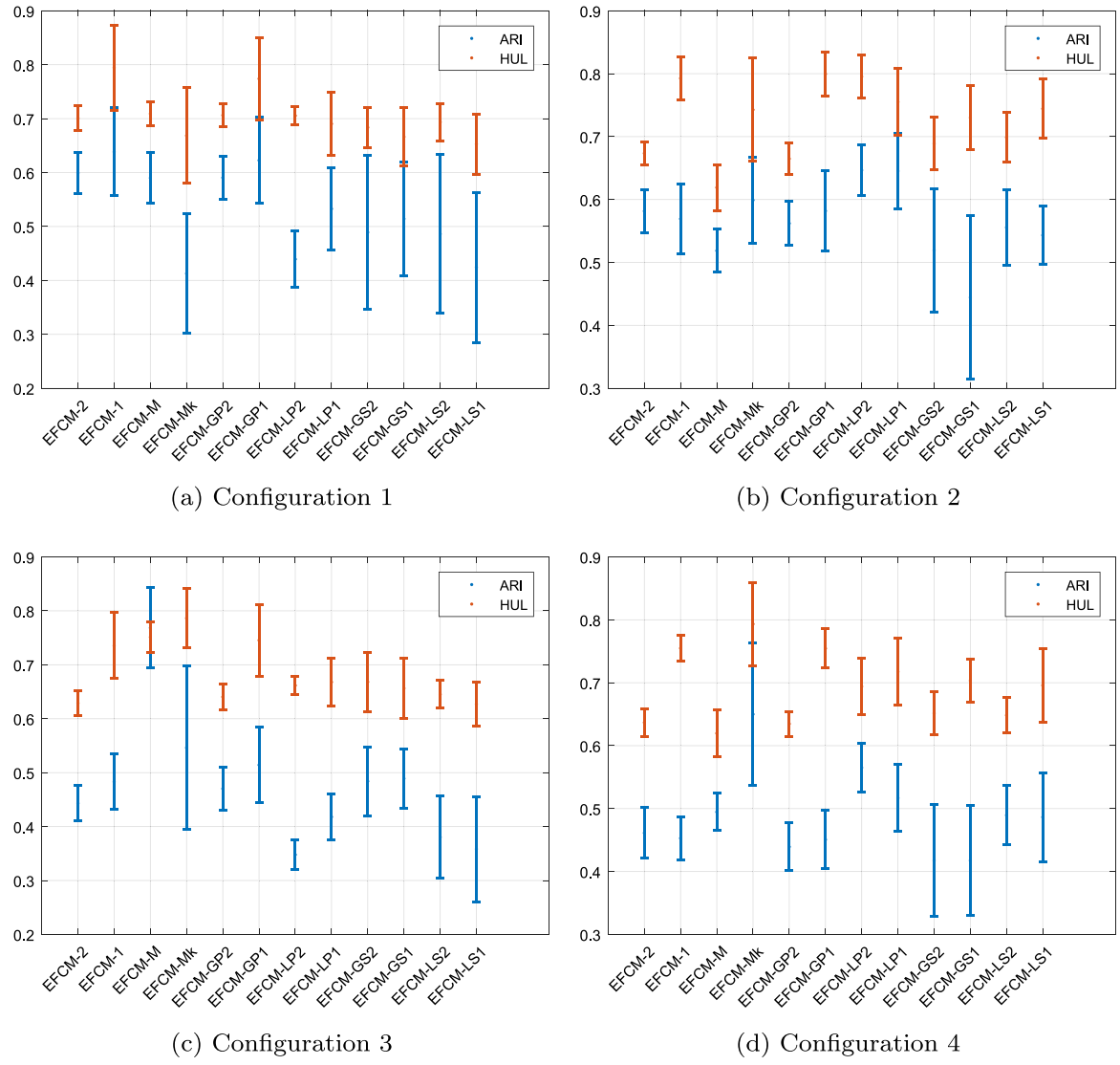


Fig. 1. Clusters drawn according to the (a) first, (b) second, (c) third and (d) fourth configurations.

EFCM-LP2, EFCM-1, EFCM-LP1, while for configuration 3, EFCM-Mk, EFCM-M, and EFCM-GP1 showed the highest performance. Finally, for the last configuration, EFCM-Mk achieved the best results on average. Nonetheless, it did not present consistent evidence to indicate statistical significant differences regarding EFCM-GP1, EFCM-1.

Analyzing the results according to the index *ARI*, Fig. 4 shows that the global models EFCM-1, EFCM-GP1, EFCM-2, EFCM-M and EFCM-GP2 presented the best performances. However, the local methods EFCM-LP2 and EFCM-LP1 are more appropriate to configuration 2. The proposed method with the global covariance matrix EFCM-M was statistically better than the other clustering



(a) Configuration 1 (b) Configuration 2

(c) Configuration 3 (d) Configuration 4

Fig. 2. Mean and standard deviation for each synthetic data configuration.

algorithms. Finally, for configuration 4, our proposal with the local covariance matrices showed the highest average performance ranking. However, there are no statistical significant differences regarding local methods EFCM-LP2 and EFCM-LP1.

4.3. Experimental analysis of synthetic datasets with outliers

We conducted another experiment to verify the behavior of the proposed methods in the presence of outliers. For this purpose, a synthetic dataset (Fig. 5(a)) with 80 objects described by two-dimensional vectors was randomly generated from a normal distribution according to the parameters in Table 6. Three different percentages of outliers (10%, 20%, 30%) were added to the dataset (Fig. 5(b), (c) and (d)) to assess the robustness of the methods. Table 6 shows the configuration of the parameters for the outliers datasets. Fifty replications of the synthetic dataset were carried out in a framework of a Monte Carlo experiment. For each dataset, we performed 50 random initializations of the clustering algorithm. The average and standard deviation of the indices were calculated based on the 50 Monte Carlo iterations, and the number of clusters was set to two.

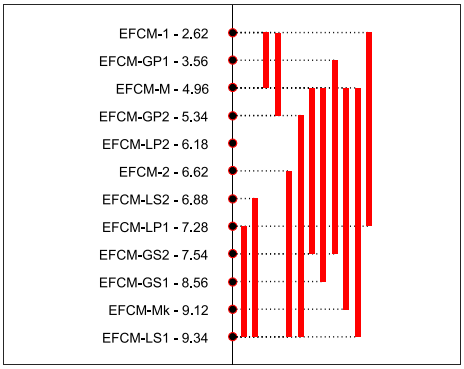
We measured the performance considering misclassification (*HUL* and *ARI*) and robustness detection (*rd*). This last index was introduced by D'Urso et al. [39] to assess the robustness of algorithms in the presence of outliers. It compares the representatives provided by the algorithms with the ideal representative of the a priori classes. Herein, we adopt such index as follows:

$$rd = \frac{\Delta(\mathbf{g}_1^{id}, \mathbf{g}_1) + \Delta(\mathbf{g}_1^{id}, \mathbf{g}_2) + \Delta(\mathbf{g}_2^{id}, \mathbf{g}_1) + \Delta(\mathbf{g}_2^{id}, \mathbf{g}_1)}{2\Delta(\mathbf{g}_1^{id}, \mathbf{g}_2^{id})} \tag{30}$$

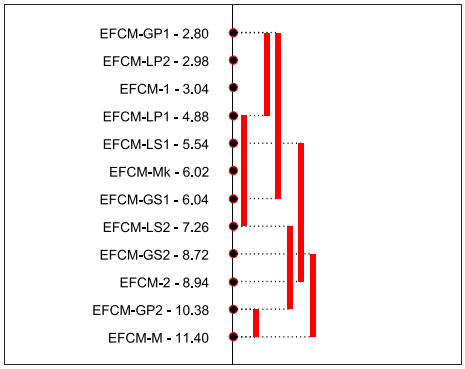
where $\mathbf{g}_1$ and $\mathbf{g}_2$ denote, respectively, the representatives of clusters 1 and 2 provided by the algorithms, $\mathbf{g}_1^{id}$ and $\mathbf{g}_2^{id}$ denote, respectively, the "ideal" representatives of the a priori classes 1 and 2, and Δ is the Euclidean distance between vectors.

4.3.1. Results

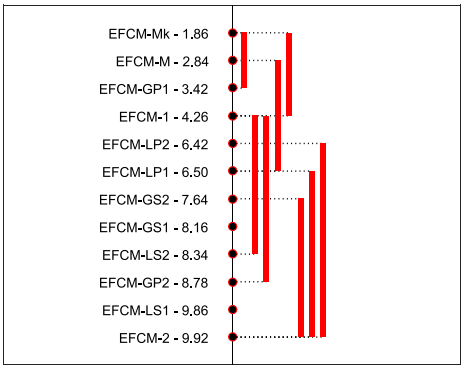
Fig. 6 presents the results for the *HUL*, *ARI*, and *rd* indices with different percentages of outliers according to the mean and standard deviation. The center of the line in Fig. 6 represents the mean, while the width of each bar above and below the mean denotes the standard deviation. Table 7, in the supplementary material, indicates the numerical values. For *ARI* and *rd* with 0% of outliers, the methods with the Euclidean distance presented the best results in relation to those based on the Mahalanobis and City-Block distances. However, concerning misclassification measured by *HUL* and *ARI* with different percents of outliers,



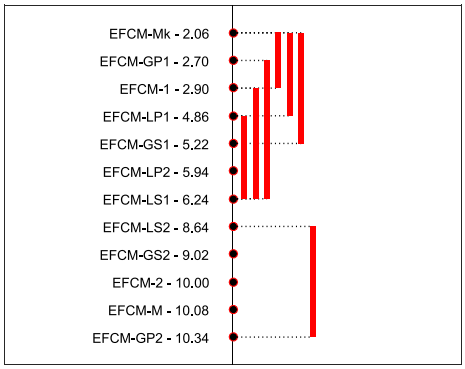
(a) Configuration 1



(b) Configuration 2

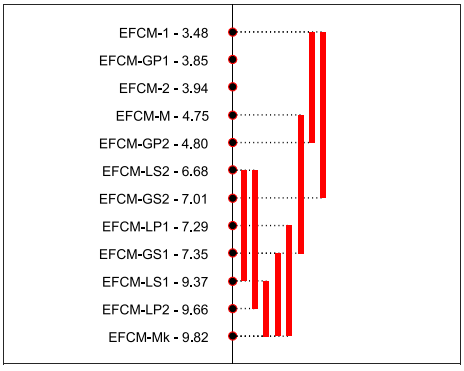


(c) Configuration 3

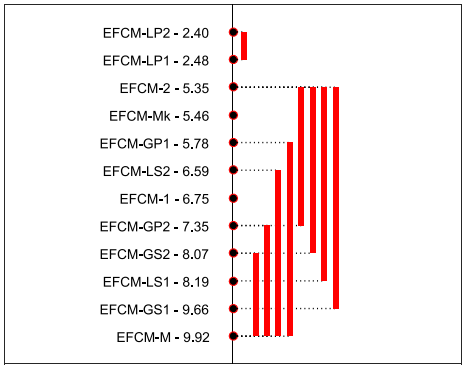


(d) Configuration 4

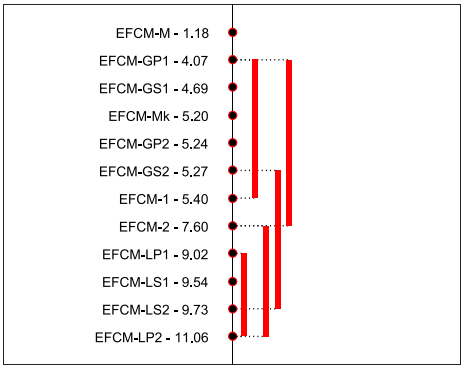
Fig. 3. Comparison of the algorithms against each other through the Nemenyi test for *HUL*.



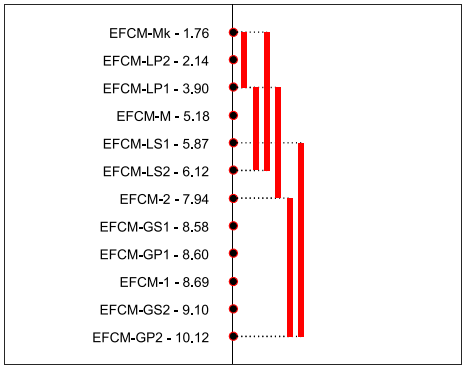
(a) Configuration 1



(b) Configuration 2



(c) Configuration 3



(d) Configuration 4

Fig. 4. Comparison of the algorithms against each other through the Nemenyi test for *ARI*.

Table 6
Parameter settings for synthetic datasets with and without outliers.

Parameters	Class 1	Class 2	Outliers 10%	Outliers 20%	Outliers 30%
μ_1	0	0.8	0.8	0.8	0.8
μ_2	0	0.8	1	1	1
σ_1^2	0.05	0.05	5	5	5
σ_2^2	0.05	0.05	5	5	5
Number of objects	40	40	8	16	24

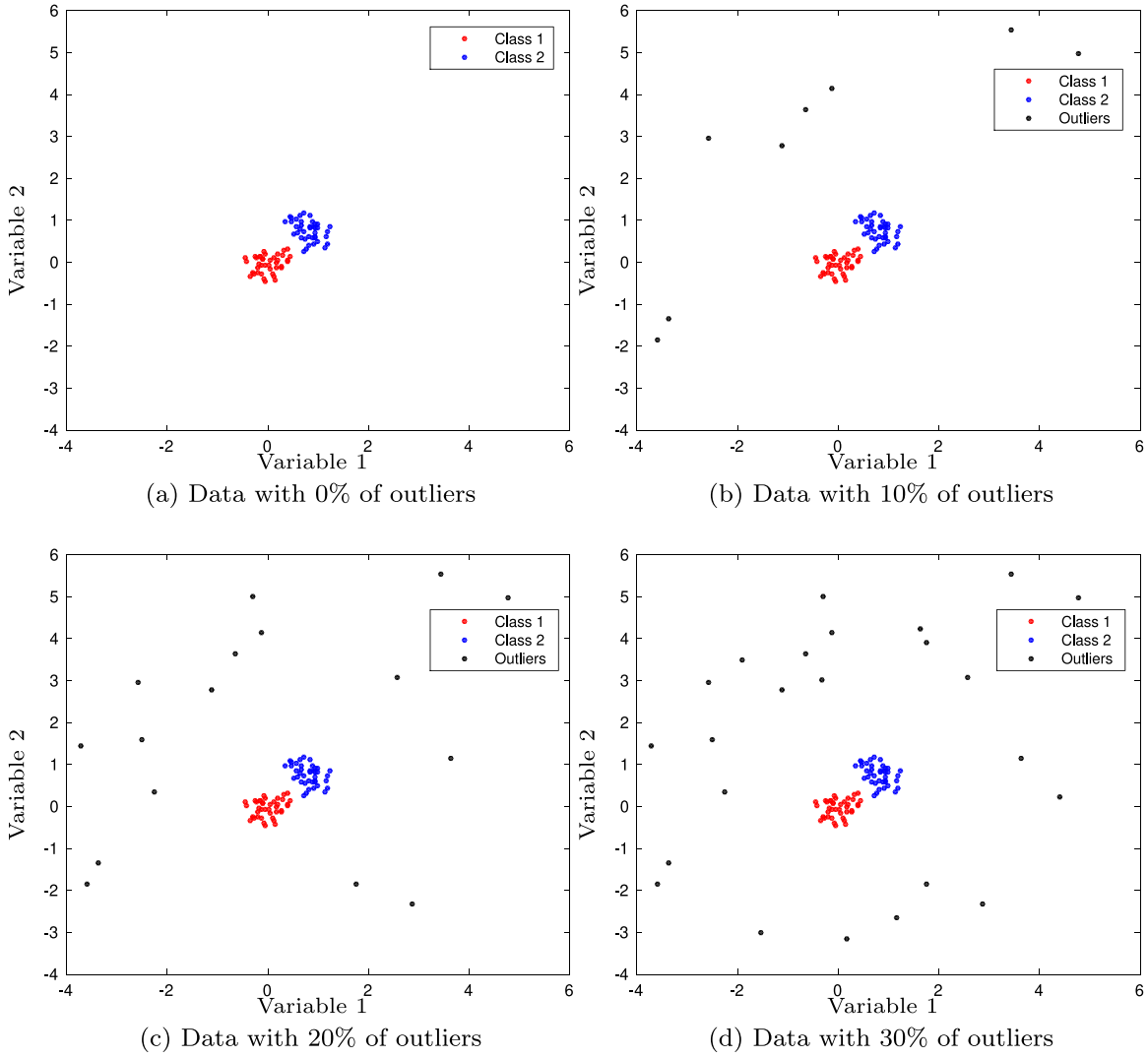


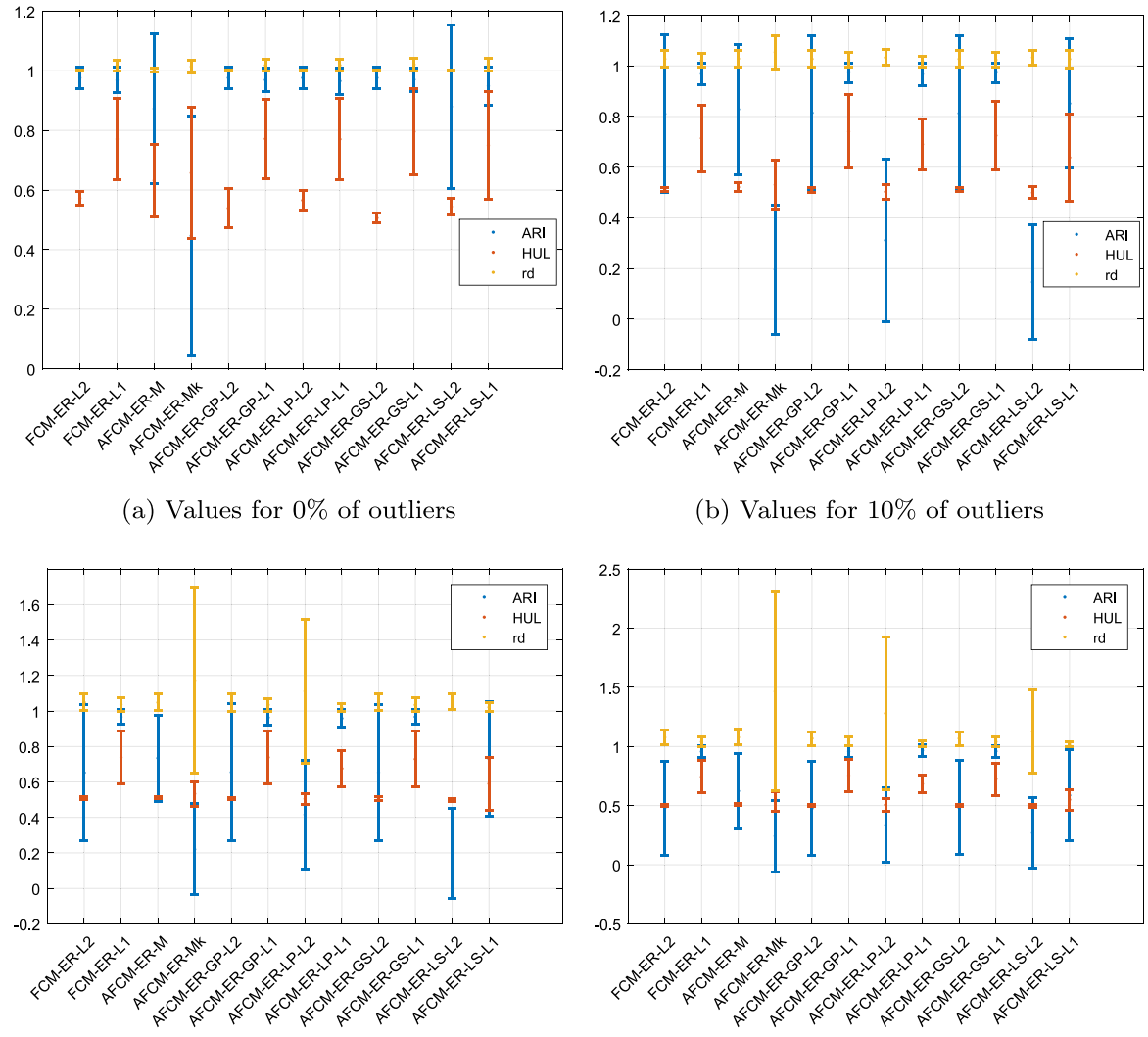
Fig. 5. Synthetic dataset with different percentage of outliers. (a) data with 0% of outliers, (b) data with 10% of outliers, (c) data with 20% of outliers and (d) data with 30% of outliers.

the City-Block distance-based algorithms outperformed other approaches, identifying the presence of clusters even in a noisy environment. The performance clustering degrades very slowly as the percentage of outliers increases. In addition, these methods are more robust in producing cluster prototypes that are not very dissimilar from the "ideal" centers (*rd* index). In conclusion, as expected, regardless of the index considered, the algorithms based on the Euclidean and Mahalanobis distances are less robust in the presence of outliers than those with the City-Block distance.

Fig. 7 shows that for *HUL*, the City-Block distance-based methods achieved the highest average performance ranking, regardless of the percentage of outliers. Additionally, almost all algorithms showed similar results when analyzing the values for *ARI* with 0% percent of outliers, see Fig. 8. However, when the number of outliers increases, the algorithms with the City-Block distance EFCM-1, EFCM-GS1, EFCM-GP1, and EFCM-LP1 presented the best performance. Besides, there is no consistent evidence to indicate statistical performance differences among them.

4.4. Real datasets

The previous and the proposed algorithms were also applied on 15 real datasets available at the *UCI* machine learning repository [40]: Automobile (Auto), Balance Scale, Haberman's Survival, Statlog (Heart), Image Segmentation, Ionosphere, Iris Plants, Mnist, Thyroid Gland, User Knowledge Modeling (UKM), Vehicle, Vertebral Column, Wisconsin Diagnostic Breast Cancer (WDBC), Wall-Following Robot Navigation (WFRN), and Wine. Table 8 briefly describes the datasets in which P, V, C and M represent the number of patterns, variables, a priori classes, and percentage



(a) Values for 0% of outliers

(b) Values for 10% of outliers

(c) Values for 20% of outliers

(d) Values for 30% of outliers

Fig. 6. Mean and standard deviation (in parentheses) of *HUL*, *ARI*, and *rd* on the synthetic interval-valued dataset for different percentages of outliers.

of objects of the majority class, respectively. As can be seen, several sample sizes, number of attributes and number of classes were considered.

4.4.1. Experimental setting

For the real datasets, we chose the values for the hyperparameters based on the same procedure used in Section 4.1. However, T_u varied between 0.01 and 300 (with step 0.01). The algorithms were executed on each dataset 100 times, and the cluster centers were randomly initialized at each time. The best result for each algorithm was selected according to its respective objective function. For each dataset, the number of clusters was set equal to the number of a priori classes, as shown in Table 8. A hard partition was obtained from the fuzzy partition provided by each algorithm as described in Section 4.1. Subsequently, the *HUL* and *ARI* indices were considered to assess the misclassification.

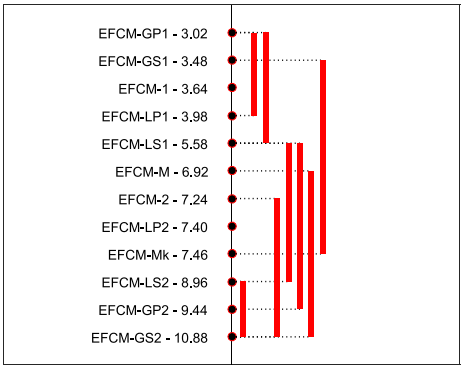
4.4.2. Results

Table 9, in the supplementary material, shows the results provided by the algorithms on real datasets and the performance rank per algorithm (in parenthesis), according to the indices and datasets. Additionally, Table 10, in the supplementary material, presents the average performance ranking of the clustering algorithms according to both indices computed from Table 9, in addition to the performance ranking of the clustering algorithms (in parentheses) according to the average performance ranking. Finally, Fig. 9 shows the cumulative performance ranking of each method calculated through Table 9.

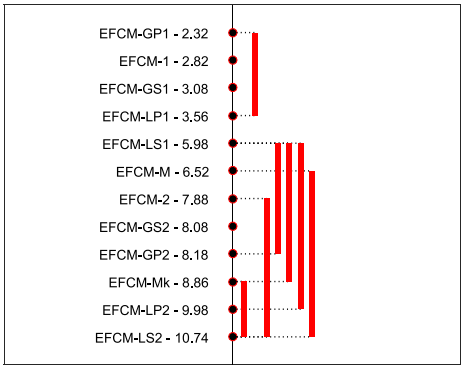
We can observe that the algorithm EFCM-GP1 presents the best average performance ranking for real datasets, regardless of the index considered. In this regard, the proposed method with the global adaptive City-Block distance showed to be a promising approach for clustering the real datasets. Moreover, the algorithms EFCM-1 and EFCM-LP1 achieved the second and third-best average rankings for the *HUL* index, while EFCM-LS1 and EFCM-LP2 showed the same results for *ARI*. Finally, the algorithms EFCM-GS2 and EFCM-Mk reached the worst results for *HUL* and *ARI*, respectively.

4.4.3. Analysis of the hyperparameters setting

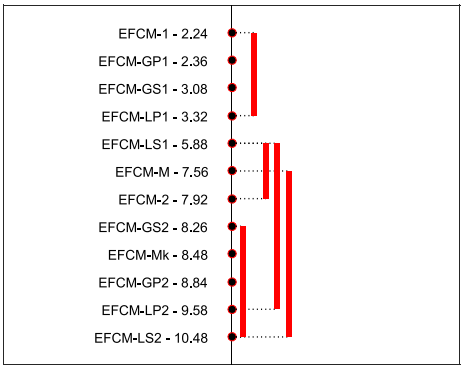
The proposed methods introduce the regularization coefficients T_u and T_v for managing the membership degree and the relevance weights of the variables. Selecting their correct values significantly influences the subsequent clustering, as shown with the Iris dataset in Fig. 10. As can be seen, upon $T_u > 5$, the performance of the algorithms reaches a flat area of low values of *HUL*. In addition, when T_u increases, the results according to *ARI* tend to decrease. The proposed methods present this behavior



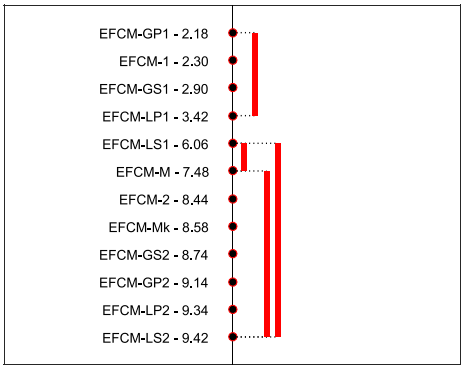
(a) Average performance ranking for 0% of outliers



(b) Average performance ranking for 10% of outliers



(c) Average performance ranking for 20% of outliers



(d) Average performance ranking for 30% of outliers

Fig. 7. Comparison of the algorithms with each other with the Nemenyi test for *HUL* with different percentages of outliers.

Table 8
Summary of the real datasets.

Dataset	*P*	*V*	*C*	*M*	Dataset	*P*	*V*	*C*	*M*	Dataset	*P*	*V*	*C*	*M*
Auto	205	25	6	33%	Balance Scale	625	4	3	46%	Haberman	306	3	2	74%
Heart	270	13	2	56%	Image Seg.	2310	19	7	14%	Ionosphere	351	33	2	64%
Iris	150	4	3	33%	Mnist	14 780	784	2	53%	Thyroid	215	5	3	70%
UKM	403	5	4	32%	Vehicle	846	18	4	26%	Vertebral	310	6	3	48%
WDBC	569	30	2	63%	WFRN	5456	4	4	40%	Wine	178	13	3	40%

because T_u controls the extent of membership shared between fuzzy clusters. The higher T_u, the more diffuse the obtained partition, and the closer the prototypes of each group, see Fig. 11. In contrast, the relevance weights of the variables in EFCM-GS2, EFCM-GS1, and EFCM-LS1 are controllable by entropy. In this regard, the weights of the variables are similar with higher T_v. However, small values reveal distinct relevance, see Fig. 12. Despite the fact that the ability to discriminate the relevance of the variables deteriorates with high T_v value, the models still manage to determine such importance, even for high values such as $T_v = 1000$.

In general, even though selecting the optimal value for the hyperparameters in an unsupervised manner is a difficult task, we noted that acceptable solutions could be obtained by avoiding values that lead to nearby centroids. Addressing such, we chose the optimal values for T_u and T_v without supervision, as described in Section 1. In this sense, we provide a new perspective for fuzzifying the clusterization of the units while ensuring maximum cluster compactness.

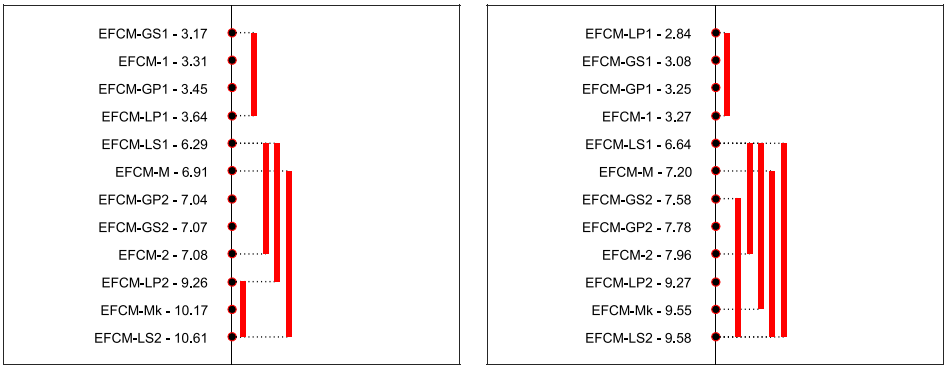
4.5. Brodatz texture images for segmentation

Image segmentation is a challenging but essential task in several imaging analyzes or computer vision applications [41]. Texture image segmentation plays an essential role in the human visual system for recognition and interpretation [42–44]. It aims to segment a texture image into several regions with different texture features, providing surface characteristics for analyzing many types of images, including natural scenes, remotely sensed data, and biomedical modalities.

This section assesses the performance and robustness of the proposed algorithms in texture image segmentation with and without noise. The images used were acquired from the Brodatz texture dataset [45], and are employed to demonstrate the performance of the algorithms for datasets with high dimensionality. The first and second rows of Fig. 13 show the six texture images with and without Gaussian noise (mean 0, variance 0.3). Moreover, the last row of Fig. 13 illustrates the corresponding ideal segmentation results used as a reference to quantitatively determine the segmentation performance. The images are synthesized with different types of texture images: two-textural images (D4

(a) Average performance ranking for 0% of outliers

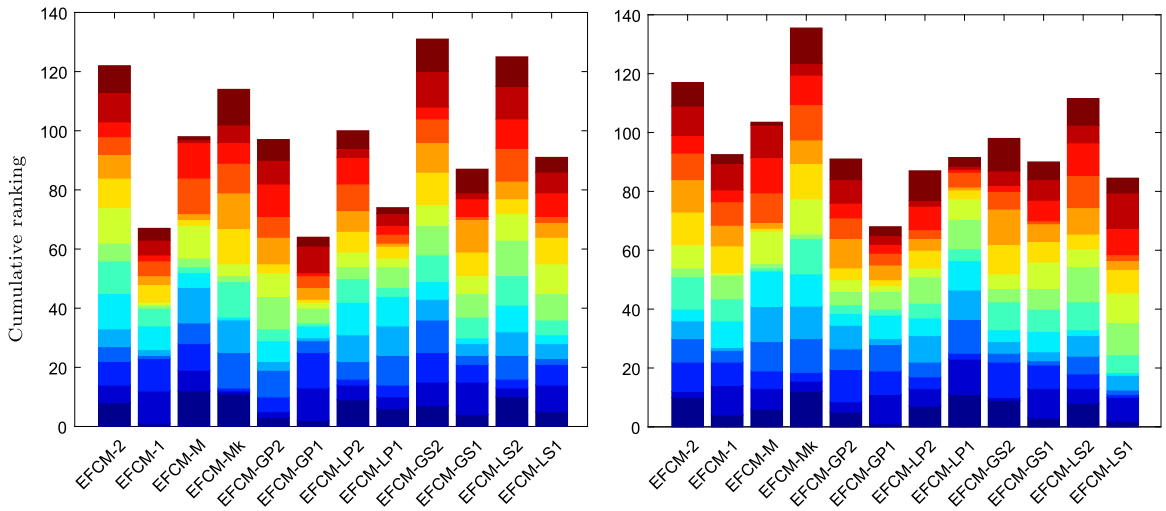
(b) Average performance ranking for 10% of outliers



(c) Average performance ranking for 20% of outliers

(d) Average performance ranking for 30% of outliers

Fig. 8. Comparison of the algorithms with each other with the Nemenyi test for *ARI* with different percentages of outliers.



(a) HUL ranking

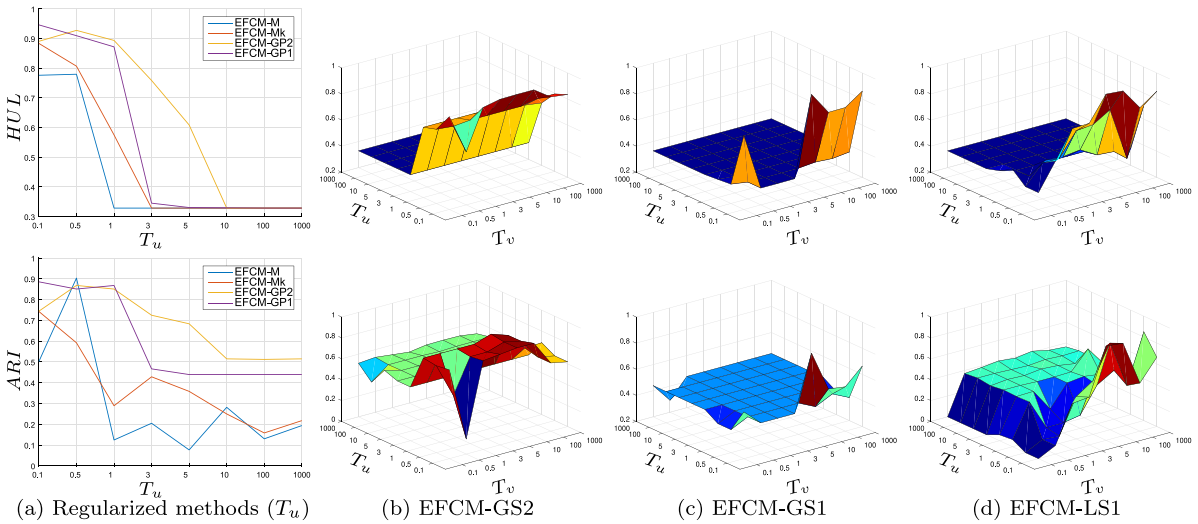
(b) ARI ranking

Fig. 9. Algorithms performance ranking for the real datasets according to the indices (a) *HUL* and (b)*ARI*. Table 9, supplementary material, shows the numeric values.

and D49), five-textural images (D21, D22, D49, D53, and D55), and seven-textural images (D3, D6, D21, D49, D53, D56, and D93).

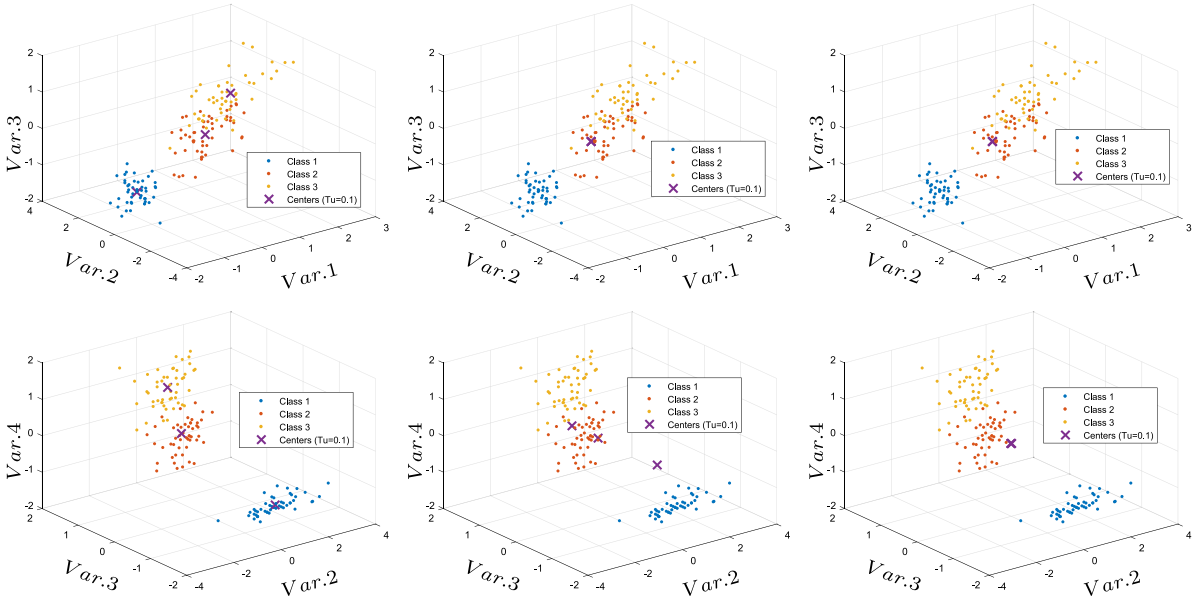
4.5.1. Experimental setting

The features of the texture images were extracted using the Gabor filter as in Ref. [46]. A filter bank with six orientations (every 30°) and five frequencies starting from 0.4 was created by extracting 30-dimensional features for every pixel of the 100 × 100 texture images filtered by the filter bank. After extracting the texture features, we used the algorithms to segment each texture image. The choice of the parameter values was obtained as in previous sections. However, we vary the values of T_u between 0.1 and 100 (with step 0.1). The algorithms were executed ten times on each dataset. We selected the best results according to their respective objective function. In the experiments, the number of



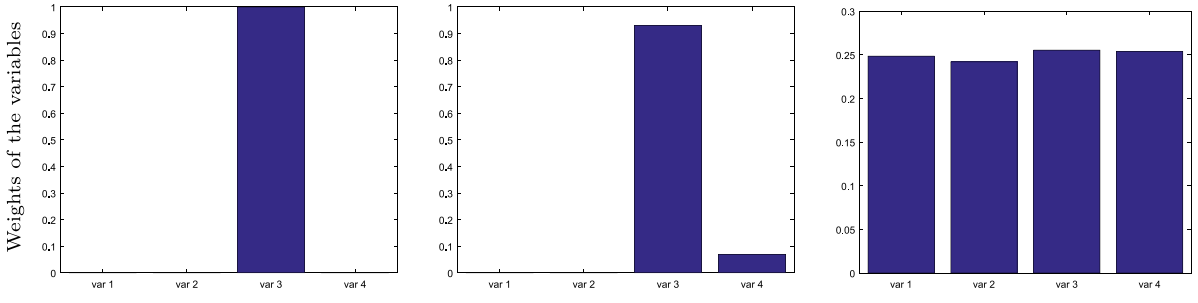
(a) Regularized methods (T_u) (b) EFCM-GS2 (c) EFCM-GS1 (d) EFCM-LS1

Fig. 10. Effect of the hyperparameters T_u and T_v on the Iris dataset. (a) performance of EFCM-M, EFCM-Mk, EFCM-GP2 and EFCM-GP1 for *HUL* and *ARI* with different values of T_u. (b), (c) and (d) performances of the algorithms EFCM-GS2, EFCM-GS1 and EFCM-LS1 varying T_u and T_v.



(a) Prototypes for $T_u = 0.1$ (b) Prototypes for $T_u = 3$ (c) Prototypes for $T_u = 1000$

Fig. 11. Prototypes obtained using the proposed method EFCM-GS2 on the Iris dataset for (a) $T_u = 0.1$. (b) $T_u = 3$ and (c) $T_u = 1000$.



(a) $T_v = 0.1$ (b) $T_v = 3$ (c) $T_v = 1000$

Fig. 12. Weights of the variables obtained using the proposed method EFCM-GS2 on the Iris dataset for $T_u = 1$ and (a) $T_v = 0.1$. (b) $T_v = 3$ and (c) $T_v = 1000$.

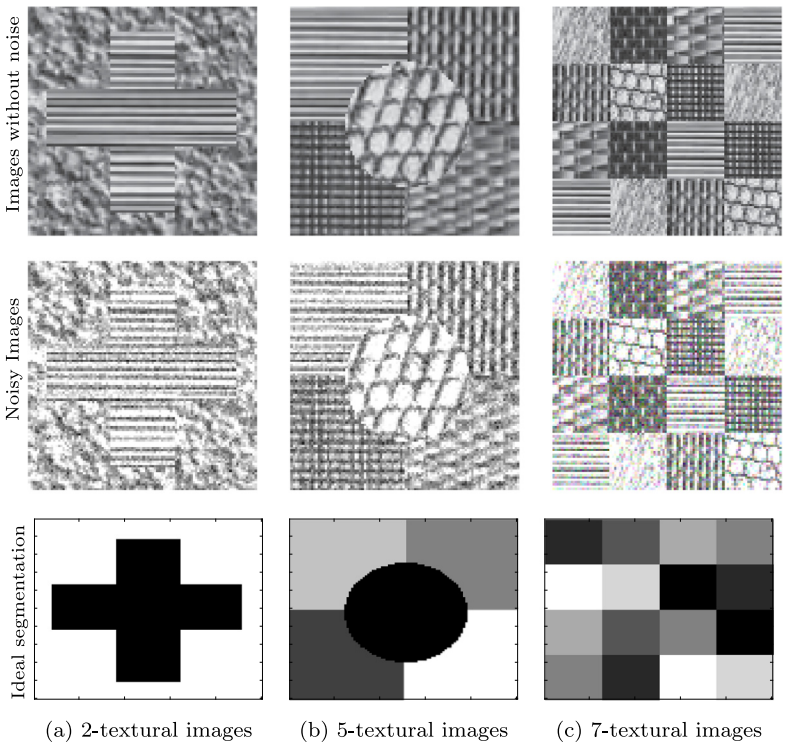


Fig. 13. Two, five, and seven-textural images with and without Gaussian noise.

clusters is assumed to be equal to 2, 5, and 7 in images with two, five, and seven textures, respectively.

4.5.2. Results

Tables 11 to 13 present the clustering results obtained by the algorithms on images without and with noise according to *HUL* and *ARI*, in addition to the time required to run the experiments for the ten iterations. Figures 14, 15, and 16, supplementary material, show the segmentation results in 2, 4, and 7 textural images, respectively, without and with Gaussian noise.

In this application, methods based on the City-Block distance generally degrade their performance more slowly than those with the Mahalanobis and Euclidean distances in a noisy environment. Tables 11 to 13 and Figures 14, 15, and 16, in the supplementary material, show that the EFCM-1 algorithm obtained higher values for *HUL* in the 2-textural image without noise. However, EFCM-LP1 and EFCM-GP1 achieved the second and third-best performances, respectively. Regarding *ARI* values, EFCM-LP1 and EFCM-GP1 outperformed the other approaches. The EFCM-LS2 algorithm presented the worst results for both indices. In the case of the 5-textural image, EFCM-LS2 showed a higher value of *HUL*, but EFCM-GP1 yielded the highest clustering result for *ARI* and produced better segmentation results. The worst performances were presented by EFCM-M and EFCM-GS1 for *HUL* and *ARI*, respectively. For 7-textural images, EFCM-M reached the best results regardless of the index considered. EFCM-GS1 and EFCM-LS1 obtained the worst performances for *HUL* and *ARI*, respectively.

Regarding images with Gaussian noise, as can be seen in Tables 11 to 13 and Figures 14, 15 and 16 in the supplementary material, EFCM-GP1 and EFCM-GP2 obtained the best results for 2 and 5-textural images, respectively, for both indexes. For 7-textural images according to *HUL*, EFCM-GP2 presented better clustering results, and EFCM-GP1 has the best performance according to *ARI*. EFCM-Mk achieved the worst results for 2-textural images for *HUL* and EFCM-LS2 for *ARI*. The algorithm EFCM-M had the worst performance regarding both indices for the five and 7-textural images.

The development of data structures and algorithms is motivated by the need for computer programs to run on machines as fast as possible for inputs of all sizes. The running time of an algorithm for a specific input depends, among other factors, on the number of operations executed. Tables 9 to 11 reveal that regularized methods present a low execution time. In this regard, despite the large dimensionality and number of objects in the datasets, the algorithms took less than 325 s to run through the ten iterations.

5. Conclusions

This paper proposed new fuzzy clustering algorithms based on suitable adaptive Euclidean, Mahalanobis, City-Block distances, and entropy regularization. These adaptive distances change at each iteration of the algorithms and can differ from one group to another. These dissimilarity measures are suitable for learning the weights of the variables during the clustering process, improving the performance of the algorithms.

The proposed algorithms are based on the minimization of clustering criteria, performed in three steps (representation, weighting, and assignment), providing a fuzzy partition, a representative for each fuzzy cluster, and a relevance weight for each variable or a matrix of weights. We consider two types of constraints to compute the relevance weights of the variables. The first type considers that the sum of the weights of the variables, or the sum of the weights of the variables per cluster, must be equal to one. In turn, the second type assumes that the product of the weights of the variables, or the product of the weights of the variables per cluster, must be equal to one.

The performance and usefulness of the proposed algorithms have been illustrated through experiments carried out on suitable synthetic and real datasets. Furthermore, the Friedman test was applied to explore the statistically significant differences in the experimental results. If the null hypothesis is rejected under

Table 11
Performance algorithms for the 2 textural images with and without noise.

Algorithms	Image without noise			Noisy image		
	HUL	*ARI*	Seconds	*HUL*	*ARI*	Seconds
EFCM-2	0.5547 (6)	0.7438 (4)	101.2	0.5694 (4)	0.5863 (7)	101.0
EFCM-1	**0.7853** (1)	0.7222 (7)	102.7	0.630 (2)	0.6328 (2)	124.0
EFCM-M	0.5398 (10)	0.0079 (10)	116.0	0.5392 (8)	0.0434 (8)	116.4
EFCM-Mk	0.5365 (11)	−0.0236 (11)	120.1	0.5195 (12)	0.0015 (9)	138.7
EFCM-GP2	0.5710 (5)	0.7438 (4)	164.1	0.5403 (5)	0.5866 (5.5)	164.8
EFCM-GP1	0.6980 (3)	0.7571 (2)	182.5	**0.6581** (1)	**0.6427** (1)	141.8
EFCM-LP2	0.5506 (7)	0.7266 (6)	117.7	0.5344 (10)	−0.0015 (11)	112.8
EFCM-LP1	0.7005 (2)	**0.7740** (1)	119.8	0.5759 (3)	0.5960 (4)	126.8
EFCM-GS2	0.5399 (9)	0.7438 (4)	114.5	0.5399 (7)	0.5866 (5.5)	124.1
EFCM-GS1	0.5925 (4)	0.7114 (8)	126.9	0.5403 (6)	0.6045 (3)	101.8
EFCM-LS2	0.5359 (12)	−0.0430 (12)	197.69	0.5361 (9)	−0.0375 (12)	110.1
EFCM-LS1	0.5420 (8)	0.1129 (9)	179.1	0.5215 (11)	−0.0001 (10)	97.4

Table 12
Performance algorithms for 5 textural images with and without noise.

Algorithms	Image without noise			Noisy image		
	HUL	*ARI*	Seconds	*HUL*	*ARI*	Seconds
EFCM-2	0.6484(5)	0.7572 (3)	114.5	0.4795 (9)	0.4523 (6)	124.99
EFCM-1	0.5780 (6)	0.7502 (4)	181.3	0.6164 (5)	0.5925 (4)	190.7
EFCM-M	0.2990 (12)	0.1906 (10)	192.0	0.2021 (12)	0.0678 (12)	188.4
EFCM-Mk	0.5558 (8)	0.1792 (11)	236.6	0.4318 (10)	0.1805 (10)	224.1
EFCM-GP2	0.6873 (3)	0.7740 (2)	245.9	**0.7592** (1)	**0.6730** (1)	190.0
EFCM-GP1	0.6487 (4)	**0.7826** (1)	244.2	0.6589 (4)	0.6515 (2)	260.5
EFCM-LP2	0.6964 (2)	0.4811 (8)	136.4	0.6910 (3)	0.3515 (9)	133.8
EFCM-LP1	0.5580 (7)	0.6054 (6)	204.2	0.7028 (2)	0.6102 (3)	168.5
EFCM-GS2	0.5160 (10)	0.4860 (7)	172.8	0.4844 (8)	0.4692 (5)	183.6
EFCM-GS1	0.5053 (11)	0.1230 (12)	157.2	0.5269 (7)	0.1018 (11)	171.6
EFCM-LS2	**0.7629** (1)	0.4379 (9)	197.7	0.4309 (11)	0.4246 (8)	191.7
EFCM-LS1	0.5368 (9)	0.6319 (5)	259.7	0.5923 (6)	0.4499 (7)	249.8

Table 13
Performance algorithms for 7 textural images with and without noise.

Algorithms	Image without noise			Noisy image		
	HUL	*ARI*	Seconds	*HUL*	*ARI*	Seconds
EFCM-2	0.7577 (3)	0.4716 (5)	123.5	0.6437 (4)	0.2894 (6)	123.4
EFCM-1	0.7225 (6)	0.5191 (2)	291.1	0.5830 (8)	0.3274 (2)	285.8
EFCM-M	**0.8765** (1)	**0.5231** (1)	282.6	0.1484 (12)	0.0390 (12)	222.6
EFCM-Mk	0.6561 (10)	0.2157 (10)	308.1	0.5412 (9)	0.1373 (9)	273.9
EFCM-GP2	0.8063 (2)	0.4981 (4)	228.8	**0.6964** (1)	0.3219 (3)	214.9
EFCM-GP1	0.6939 (7)	0.5051 (3)	337.8	0.6808 (2)	**0.4451** (1)	324.2
EFCM-LP2	0.6581 (9)	0.2632 (9)	147.3	0.6139 (7)	0.2076 (8)	147.4
EFCM-LP1	0.6862 (8)	0.4319 (8)	252.7	0.6608 (3)	0.3000 (4)	269.0
EFCM-GS2	0.7403 (5)	0.4666 (6)	252.0	0.4767 (10)	0.2912 (5)	300.2
EFCM-GS1	0.6003 (12)	0.1081 (11)	262.8	0.6418 (5)	0.0814 (10)	196.0
EFCM-LS2	0.7404 (4)	0.4353 (7)	219.5	0.4745 (11)	0.2873 (7)	189.9
EFCM-LS1	0.6267 (11)	0.0743 (12)	237.8	0.6264 (6)	0.0592 (11)	210.5

the Friedman test, the Nemenyi post-hoc test was used to determine which algorithms perform significantly differently. The proposed methods introduce regularization coefficients to control the membership degree of the objects and the relevance of the variables in the clustering task. We performed a sensitivity analysis of such hyperparameters to measure their influence on the quality of the clustering. Furthermore, an unsupervised process to compute their optimal value was introduced.

The experimental results showed that the proposed methods based on the Mahalanobis distance outperform other approaches when the variables are correlated. Furthermore, the proposed method with the global covariance matrix significantly outperformed other approaches in the third configuration for *ARI*, when the class covariance matrices are not diagonal but similar. Approaches with global dissimilarity functions had the highest results for datasets with the diagonal cluster covariance matrices and almost the same. However, local methods achieved better results for data with the cluster covariance matrices diagonal but unequal. Regarding datasets with outliers, the City-Block distance-based methods perform significantly better than other approaches under such conditions, regardless of the number of outliers.

Concerning the benchmark datasets, the proposed EFCM-GP1 algorithm presents the best average performance ranking. Moreover, the EFCM-1 and EFCM-LP1 algorithms achieved the second and third-best average ranking for *HUL* and EFCM-LS1 and EFCM-LP2 for *ARI*. The EFCM-GS2 algorithm obtained the worst results for *HUL* and EFCM-Mk for *ARI*.

Finally, all algorithms were executed on the Brodatz texture image dataset to examine the clustering performance and robustness for noiseless and noisy texture image segmentation. For noise-free images, the EFCM-1 algorithm obtained the best performance according to *HUL* and EFCM-LP1 for *ARI* on the 2-textural image. For 5-textural images, EFCM-LS2 showed a higher value of *HUL*, but EFCM-GP1 yielded the highest clustering result for *ARI* and produced better segmentation results. For 7-textural images, EFCM-M reached the best results regardless of the index considered. Moreover, concerning images with Gaussian noise, methods based on the City-Block distance generally degrade their performance more slowly than those with the Mahalanobis or

Euclidean distances. Furthermore, the proposed fuzzy weighting subspace clustering algorithms with global distances achieved the highest performances for both indices.

CRediT authorship contribution statement

Sara I.R. Rodríguez: Conceptualization, Methodology, Software, Data curation, Writing – review & editing. **Francisco de A.T. de Carvalho:** Supervision, Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank the anonymous referees for their careful revision, and Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE, Brazil (PBPG-0402-1.03/17) as well as Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, Brazil (311164/2020-0) for their financial support.

Appendix A. Proof of Proposition 1

The covariance matrix or the weights of the variables minimizing the proposed objective functions are calculated according to the adaptive distance function used:

(a) If the dissimilarity function is given by $(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k)$, the global covariance matrix $\mathbf{M}$ minimizing J_{EFCM-M} with $\det(\boldsymbol{M}) = 1$ is obtained with Eq. (19).

(b) If the adaptive distance function is given by $(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k)$, the local covariance matrices $\mathbf{M}_k$ that minimize $J_{EFCM-Mk}$ under $\det(\boldsymbol{M}_k) = 1$ are obtained with Eq. (18).

(c) If the adaptive distance function is given by $\sum_{j=1}^{V}(v_j)\, d(x_{ij}, g_{kj})$, then the vector of weights $v_j, (j = 1, \ldots, V)$ minimizing the criterion $J_{EFCM-GS}$ under $\epsilon\ [0, 1]\ \forall\, j$ and $\sum_{j=1}^{V}(v_j) = 1$ has its components in $v_j, (j = 1, \ldots, V)$ computed according to Eq. (21) if d is the Euclidean distance, else according to Eq. (22) if d is the City-Block distance.

(d) If the adaptive distance function is given by $\sum_{j=1}^{V}(v_j)\, d(x_{ij}, g_{kj})$, the vector of weights $\boldsymbol{v} = (v_1, \ldots, v_V)$ minimizing the criterion $J_{EFCM-GP}$ under $v_j > 0\ \forall\, j$ and $\prod_{j=1}^{V}(v_j) = 1$, has its components $v_j (j = 1, \ldots, V)$ computed according to Eq. (24) if d is the Euclidean distance and according to Eq. (25) if d is the City-Block distance.

(e) If the adaptive distance function is given by $\sum_{j=1}^{V}(v_{kj})\, |x_{ij} - g_{kj}|$, the vector of weights $\boldsymbol{v}_k = (v_{k1}, \ldots, v_{kV})$ minimizing the criterion $J_{EFCM-LS1}$ under $v_{kj} \geq 0\ \forall\, k, j$ and $\sum_{j=1}^{V}(v_{kj}) = 1$ has its components in $v_{kj} (k = 1, \ldots, C, j = 1, \ldots, V)$ computed as in Eq. (26).

Proof (a)**.** We wish to minimize J_{EFCM-M} regarding $\mathbf{M}$ under $\det(\boldsymbol{M}) = 1$. Let the Lagrangian function be:

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) + T_u\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) + \beta\,[1 - \det(\boldsymbol{M})] \tag{A.1}$$

Considering the derivative $\frac{\partial \mathcal{L}}{\partial \mathbf{M}}$ and using the identities $\frac{\partial(\mathbf{y}^T\mathbf{M}\mathbf{y})}{\partial \mathbf{M}} = \mathbf{y}\mathbf{y}^T$, $\frac{\partial \det(\mathbf{M})}{\partial \mathbf{M}} = \det(\mathbf{M})\mathbf{M}^{-1}$ that hold for a non-singular matrix $\mathbf{M}$ and any compatible vector $\mathbf{y}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{M}} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T - \beta\det(\boldsymbol{M})\boldsymbol{M}^{-1} = 0 \tag{A.2}$$

It follows that $\boldsymbol{M}^{-1} = \frac{Q}{\beta}$ where $Q = \sum_{k=1}^{C} C_k$ and $C_k = \sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T$ because $\det(\mathbf{M}) = 1$. As $\det(\mathbf{M^{-1}}) = \frac{1}{\det(\mathbf{M})} = 1$, from $\boldsymbol{M}^{-1} = \frac{Q}{\beta}$ it follows that $\det(\mathbf{M^{-1}}) = \frac{\det(Q)}{\beta^P} = 1$, then $\beta = (\det(Q))^{\frac{1}{V}}$. Moreover, as $\mathbf{M^{-1}} = \frac{Q}{\beta} = \frac{Q}{(\det(Q))^{\frac{1}{V}}}$, it follows that $\boldsymbol{M} = (\det(Q))^{\frac{1}{V}}Q^{-1}$.

An extremum value of J_{EFCM-M} is reached when $\boldsymbol{M} = (\det(Q))^{\frac{1}{V}}Q^{-1}$. This extremum value is $J_{EFCM-M}((\det(Q))^{\frac{1}{V}}Q^{-1}) = trace[Q(\det(Q))^{\frac{1}{V}}Q^{-1}] = V\det(Q)^{\frac{1}{V}}$. On the other hand, $J_{EFCM-M}(\mathbf{I}) = trace[Q\mathbf{I}] = trace[Q]$. As a positive definite symmetric matrix, $Q = \mathbf{P}\Lambda\mathbf{P}^T$ (according to the singular value decomposition procedure) where: $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}$, $\Lambda = diag(\varsigma_1, \ldots, \varsigma_V)$, and $\varsigma_j (j = 1, \ldots, V)$ are the eigenvalues of Q. Thus $J_{EFCM-M}(\mathbf{I}) = trace[\mathbf{P}\Lambda\mathbf{P}^T] = trace[\Lambda] = \sum_{j=1}^{V}\varsigma_j$. Moreover, $\det(Q) = \det(\mathbf{P}\Lambda\mathbf{P}^T) = \det(\Lambda) = \prod_{j=1}^{V}\varsigma_j$. As it is well known that the arithmetic mean is greater than the geometric mean, i.e., $(1/V)(\varsigma_1 + \cdots + \varsigma_V) > \{\varsigma_1 \times .. \times \varsigma_V\}^{1/V}$. Note that the equality holds only if $\varsigma_1 = \cdots = \varsigma_V$. Then, it follows that $J_{EFCM-M}(\mathbf{I}) > J_{EFCM-M}(\det(Q))^{\frac{1}{V}}Q^{-1}$. Thus, we conclude that this extreme is a minimum.

Remark. The matrix C_k is related to the fuzzy covariance matrix in the kth cluster, and therefore the matrix $\mathbf{M}$ is related to the pooled fuzzy covariance matrix.

(b) Following a reasoning similar to that of part (a), we conclude that $\mathbf{M}_k = [\det(C_k)]^{\frac{1}{V}}C_k^{-1}$ with $C_k = \sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T$.

(c) We want to minimize $J_{EFCM-LS1}$ with respect to $v_{kj}, (k = 1, \ldots, C, j = 1, \ldots, V)$ under $v_{kj} \in [0, 1]\ \forall j$ and $\sum_{j=1}^{V}(v_{kj}) = 1$. We use the Lagrangian multiplier to solve the unconstrained minimization problem in Eq. (12).

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\sum_{j=1}^{V}(v_{kj})|x_{ij} - g_{kj}| + T_u\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) + T_v\sum_{k=1}^{C}\sum_{j=1}^{V}(v_{kj})\ln(v_{kj}) - \sum_{k=1}^{C}\gamma_k\left[\sum_{j=1}^{V}(v_{kj}) - 1\right] \tag{A.3}$$

Taking the partial derivative of $\mathcal{L}$ in Eq. (A.3) with respect to v_{kj} and setting the gradient to zero we have Eq. (A.4). Then, from Eq. (A.4), is obtained Eq. (A.5).

$$\frac{\partial \mathcal{L}}{\partial v_{kj}} = \sum_{i=1}^{P}(u_{ik})|x_{ij} - g_{kj}| + T_v(\ln(v_{kj}) + 1) - \gamma_k = 0 \tag{A.4}$$

$$v_{kj} = \exp\{\frac{\gamma_k}{T_v} - 1\}\exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\} \tag{A.5}$$

Substituting Eq. (A.4) in $\sum_{w=1}^{V}(v_{kj}) = 1$, we have

$$\sum_{w=1}^{V}(v_{kw}) = \sum_{w=1}^{V}\exp\{\frac{\gamma_k}{T_v} - 1\}\exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\} = 1 \tag{A.6}$$

It follows that

$$\exp\{\frac{\gamma_k}{T_v} - 1\} = \frac{1}{\sum_{w=1}^{V}\exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{A.7}$$

Substituting Eq. (A.7) in Eq. (A.4), we obtain

$$v_{kj} = \frac{\exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{ij}-g_{kj}|}{T_v}\}}{\sum_{w=1}^{V}\exp\{-\frac{\sum_{i=1}^{P}(u_{ik})|x_{iw}-g_{kw}|}{T_v}\}} \tag{A.8}$$

Also we have that

$$\frac{\partial J_{EFCM-LS1}}{\partial v_{kj}} = \sum_{i=1}^{P}(u_{ik})|x_{ij}-g_{kj}| + T_v(\ln(v_{kj})+1) \tag{A.9}$$

The Hessian matrix of $J_{EFCM-LS1}$ with respect to $\mathbf{V}$ is:

$$\partial^2 J_{EFCM-LS1}(\mathbf{V}) = \begin{bmatrix} \frac{T_v}{v_{11}} & \dots & 0 \\ & \dots\dots & \\ 0 & \dots & \frac{T_v}{v_{CV}} \end{bmatrix}$$

Since we know $T_v > 0$ and $v_{kj} \geq 0$ (according to Eq. (26)), the Hessian matrix $\partial^2 J_{EFCM-LS1}(\mathbf{V})$ is positive definite, so that we can conclude that this extremum is a minimum.

(d) Following reasoning similar as in part (c), we conclude that

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{ij},g_{kj})}{T_v}\}}{\sum_{w=1}^{V} exp\{-\frac{\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})d(x_{iw},g_{kw})}{T_v}\}}$$

(e) We want to minimize $J_{EFCM-GP}$ with respect to v_j, $(k = 1,\dots,C)$, under $v_j > 0\ \forall\ j$ and $\prod_{j=1}^{V}(v_j) = 1$. We use the Lagrangian multiplier to solve the unconstrained minimization problem in Eq. (6).

$$\mathcal{L} = \sum_{i=1}^{P}\sum_{k=1}^{C}\sum_{j=1}^{V}(u_{ik})(v_j)d(x_{ij},g_{kj}) + T_u\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) - \gamma\left[\prod_{j=1}^{V}(v_j) - 1\right] \tag{A.10}$$

Taking the partial derivative of $\mathcal{L}$ in Eq. (A.10) with respect to v_j and setting the gradient to zero, we have Eq. (A.11). Then, from Eq. (A.11), Eq. (A.12) is obtained.

$$\frac{\partial \mathcal{L}}{\partial v_j} = \sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj}) - \frac{\gamma}{v_j} = 0 \tag{A.11}$$

$$v_j = \frac{\gamma}{\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj})} \tag{A.12}$$

Substituting Eq. (A.12) in $\prod_{j=1}^{V}(v_j) = 1$, we have Eq. (A.13). Then after some algebra, Eq. (A.14) is obtained.

$$\prod_{h=1}^{V}(v_h) = \prod_{h=1}^{V}\frac{\gamma}{\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})} = 1 \tag{A.13}$$

$$\gamma = \{\prod_{h=1}^{V}\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})\}^{\frac{1}{V}} \tag{A.14}$$

Substituting Eq. (A.14) in Eq. (A.12) we obtain

$$v_j = \frac{\{\prod_{h=1}^{V}\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})\}^{\frac{1}{V}}}{\sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj})} \tag{A.15}$$

If we rewrite the criterion $J_{EFCM-GP}$ as $J(v_1,\dots,v_V) = \sum_{j=1}^{V} v_j J_j$ where $J_j \sum_{i=1}^{P}(u_{ik})\, d(x_{ij},g_{kj})$ and $T_u \sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik})$ is seen like a constant. Thus, an extreme value of J is reached when $J(v_1,\dots,v_V) = V\{J_1,\dots,J_V\}^{\frac{1}{V}}$. As $J(1,\dots,1) = \sum_{j=1}^{V} J_j = J_1 + \dots + J_V$, and as it is well known that the arithmetic mean is greater than the geometric mean, i.e., $\frac{1}{V}\{J_1+\dots+J_V\} > \{J_1\times\dots\times J_V\}^{\frac{1}{V}}$, (the equality holds only if $J_1 = \dots = J_V$), we conclude that this extremum is a minimum. Thus, Proposition 1 was proved. □

Appendix B. Proof of Proposition 2

Proof. We want to minimize the clustering criterion with respect to u_{ik} under $u_{ik} \in [0,1]$ and $\sum_{k=1}^{C}(u_{ik}) = 1$.

(a) If the adaptive distance function is given by $(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)$ and we want to minimize J_{EFCM-M} with respect to u_{ik} under $u_{ik} \in [0,1]$ and $\sum_{k=1}^{C}(u_{ik}) = 1$. Let the Lagrangian function be:

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k) + T_u\sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\ln(u_{ik}) - \sum_{i=1}^{P}\lambda_i\left[\sum_{k=1}^{C}(u_{ik}) - 1\right] \tag{B.1}$$

Taking the partial derivative of $\mathcal{L}$ with respect to u_{ik} and setting the gradient to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial u_{ik}} = (\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k) + T_u(\ln(u_{ik})+1) - \lambda_i = 0 \tag{B.2}$$

From Eq. (B.2) is obtained:

$$u_{ik} = \exp\{\frac{\lambda_i}{T_u} - 1\}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\} \tag{B.3}$$

If $\sum_{w=1}^{C}(u_{iw}) = 1$ then:

$$\sum_{w=1}^{C}\exp\{\frac{\lambda_i}{T_u} - 1\}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\} = 1 \tag{B.4}$$

From Eq. (B.4) we obtain Eq. (B.5). Then, substituting Eq. (B.5) in Eq. (B.3) we have Eq. (B.6).

$$\exp\{\frac{\lambda_i}{T_u} - 1\} = \frac{1}{\sum_{w=1}^{C}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\}} \tag{B.5}$$

$$u_{ik} = \frac{\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\}}{\frac{1}{\sum_{w=1}^{C}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\}}} \tag{B.6}$$

Additionally, we know that:

$$\frac{\partial J_{EFCM-M}}{\partial u_{ik}} = (\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k) + T_u(\ln(u_{ik})+1) \quad \text{and} \quad \frac{\partial^2 J_{EFCM-M}}{\partial u_{ik}} = \frac{T_u}{u_{ik}} \tag{B.7}$$

The Hessian matrix of J_{EFCM-M} according to $\mathbf{U}$ is:

$$\partial^2 J_{EFCM-M}(\mathbf{U}) = \begin{bmatrix} \frac{T_u}{u_{11}} & \dots & 0 \\ & \dots\dots & \\ 0 & \dots & \frac{T_u}{u_{PC}} \end{bmatrix}$$

Since $T_u > 0$ and $u_{ik} \geq 0$, the Hessian matrix $\partial^2 J_{EFCM-M}(\mathbf{U})$ is positive definite, it is possible to conclude that such extremum is a minimum. The matrix of membership degree of the objects into the fuzzy clusters for the other proposed approaches is obtained similarly as in part (a). Thus, Proposition 2 was proved. □

Appendix C. Proof of Proposition 3

(i) The series $u^{(t)}_{EFCM-M} = J_{EFCM-M}(v^{(t)}_{EFCM-M}) = J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;
(ii) The series $u^{(t)}_{EFCM-Mk} = J_{EFCM-Mk}(v^{(t)}_{EFCM-Mk}) = J_{EFCM-Mk}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;
(iii) The series $u^{(t)}_{EFCM-GS} = J_{EFCM-GS}(v^{(t)}_{EFCM-GS}) = J_{EFCM-GS}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;
(iv) The series $u^{(t)}_{EFCM-GP} = J_{EFCM-GP}(v^{(t)}_{EFCM-GP}) = J_{EFCM-GP}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;
(v) The series $u^{(t)}_{EFCM-LS1} = J_{EFCM-LS1}(v^{(t)}_{EFCM-LS1}) = J_{EFCM-LS1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;

Proof.

(i) The series $u^{(t)}_{EFCM-M} = J_{EFCM-M}(v^{(t)}_{EFCM-M}) = J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ decreases at each iteration and converges;

The objective function J_{EFCM-M} measures the heterogeneity of the partition as the sum of the heterogeneity in each cluster. We will first show that the inequalities (I), (II) and (III) below hold (i.e., the series decreases at each iteration).

$$\underbrace{J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)})}_{u^{(t)}_{EFCM-M}} \overbrace{\geq}^{(I)} J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)})$$
$$\overbrace{\geq}^{(II)} J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t+1)}, \mathbf{U}^{(t)}) \overbrace{\geq}^{(III)}$$
$$\underbrace{J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t+1)}, \mathbf{U}^{(t+1)})}_{u^{(t+1)}_{EFCM-M}}$$

The inequality (I) holds because $J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{g}^{(t)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$ and $J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$, and according to Section 3.1.1,

$$\mathbf{G}^{(t+1)=(\mathbf{g}^{(t+1)}_1, \ldots, \mathbf{g}^{(t+1)}_C)} = \underbrace{\operatorname{argmin}}_{\mathbf{G}=(\mathbf{g}_1, \ldots, \mathbf{g}_C)\in\mathbb{L}^C} \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$$

Moreover, inequality (II) holds because $J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t+1)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t+1)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$ and according to Proposition 1,

$$\mathbf{M}^{(t+1)} = \underbrace{\operatorname{argmin}}_{\mathbf{M}\in\mathbb{M}} \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik})\, d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$$

The inequality (III) also holds because $J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t+1)}, \mathbf{U}^{(t+1)}) = \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t+1)}_{ik})\, d_{\mathbf{M}^{(t+1)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u^{(t+1)}_{ik}) \ln(u^{(t+1)}_{ik})$ and according to Proposition 2,

$$\mathbf{U}^{(t+1)} = (\mathbf{u}^{(t+1)}_1, \ldots, \mathbf{u}^{(t+1)}_P) = \underbrace{\operatorname{argmin}}_{\mathbf{U}=(\mathbf{u}_1, \ldots, \mathbf{u}_P)\in\mathbb{U}^P} \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik})\, d_{\mathbf{M}^{(t+1)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{P}(u_{ik}) \ln(u_{ik})$$

Finally, since the series $u^{(t)}_{EFCM-M}$ decreases and it is bounded ($J(v^{(t)}_{EFCM-M}) \geq 0$), it converges.

The proof of the convergence of the series $u^{(t)}_{EFCM-Mk}, t = 0, 1, \ldots,$ $u^{(t)}_{EFCM-GS}, t = 0, 1, \ldots,$ $u^{(t)}_{EFCM-GP}, t = 0, 1, \ldots$ and $u^{(t)}_{EFCM-LS1}, t = 0, 1, \ldots$ proceeds similarly to the proof of the convergence of the series $u^{(t)}_{EFCM-M}, t = 0, 1, \ldots$ that has been just presented. □

Appendix D. Proof of Proposition 4

(i) The series $v^{(t)}_{EFCM-M} = (\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges;
(ii) The series $v^{(t)}_{EFCM-Mk} = (\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges;
(iii) The series $v^{(t)}_{EFCM-GS} = (\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges;
(iv) The series $v^{(t)}_{EFCM-GP} = (\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges;
(v) The series $v^{(t)}_{EFCM-LS1} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges.

Proof.

(i) The series $v^{(t)}_{EFCM-M} = (\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ converges;

Assuming that the stationarity of the series $u^{(t)}_{EFCM-M}$ is achieved in the iteration $t = T$, then, we have $u^{(T)}_{EFCM-M} = u^{(T+1)}_{EFCM-M}$ and then $J_{EFCM-M}(v^{(T)}_{EFCM-M}) = J_{EFCM-M}(v^{(T+1)}_{EFCM-M})$.

From $J_{EFCM-M}(v^{(T)}_{EFCM-M}) = J_{EFCM-M}(v^{(T+1)}_{EFCM-M})$ we arrive at $J_{EFCM-M}(\mathbf{G}^{(T)}, \mathbf{M}^{(T)}, \mathbf{U}^{(T)}) = J_{EFCM-M}(\mathbf{G}^{(T+1)}, \mathbf{M}^{(T+1)}, \mathbf{U}^{(T+1)})$. This equality, according to Proposition 4, can be rewritten as the equalities (I)–(III):

$$\underbrace{J_{EFCM-M}(\mathbf{G}^{(T)}, \mathbf{M}^{(T)}, \mathbf{U}^{(T)})}_{u^{(T)}_{EFCM-M}} \overbrace{=}^{(I)} J_{EFCM-M}(\mathbf{G}^{(T+1)}, \mathbf{M}^{(T)}, \mathbf{U}^{(T)})$$
$$\overbrace{=}^{(II)} J_{EFCM-M}(\mathbf{G}^{(T+1)}, \mathbf{M}^{(T+1)}, \mathbf{U}^{(T)}) \overbrace{=}^{(III)} J_{EFCM-M}(\mathbf{G}^{(T+1)}, \mathbf{M}^{(T+1)}, \mathbf{U}^{(T+1)})$$

From the first equality (I), the result is $\mathbf{G}^{(T)} = \mathbf{G}^{(T+1)}$, since $\mathbf{G}$ is unique, minimizing J_{EFCM-M} when the fuzzy partition represented by $\mathbf{U}^{(T)}$ and the matrix $\mathbf{M}^{(T)}$ are maintained fixed. From the second equality (II), the result is $\mathbf{M}^{(T)} = \mathbf{M}^{(T+1)}$ because $\mathbf{M}$ is unique, minimizing J_{EFCM-M}, when the fuzzy partition represented by $\mathbf{U}^{(T)}$ and the matrix of prototypes $\mathbf{G}^{(T+1)}$ are maintained fixed. Furthermore, from the third equality (III), the result is $\mathbf{U}^{(T)} = \mathbf{U}^{(T+1)}$ since $\mathbf{U}$ is unique minimizing J_{EFCM-M} when the prototypes $\mathbf{G}^{(T+1)}$ and the matrix $\mathbf{M}^{(T+1)}$ are maintained fixed.

Therefore, it can be concluded that $v^{(T)}_{EFCM-M} = v^{(T+1)}_{EFCM-M}$, which stands for all $t \geq T$ and $v^{(t)}_{EFCM-M} = v^{(T)}_{EFCM-M}, \forall t \geq T$ and follows that the series $v^{(t)}_{EFCM-M}$ converges.

The proof of the convergence of the series $v^{(t)}_{EFCM-Mk}, t = 0, 1, \ldots,$ $v^{(t)}_{EFCM-GS}, t = 0, 1, \ldots,$ $v^{(t)}_{EFCM-GP}, t = 0, 1, \ldots$ and $v^{(t)}_{EFCM-LS1}, t = 0, 1, \ldots$ proceeds similarly to the proof of the convergence of the series $v^{(t)}_{EFCM-M}$ presented above. □

Appendix E. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.107922.

References

[1] T.C. Havens, J.C. Bezdek, C. Leckie, L.O. Hall, M. Palaniswami, Fuzzy C-means algorithms for very large data, IEEE Trans. Fuzzy Syst. 20 (6) (2012) 1130–1146.

[2] J. Wu, H. Xiong, C. Liu, J. Chen, A generalization of distance functions for fuzzy C-means clustering with centroids of arithmetic means, IEEE Trans. Fuzzy Syst. 20 (3) (2011) 557–571.

[3] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, SIAM, 2020.

[4] A. Namburu, S. kumar Samay, S.R. Edara, Soft fuzzy rough set-based MR brain image segmentation, Appl. Soft Comput. 54 (2017) 456–466.

[5] Y. Tang, F. Ren, W. Pedrycz, Fuzzy C-means clustering through SSIM and patch for image segmentation, Appl. Soft Comput. 87 (2020) 105928.

[6] A. Patil, C. Patil, R. Karhe, M. Aher, et al., Comparative study of different clustering algorithms, Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 3 (7) (2014) 10490–10497.

[7] A.M. Ali, G.C. Karmakar, L.S. Dooley, Review on fuzzy clustering algorithms, J. Adv. Comput. 2 (3) (2008) 169–181.

[8] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction To Cluster Analysis, vol. 344, John Wiley & Sons, 2009.

[9] M.-C. Chiu, Y.-W. Hsu, Using fuzzy C-means clustering based on integration of psychological and physiological data for therapeutic music design, J. Ind. Prod. Eng. 34 (5) (2017) 382–397.

[10] M.R. Mahmoudi, D. Baleanu, Z. Mansor, B.A. Tuan, K.-H. Pho, Fuzzy clustering method to compare the spread rate of Covid-19 in the high risks countries, Chaos Solitons Fractals 140 (2020) 110230.

[11] J. Nayak, B. Naik, H. Behera, Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014, Comput. Intell. Data Min. 2 (2015) 133–149.

[12] E.H. Ruspini, J.C. Bezdek, J.M. Keller, Fuzzy clustering: A historical perspective, IEEE Comput. Intell. Mag. 14 (1) (2019) 45–55.

[13] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer Science & Business Media, 2013.

[14] X. Tao, R. Wang, R. Chang, C. Li, Density-sensitive fuzzy kernel maximum entropy clustering algorithm, Knowl.-Based Syst. 166 (2019) 42–57.

[15] R.-P. Li, M. Mukaidono, A maximum-entropy approach to fuzzy clustering, in: Proceedings of 1995 IEEE International Conference on Fuzzy Systems, 4, IEEE, 1995, pp. 2227–2232.

[16] M. Sadaaki, M. Masao, Fuzzy C-Means as a regularization and maximum entropy approach, in: Proceedings of the 7th International Fuzzy Systems Association World Congress (IFSA'97), vol. 2, 1997, 86–92.

[17] R. Coppi, P. D'Urso, Fuzzy unsupervised classification of multivariate time trajectories with the Shannon entropy regularization, Comput. Statist. Data Anal. 50 (6) (2006) 1452–1477.

[18] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, S. Wang, A survey on soft subspace clustering, Inform. Sci. 348 (2016) 84–106.

[19] L. Zhu, L. Cao, J. Yang, J. Lei, Evolving soft subspace clustering, Appl. Soft Comput. 14 (2014) 210–228.

[20] J. Wang, Z. Deng, K.-S. Choi, Y. Jiang, X. Luo, F.-L. Chung, S. Wang, Distance metric learning for soft subspace clustering in composite kernel space, Pattern Recognit. 52 (2016) 113–134.

[21] L. Chen, S. Wang, K. Wang, J. Zhu, Soft subspace clustering of categorical data with probabilistic distance, Pattern Recognit. 51 (2016) 322–332.

[22] X. Wang, Y. Wang, L. Wang, Improving fuzzy C-means clustering based on feature-weight learning, Pattern Recognit. Lett. 25 (10) (2004) 1123–1132.

[23] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Eew-SC: Enhanced entropy-weighting subspace clustering for high dimensional gene expression data clustering analysis, Appl. Soft Comput. 11 (8) (2011) 4798–4806.

[24] M. Hanmandlu, O.P. Verma, S. Susan, V.K. Madasu, Color segmentation by fuzzy co-clustering of chrominance color features, Neurocomputing 120 (2013) 235–249.

[25] S.I.R. Rodríguez, F.A.T. de Carvalho, Fuzzy clustering algorithm with automatic variable selection and entropy regularization, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[26] S.I.R. Rodríguez, F.A.T. de Carvalho, Fuzzy clustering algorithm based on adaptive city-block distance and entropy regularization, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2018, pp. 1–8.

[27] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognit. 43 (3) (2010) 767–781.

[28] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: 1978 IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes, IEEE, 1979, pp. 761–766.

[29] J.Z. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in K-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.

[30] K. Jajuga, L1-norm based fuzzy clustering, Fuzzy Sets and Systems 39 (1) (1991) 43–50.

[31] R.-P. Li, M. Mukaidono, Gaussian clustering method based on maximum-fuzzy-entropy interpretation, Fuzzy Sets and Systems 102 (2) (1999) 253–258.

[32] E. Diday, J. Simon, Clustering analysis, in: Digital Pattern Recognition, Springer, 1976, pp. 47–94.

[33] V. Schwämmle, O.N. Jensen, A simple and fast method to determine the parameters for fuzzy C-means cluster analysis, Bioinformatics 26 (22) (2010) 2841–2848.

[34] E. Hullermeier, M. Rifqi, S. Henzgen, R. Senge, Comparing fuzzy partitions: A generalization of the rand index and related measures, IEEE Trans. Fuzzy Syst. 20 (3) (2011) 546–556.

[35] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1) (1985) 193–218.

[36] F.A.T. de Carvalho, C.P. Tenório, N.L. Cavalcanti Junior, Partitional fuzzy clustering methods based on adaptive quadratic distances, Fuzzy Sets and Systems 157 (21) (2006) 2833–2857.

[37] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (200) (1937) 675–701.

[38] B. Nemenyi, Distribution-free multiple comparison, (Ph.D. thesis), Princeton University, 1963.

[39] P. D´ Urso, L. De Giovanni, R. Massari, Trimmed fuzzy clustering for interval-valued data, Advances in Data Analysis and Classification 9 (1) (2015) 21–40.

[40] K. Bache, M. Lichman, UCI machine learning repository, 2013, URL http://archive.ics.uci.edu/ml.

[41] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognit. 26 (9) (1993) 1277–1294.

[42] G. Akbarizadeh, M. Rahmani, Efficient combination of texture and color features in a new spectral clustering method for PolSAR image segmentation, National Acad. Sci. Lett. 40 (2) (2017) 117–120.

[43] P. Qian, K. Zhao, Y. Jiang, K.-H. Su, Z. Deng, S. Wang, R.F. Muzic Jr., Knowledge-leveraged transfer fuzzy C-means for texture image segmentation with self-adaptive cluster prototype matching, Knowl.-Based Syst. 130 (2017) 33–50.

[44] H. Du, Z. Wang, D. Wang, X. Wang, Multi-feature fusion method applied in texture image segmentation, in: 2018 14th International Conference on Computational Intelligence and Security (CIS), IEEE, 2018, pp. 135–139.

[45] T. Randen, Brodatz Texture, [Online; accessed July, 2021], http://www.ux.uis.no/~tranden/brodatz.html.

[46] V. Kyrki, J.-K. Kamarainen, H. Kälviäinen, Simple Gabor feature space for invariant object recognition, Pattern Recognit. Lett. 25 (3) (2004) 311–318.