Universidade Federera de Pernambuco
Centro de Informática
Programa de Pós-Graduação em Ciência da Computação

Sara Inés Rizo Rodríguez

Clustering algorithms with new automatic variables weighting

Recife
2022

Sara Inés Rizo Rodríguez

Clustering algorithms with new automatic variables weighting

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Advisor: Francisco de Assis Tenorio de Carvalho

Recife
2022

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

R627c Rizo Rodríguez, Sara Inés
Clustering algorithms with new automatic variables weighting / Sara Inés Rizo Rodríguez. – 2022.
135 f.: il., fig., tab.

Orientador: Francisco de Assis Tenório de Carvalho.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.
Inclui referências e apêndices.

1. Inteligência computacional. 2. Agrupamento. I. Carvalho, Francisco de Assis Tenório de (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2022 – 58

Sara Inés Rizo Rodríguez

“Clustering algorithms with new automatic variables weighting”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovado em: 21/02/2022.

_____

Orientador: Prof. Dr. Francisco de Assis Tenório de Carvalho

BANCA EXAMINADORA

_____

Prof. Dr. Cleber Zanchettin
Centro de Informática /UFPE

_____

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática /UFPE

_____

Prof. Dr. Tsang Ing Ren
Centro de Informática /UFPE

_____

Profa. Dra. Heloísa de Arruda Camargo
Departamento de Computação / UFSCar

_____

Prof. Dr. Carlos e Vinicius Layter Xavier
Departamento de Engenharia de Sistemas e Computação / UERJ

ABSTRACT

Every day a large amount of information is stored or represented as data for further analysis and management. Data analysis plays an indispensable role in understanding different phenomena. One of the vital means of handling these data is to classify or group them into a set of categories or clusters. Clustering or cluster analysis aims to divide a collection of data items into clusters given a measure of similarity. Clustering has been used in various fields, such as image processing, data mining, pattern recognition, and statistical analysis. Usually, clustering methods deal with objects described by real-valued variables. Nevertheless, this representation is too restrictive for representing complex data, such as lists, histograms, or even intervals. Furthermore, in some problems, many dimensions are irrelevant and can mask existing clusters, e.g., groups may exist in different subsets of features. This work focuses on the clustering analysis of data points described by both real-valued and interval-valued variables. In this regard, new clustering algorithms have been proposed, in which the correlation and relevance of variables are considered to improve their performance. In the case of interval-valued data, we assume that the boundaries of the interval-valued variables have the same and different importance for the clustering process. Since regularization-based methods are robust for initializations, the proposed approaches introduce a 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. Experiments on synthetic and real datasets corroborate the robustness and usefulness of the proposed clustering methods.

Keywords: partitioning clustering; adaptive distances; robust clustering; interval-valued data analysis; regularized-based methods.

RESUMO

Todos os dias, uma grande quantidade de informações é armazenada ou representada como dados para posterior análise e gerenciamento. A análise de dados desempenha um papel indispensável na compreensão de diferentes fenômenos. Um dos meios vitais de lidar com esses dados é classificá-los ou agrupá-los em um conjunto de categorias ou grupos. O agrupamento ou análise de agrupamento visa dividir uma coleção de itens de dados em grupos, dada uma medida de similaridade. O agrupamento tem sido usado em vários campos, como processamento de imagens, mineração de dados, reconhecimento de padrões e análise estatística. Geralmente, os métodos de agrupamento lidam com objetos descritos por variáveis de valor real. No entanto, essa representação é muito restritiva para representar dados complexos, como listas, histogramas ou mesmo intervalos. Além disso, em alguns problemas, muitas dimensões são irrelevantes e podem mascarar os grupos existentes, por exemplo, os grupos podem existir em diferentes subconjuntos das variáveis. Este trabalho enfoca a análise de agrupamento de dados descritos por variáveis de valor real e de valor de intervalo. Nesse sentido, novos algoritmos de agrupamento foram propostos, nos quais a correlação e a relevância das variáveis são consideradas para melhorar o desempenho. No caso de dados com valor de intervalo, assumimos que a importância dos limites das variáveis com valor de intervalo pode ser a mesma ou pode ser diferente para o processo de agrupamento. Como os métodos baseados em regularização são robustos à inicializações, as abordagens propostas introduzem um termo de regularização para controlar o grau de pertinência dos objetos aos grupos. Essas regularizações são populares devido ao alto desempenho no agrupamento de dados em grande escala e baixa complexidade computacional. Esses algoritmos iterativos de três etapas fornecem uma partição difusa, um representante para cada grupo, e o peso de relevância das variáveis ou sua correlação, minimizando uma função objetivo adequada. Experimentos com conjuntos de dados sintéticos e reais corroboram a robustez e utilidade dos métodos de agrupamento propostos.

Palavras-chaves: agrupamento particional; distâncias adaptativas; agrupamento robusto; análise de dados intervalares; métodos baseados em regularização.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ARI	Adjusted Rand Index
EFCM-LS1	Entropy Fuzzy Clustering Method with Local Sum restriction and adaptive City-Block distance
EFCM-LP1	Entropy Fuzzy Clustering Method with Local Product restriction and adaptive City-Block distance
EFCM-GS1	Entropy Fuzzy Clustering Method with Global Sum restriction and adaptive City-Block distance
EFCM-GS2	Entropy Fuzzy Clustering Method with Global Sum restriction and adaptive Euclidean distance
EFCM-GP1	Entropy Fuzzy Clustering Method with Global Product restriction and adaptive City-Block distance
EFCM-GP2	Entropy Fuzzy Clustering Method with Global Product restriction and adaptive Euclidean distance
EFCM-M	Entropy Fuzzy Clustering Method with Global Mahalanobis distance
EFCM-Mk	Entropy Fuzzy Clustering Method with Local Mahalanobis distance
EFCM-LS2	Entropy Fuzzy Clustering Method with Local Sum restriction and adaptive Euclidean distance
EFCM-LP2	Entropy Fuzzy Clustering Method with Local Product restriction and adaptive Euclidean distance
EIFCM-LJ1	Entropy Fuzzy Clustering Method with Local Joint relevance of the interval-valued variables and adaptive City-Block distance
EIFCM-LJ2	Entropy Fuzzy Clustering Method with Local Joint relevance of the interval-valued variables and adaptive Euclidean distance
EIFCM-GJ1	Entropy Fuzzy Clustering Method with Global Joint relevance of the interval-valued variables and adaptive City-Block distance
EIFCM-GJ2	Entropy Fuzzy Clustering Method with Global Joint relevance of the interval-valued variables and adaptive Euclidean distance
EWSC	Entropy Weighting Subspace Clustering
FCM	Fuzzy C-Means
FWC	Feature Weighting Clustering
FWSC	Fuzzy Weighting Subspace Clustering
HCA	Hierarchical Clustering Algorithms

HSC	Hard Subspace Clustering
HUL	Hullermeier index
PC	Partition Coefficient
PE	Partition Entropy
SC	Subspace clustering
SDA	Symbolic Data Analysis
SSC	Soft Subspace Clustering

CONTENTS

1 INTRODUCTION

Clustering is one of the most useful unsupervised machine learning approaches. This technique groups similar objects together while separating dissimilar ones apart based on some similarity measure (HAVENS et al., 2012; CHEN et al., 2019). The higher the similarity within a group and the more significant the difference between clusters, the better or more distinct the clustering will be. Clustering algorithms are an efficient tool for image processing, data mining, pattern recognition, and statistical analysis (GAN; MA; WU, 2020a; ZHAO; LIU; FAN, 2015; NAMBURU; SAMAY; EDARA, 2017; TANG; REN; PEDRYCZ, 2020). The most popular clustering algorithms provide hierarchical and partitioning structures. Hierarchical methods deliver an output represented by a hierarchical structure of groups known as a dendrogram, i.e., a nested sequence of partitions of the input data. Partitioning approaches create a partition of the input data into a fixed number of clusters, typically by optimizing an objective function. An advantage of such 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 can be divided into hard and fuzzy approaches (also known as crisp and soft) (GAN; MA; WU, 2020b). The groups provided by hard partitioning clustering are disjoint and non-overlapping. However, fuzzy partitioning clustering (BEZDEK, 2013) generates a fuzzy partition in which an object belongs to all clusters with a specific membership degree. Therefore, fuzzy methods can be helpful when the boundaries between groups are ambiguous. Generally, each object is defined by a set of features of a single quantitative or qualitative value. However, this representation may not be adequate when analyzing a group rather than an individual. For example, if the variability or uncertainty inherent in the data is considered. Symbolic Data Analysis (SDA) (DIDAY, 1988) was introduced as a domain associated with multivariate analysis. It provides suitable methods and extends classical data analysis techniques to symbolic data (BOCK; DIDAY, 2012). In this case, variables can consider the variability present in the group of observations, individuals, objects. This means that a variable can assume values such as a set of categories, an ordered list of categories, a histogram, or an interval from a set of real numbers. The interest in developing methods, particularly for this last category, has recently increased. Therefore, this work deals with the analysis of objects described by both real-valued and interval-valued variables.

One of the best known clustering algorithms for real-valued variables are the K-Means (MACQUEEN et al., 1967) and Fuzzy C-Means (FCM) (BEZDEK, 2013). Based on these approaches, several methods have been proposed. Ref. (CHIU; HSU, 2017) used the FCM clustering based on an integration of psychological and physiological data for therapeutic music design. Mahmoudi et al. (MAHMOUDI et al., 2020) compare the spread rate of Covid-19 in high-risk countries using the fuzzy clustering method. Refs. (NAYAK; NAIK; BEHERA, 2015; RUSPINI; BEZDEK; KELLER, 2019) include a review of fuzzy clustering and its applications.

On the other hand, to manage symbolic data, Irpino and Verde (IRPINO; VERDE, 2008)

proposed a Wasserstein-based distance approach and showed its properties in the context of clustering techniques. For avoiding the disruptive effects of possible outliers, D'Urso et al. (D'URSO; GIOVANNI; MASSARI, 2015) suggested a fuzzy C-medoids method with a trimming rule. After discarding a fixed fraction of outlying data, the clustering procedure is applied to the data. The percentage of data discarded and, thus, not considered in the optimization problem is determined by combining a validity criterion with the trimming algorithm. Later, Leski et al. (LESKI et al., 2016) introduced a clustering approach combining the fuzzy C-medoids clustering with the robust ordered statistics using Huber's M-estimator. Also, Ref. (D'URSO et al., 2017) proposed the use of a robust metric based on the exponential distance in a framework of the fuzzy C-medoids clustering model for interval-valued data.

1.1 MOTIVATION

In fuzzy clustering, most algorithms have imposed a fuzziness coefficient m in the unknown degree of membership. This has been viewed as an artificial device, lacking a strong theoretical justification (LI; MUKAIDONO, 1995; COPPI; D'URSO, 2006). Besides, these types of approaches are sensitive to initial cluster centers, so performance tends to deteriorate in some cases, especially with incomplete data (TAO et al., 2019; SING; ADHIKARI; BASU, 2015). Consequently, a new line of research has been started. It is based on adopting regularization terms to be adjoined to the maximum internal homogeneity criterion (SADAAKI; MASAO, 1997). 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.

Another necessary point to consider when proposing new clustering methods is finding a suitable distance metric in feature space. Such is essential to determine the closeness between a pair of objects, an object and a cluster representative, or a pair of cluster representatives. Traditional methods assume that the variables are uncorrelated and equally relevant to the clustering process. For example, the Euclidean distance as a dissimilarity 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 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 to the formation of a particular cluster (DENG et al., 2016). Then, each dimension contributes differently to every cluster. In this regard, the subspaces can be identified by the weight values after the clustering. Such approaches can be divided into two main categories: fuzzy weighting subspace clustering and entropy weighting subspace clustering. A review can be found in Refs. (SOUZA; CARVALHO, 2004; CARVALHO; LECHEVALLIER, 2009; DENG et al., 2010; ZHU et al., 2014; PIMEN-

TEL; SOUZA, 2014; DENG et al., 2016). Most of these prototype-based clustering algorithms are based on the Euclidean distance for comparing objects and prototypes. However, such approaches are less robust to cope with outliers in the dataset, and they do not consider the correlation between variables.

1.2 OBJECTIVES

This work proposes new fuzzy clustering algorithms for real-valued and interval-valued data based on regularizations. The specific objectives of this research are:

1. Study of the state-of-the-art clustering algorithms for real-valued and interval-valued data.
2. Study of the state-of-the-art in clustering based on regularizations.
3. Study of the state-of-the-art of SSC methods.
4. Proposal and implementation of new objective functions and constraints, allowing recognizing clusters of different shapes and sizes.
5. Introduction of new robust dissimilarity measures that diminish the effect of the outliers during clustering.
6. Validation and comparison of the proposals with state-of-the-art algorithms.

1.3 RESEARCH QUESTIONS

This document is guided by some research questions to propose clustering algorithms capable of overcoming the problems mentioned in Chapter 1:

1. Several clustering algorithms have been proposed for real-valued data with regularizations. However, they do not consider the covariance of the variables. Multivariate methods based on the Mahalanobis distance may address this type of challenge. Then, will Mahalanobis distance-based approaches really be more appropriate in certain situations than other clustering methods?
2. Usually, clustering algorithms use the Euclidean distance as a dissimilarity measure. Despite the usefulness of these proposals in several problems, such approaches are sensitive to outliers in the dataset. So how can the detrimental effect of outliers be avoided? Also, in many of these algorithms, hyper-parameter settings are often required for cluster and subspace detection. However, one of the biggest problems is determining feasible values, especially for complicated datasets or unknown information. How to simplify the definition of the optimization problem to reduce the number of hyper-parameters?

3. Soft subspace clustering methods usually assume that the set of relevant variables is different for each group. However, global approaches are more appropriate for cases when the internal dispersion of the cluster is almost the same. In this situation, can approaches with the same set of relevant variables to all groups improve clustering results?

4. In the literature, interval-valued data clustering assigns the same relevance weight for lower and upper boundaries. As a result, a boundary that plays a minor role in the clustering task can still have an important relevance weight. However, some interval boundaries associated with the variables could be more or less relevant or even irrelevant. Based on this, is it possible that considering the relevance of each boundary, the performance of the algorithms may be improved? How?

5. Clusters are formed so that any two data objects within a group have a minimum distance value and any two data objects across different clusters have a maximum distance value. Usually, the dissimilarity measures in the clustering methods are based on the Euclidean distance because finding an algebraic solution for the minimizers of the objective function is relatively simple. What would be the advantage of using another type of distance to group interval-valued data?

6. Regularization is somewhat similar to the penalty method for optimization by treating the regularization function as a penalty. In general, it implies a modification of a given problem that is singular in some sense into a regular one. The singular problem is challenging to solve, but the latter problem is easier to handle. What are the advantages of proposing clustering methods based on regularizations?

1.4 RESEARCH HYPOTHESES

In order to present possible solutions for each research question in Section 1.3, it is presented in this section our respective research hypotheses:

1. In some cases, it is hypothesized that clustering algorithms based on Mahalanobis distance should outperform other approaches for problems with correlated variables. This is because the definition of the Mahalanobis spaces takes the correlation between random vectors into account. By standardizing the random vectors with their covariance matrix, the measures on the individuals are more reasonable and comparable. Also, methods with Mahalanobis distance are invariant to linear transformations.

2. It is hypothesized that introducing robust dissimilarity measures may decrease the sensitivity to outliers. These measures can be based on the City-Block distance since theoretical studies indicate that such approaches are robust to outliers. Furthermore, it is

hypothesized that introducing some constraints in the optimization problem can reduce the number of hyper-parameters to configure, facilitating the search to obtain good clustering results.

3. In contrast to soft subspace clustering, it is hypothesized that introducing new objective functions and constraints such that the set of relevant variables is the same for all clusters can be more appropriate when the internal dispersion of the groups is almost the same. Since this topic was not well explored in the literature, we offer a detailed background through experiments on synthetic datasets.

4. For cluster analysis of interval-valued data, it is hypothesized that with the introduction of new objective functions and constraints, taking into account the weights of the relevance of the lower and upper boundaries altogether, the clustering results may be improved. As a result, a boundary that plays a significant role compared to the other will have a higher relevance weight.

5. It is hypothesized that the performance of clustering methods can be improved in a noisy environment by proposing new dissimilarity measures based on City-Block distance. Details are shown through experiments with synthetic datasets.

6. By regularizing the clustering process, it is hypothesized that the methods will be less sensitive to initial clustering centers and show high clustering performance on large-scale data.

1.5 CONTRIBUTIONS

According to the previous hypotheses, the contributions of this proposal are summarized below.

1. Proposal of new partitioning clustering algorithms with regularizations for 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.

2. Introduction of new fuzzy objective functions and constraints for grouping real-valued data with global and local Mahalanobis distance and entropy regularization. As an advantage, the correlation between variables is considered, showing a robust behavior to linear transformations.

3. New optimization problems for interval-valued data clustering are proposed, i.e., new objective functions and constraints, for interval-valued data clustering, which allows

recognizing clusters of different shapes and sizes in each boundary. This setup allows us to minimize the subsequent problem by alternating three steps.

4. New adaptive dissimilarity functions are introduced that consider the joint weights of the relevance of the lower and upper boundaries of the interval-valued variables. As a result, a boundary that plays a significant role in the clustering task will have a higher relevance weight.

5. 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 for being one of the most used in the literature and on the City-Block distance for its robust performance in a noisy data environment.

6. Local (a different set of relevant variables per cluster) and global (the same set of relevant variables for all groups) adaptive distances are 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 importance 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 hyper-parameters, such as controlling the membership degree of objects.

7. The proposals add an entropy term that regularizes the clustering results to satisfy all the constraint conditions during the optimization process. 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.

8. Finally, this work 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 an algebraic solution to obtain the prototype minimizer in City-Block distance-based approaches, an algorithmic solution is presented, in addition to analyzing the convergence properties of the proposed algorithms.

1.6 THESIS STRUCTURE

Besides the Introduction chapter, this work is divided into four additional chapters organized in the following way:

- Chapter 2 shows a general description of the problem, where some concepts are presented to understand the research problem and the proposed methods.
- Real-valued data clustering with entropy regularization.
  - Section 3.2 presents some of the main studies in the literature related to this work and their limitations.
  - Section 3.3 introduces proposed fuzzy clustering algorithms based on adaptive Euclidean, Mahalanobis and City-block distances and entropy regularization.
  - Section 3.4 shows a set of experiments performed with both simulated and real datasets, to demonstrate the usefulness of proposed methods.
  - Section 3.5 offers some concluding remarks.
- Interval-valued data clustering
  - Section 4.2 reviews interval-valued data clustering works closely related to the proposed approaches.
  - Section 4.3 introduces the proposed approaches and discusses the order of how they implement the research hypotheses.
  - Section 4.4 describes experimental setups to allow future replication of the results. A set of experiments carried out with both synthetic and real datasets is also presented to demonstrate the effectiveness of the proposed methods.
  - Section 4.5 summarizes the main results for interval-valued data clustering.
- Chapter 5 summarizes the limitations and future work .

To building this work, part of the material published in the following articles was used:

1. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** Fuzzy clustering algorithms with distance metric learning and entropy regularization. Applied Soft Computing, 2021.
2. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** Soft subspace clustering of interval-valued data with regularizations. Knowledge-Based Systems, v. 227, p. 107191, 2021.
3. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** Clustering interval-valued data with automatic variables weighting. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019. p. 1-8.
4. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** A new fuzzy clustering algorithm for interval-valued data based on City-Block distance. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2019. p. 1-6.

5. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** Fuzzy clustering algorithm based on Adaptive Euclidean distance and Entropy Regularization for interval-valued data. In: 27th International Conference on Artificial Neural Networks (ICANN). Springer Cham, 2018. p. 695-705.

6. **Rodríguez, S.I.R.; De Carvalho, F.A.T.** Fuzzy clustering Algorithm based on Adaptive City-block distance and Entropy Regularization. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2018. p. 1-8.

For more information, see Appendix A.

2 CLUSTER ANALYSIS

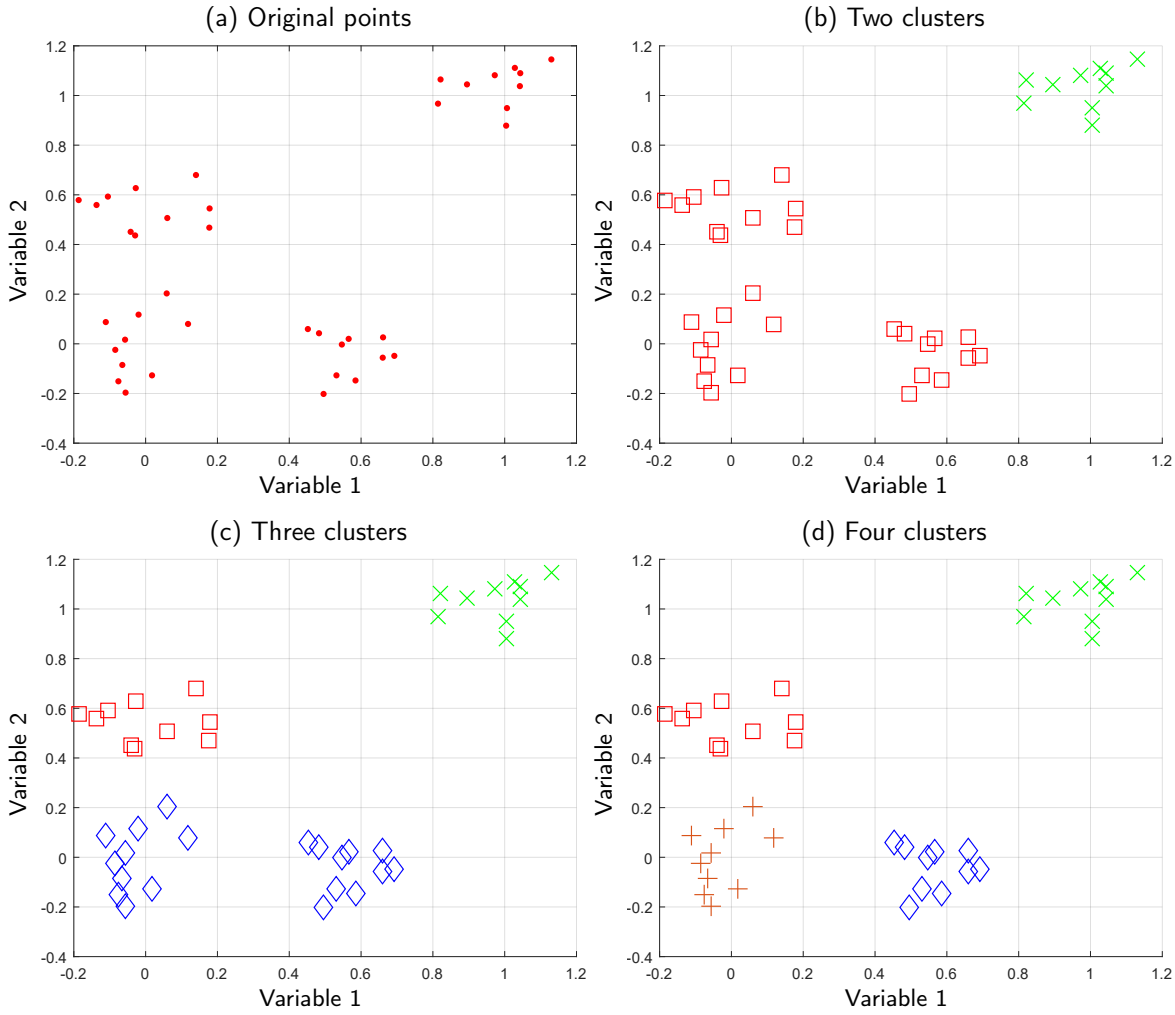
People collect and store a significant amount of information every day for further analysis and management. One of the main ways these data can be treated is by classifying or grouping them into a collection of categories or clusters. In order to learn a new item or understand a new phenomenon, people always try to find characteristics that can describe them and compare them with other known objects. This comparison is based on some similarity or dissimilarity, generalized as proximity according to specific standards or rules (XU; WUNSCH, 2005).

Clustering is an essential process for pattern recognition and machine learning that groups similar objects while separating dissimilar ones based on some similarity measure (HAVENS et al., 2012; WU et al., 2011). The purpose is that objects within a cluster are similar and different from items in other groups. The higher the similarity within a group and the more significant the difference between clusters, the better or more distinct the clustering will be. Intra-connectivity measures the density of connections between instances of a single cluster. A high value indicates a good clustering arrangement because the samples grouped within the same cluster are highly dependent. Besides, inter-connectivity is a measure of the connectivity between distinct clusters. A low degree of inter-connectivity is desirable because it indicates that individual groups are independent (KOTSIANTIS; PINTELAS, 2004).

Clustering is called unsupervised learning. In contrast to classification (supervised learning), there is no a priori labeling of some patterns to categorize others and inferring the cluster structure of all data. Every instance in the dataset is represented using the same set of attributes or variables. Consider $E = \{e_1, \ldots, e_N\}$ a set of N input objects. Each one $e_i\,(1 \leq i \leq N)$ is described by the vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{iP})$, with $1 \leq j \leq P$. Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be the dataset.

The clustering algorithms are classified according to the approach used to generate the clusters and the presented results (JAIN; MURTY; FLYNN, 1999). Hence, the choice of an algorithm depends on both the available data types and the desired application (see Figure 1). Clustering approaches can usually be divided into two main categories: hierarchical and partitional (KAUFMAN; ROUSSEEUW, 1987; KAUFMAN; ROUSSEEUW, 2009; GORUNESCU, 2011; CHEN et al., 2019; TANG et al., 2019). Hierarchical methods deliver an output represented by a hierarchical structure of groups known as a dendrogram. Partitioning methods create a partition of the input data into a fixed number of clusters, typically by optimizing an objective function. Latter algorithms have been used in several applications due to their simplicity and ease of implementation relative to other approaches.

Figure 1 – Clustering results for different numbers of clusters.



Source: Author (2022)

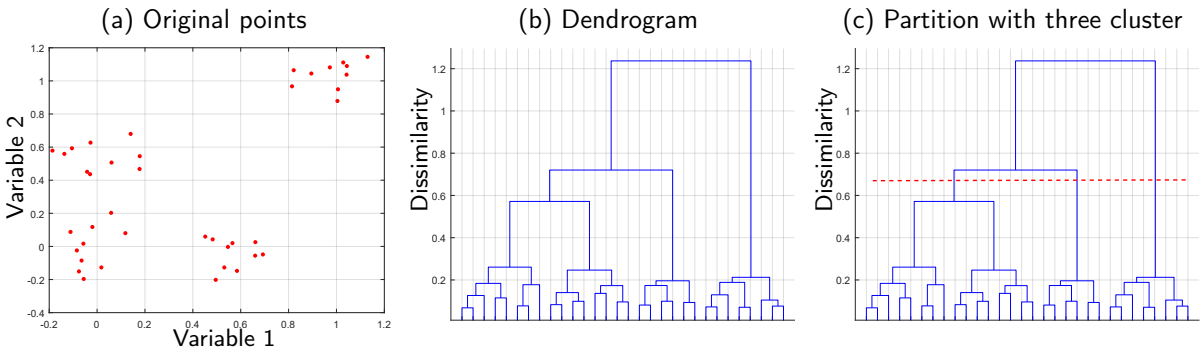
2.1 HIERARCHICAL CLUSTERING

Hierarchical clustering organizes the dataset (Figure 2 (a)) into a hierarchical structure according to a proximity matrix. The binary tree or dendrogram (Figure 2 (b)) usually depicts the results of Hierarchical Clustering Algorithms (HCA). The root node of the dendrogram represents the whole dataset, and each leaf node is regarded as a data object. The intermediate nodes describe the extent that the objects are proximal to each other. The height of a dendrogram usually expresses the distance between each pair of objects or clusters or an object and a group. Once the dendrogram is constructed, one can automatically choose a given number of clusters by splitting the tree at different levels to obtain different clustering solutions for the same dataset without rerunning the clustering algorithm (Figure 2 (c)). This representation provides very informative descriptions and visualization for the potential data clustering structures, mainly when real hierarchical relations exist in the data.

HCA can be categorized into major subclasses, considering how the final hierarchy is ob-

tained: agglomerative or divisive ((ANDERBERG, 2014)). In agglomerative clustering methods, each object is initially in a unique cluster, and the groups merge two at a time until all the observations belong to a final cluster. Divisive clustering starts with all observations in a single cluster and successively divides a group at each stage until a stopping criterion is reached. For a cluster with N objects, there are $2^{N-1} - 1$ possible two-subset divisions. Due to their high computational cost, divisive hierarchical clustering algorithms are rarely employed in the literature (XU; WUNSCH, 2008), with a few exceptions, e.g., (CAMPELLO; MOULAVI; SANDER, 2013). Figure 2 shows a HCA results, in which each leaf node of the dendrogram represents an object. The dissimilarities represented in the ordinate axis indicate the points at which a cluster is formed (agglomerative HCA) or dissolved (divisive HCA).

Figure 2 – (a) Original data. (b) HCA results represented as a dendrogram. (c) The same dendrogram with a cut (red dashed line) that produces a partition with three cluster.



Source: Author (2022)

2.2 PARTITIONING CLUSTERING

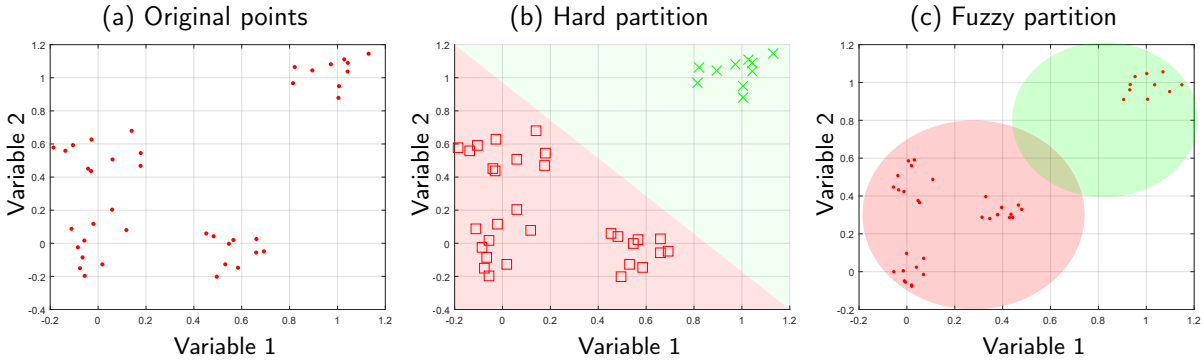
Partitioning clustering divides the input data into a fixed number of clusters based on either distance or density criteria computed on the dataset, typically by optimizing an objective function (XU; WUNSCH, 2005), please see Figure 3. Partitioning methods can manipulate large datasets, while for hierarchical approaches, building a dendrogram can be computationally impractical in some applications. Furthermore, since hierarchical methods only consider local neighbors at each step, they cannot incorporate a priori knowledge about clusters' global shape or size (FRIGUI; KRISHNAPURAM, 1999).

In general, partitioning methods differ in how to establish the best partition. They can be mainly divided in two different ways: hard and fuzzy. The groups provided by hard partitioning clustering (Figure 3 (b)) are disjoint and non-overlapping. In contrast, fuzzy approaches (Figure 3 (c)) generate a fuzzy partition such that an object belongs to all clusters with a specific membership degree (BEZDEK, 2013). This characteristic allows flexibility to express that objects belong to more than one cluster at the same time. Hence, fuzzy approaches are essential when the boundaries between groups are ambiguous. Each cluster is represented by a central

vector, which may not necessarily be a member of the dataset (centroid). The location of a center is also interpreted as the selection of an object as a representative object (or medoid) of a cluster. In the end, a representative element for each cluster, called hereafter a prototype, can be represented as $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_C)$.

The most popular partitioning methods are K-means (JAIN, 2010) and the FCM (BEZDEK, 2013). The K-means provides a hard partition $\mathcal{P} = \{P_1, \ldots, P_C\}$ of E into C disjoint clusters. However, the FCM returns a matrix of membership degrees of the objects for each cluster $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ with $\mathbf{u}_i = (u_{i1}, \ldots, u_{iC})$. These approaches partition the dataset into C subsets such that all points in a given subset are closest to the same center. If C is not known, different values of C can be evaluated until the most suitable one is found. The effectiveness of these approaches and others depends mainly on the function used to measure the distance between objects. The major difficulty is in finding a distance measure that works well with all types of data.

Figure 3 – Partitional clustering results. (a) Original dataset. (b) Hard clustering results with the number of clusters equal to two. (c) Fuzzy clustering results with the number of clusters equal to two.



Source: Author (2022)

2.3 DISTANCE MEASURES

Distance metrics play a significant role in measuring the similarity or dissimilarity among the data objects. In most applications, this measure is commonly based on distance functions such as the Euclidean distance, Manhattan distance, Minkowski distance, among others. Then, the clusters are formed so that any two objects within a cluster have a minimum distance value and any two objects across different groups have a maximum distance value. It is useful to denote the distance between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ as $d(\mathbf{x}_i, \mathbf{x}_j)$. A valid distance measure should be symmetric and obtain its minimum value (usually zero) in identical vectors. The distance measure is called a metric distance measure if it also satisfies the following properties:

- Triangle inequality $d(\mathbf{x}_i, \mathbf{x}_k) \leq d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_k), \forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in D$.

- $d(\mathbf{x}_i, \mathbf{x}_j) = 0 \iff \mathbf{x}_i = \mathbf{x}_j, \forall \mathbf{x}_i, \mathbf{x}_j \in D$

Real-world datasets usually contain noise and outliers due to the volume, velocity, and variety of the current data universe. Most algorithms focus on spherical or Gaussian clusters, making them sensitive to outliers. They try to optimize the log-likelihood of a Gaussian that is equivalent to the Euclidean or Mahalanobis distance. For example, in the K-means methods, each cluster center is represented by the sample mean. The sample mean is extremely sensitive toward outlying values. Since the mean suffers from a lack of robustness, only a few erroneous or missing data values may hide the inherent cluster structure of the dataset. However, as pointed out by Jin et al. (JIN et al., 2006), from a knowledge discovery point of view, outliers are sometimes more interesting than the rest of the data. They contain underlying information anomalies in contrast to the frequent cases represented by the rest of the data, i.e., examples of such applications are fraud and intrusion detection. Hence the importance of studying robust approaches to outliers.

The simple framework of a K-means algorithm makes it very flexible to modify and build more efficient algorithms on top of it. Some of the variations proposed to the K-means are based on:

- Choosing different prototypes for the clusters (K-medoids, K-medians, K-modes).
- Choosing better initial centroid estimates (Intelligent K-means, Genetic K-means).
- Applying some kind of feature transformation technique (Weighted K-means, Kernel K-means).

K-medoids (FRIEDMAN; HASTIE; TIBSHIRANI, 2001) is a clustering algorithm similar to K-means. The goal is to find a clustering solution that minimizes a predefined objective function. However, unlike K-mean, the method chooses the actual data points as the prototypes, making it more robust to noise and outliers. The technique aims to minimize a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. Despite the robustness of such an approach, it is also computationally more expensive, which is not suitable for large datasets.

K-medians clustering is a variation of K-means clustering that calculates the median for each cluster as opposed to calculating the mean of the cluster. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to the squared 2-norm distance metric. K-medians is more robust to outliers compared with the K-means. The goal is to determine those subsets of median points that minimize the cost of assignment of the objects to the nearest medians. More information about the K-means variations can be found in Refs. (REDDY; VINZAMURI, 2013; CHARU; CHANDAN, 2013).

Since the FCM algorithm is very similar to the K-means algorithm, many authors have proposed several robust variants. For example, Ahmed (AHMED et al., 2002) has proposed a modified FCM algorithm based on a modified objective function based on neighbor's information. This algorithm is more robust than FCM and gives good results, for noisy MRI medical

images. Krishnapuram et al. (KRISHNAPURAM et al., 2001) presented a robust fuzzy-medoids for fuzzy clustering of relational data. The objective function is based on selecting representative objects (medoids) from the dataset so that the total fuzzy dissimilarity within each cluster is minimized. Ref. (D'URSO; GIOVANNI; MASSARI, 2015) introduced a fuzzy C-medoids clustering model for interval-valued data for avoiding the disruptive effects of possible outliers.

Sometimes distance metrics are not always good enough to capture correlations among variables. There is a high probability of the existence of similar data patterns among a set of features, even if they are far apart from each other as measured by the distance metrics (KULKARNI; TOKEKAR; KULKARNI, 2015). This can happen, for example, when all variables are used for the clustering process. Irrelevant dimensions can confuse the clustering algorithms by hiding clusters formed in specific dimensions. In high-dimensional problems, it is common for all of the objects in a dataset to be nearly equidistant from each other, completely masking the clusters (PARSONS; HAQUE; LIU, 2004).

2.4 ADAPTIVE PARTITIONING CLUSTERING

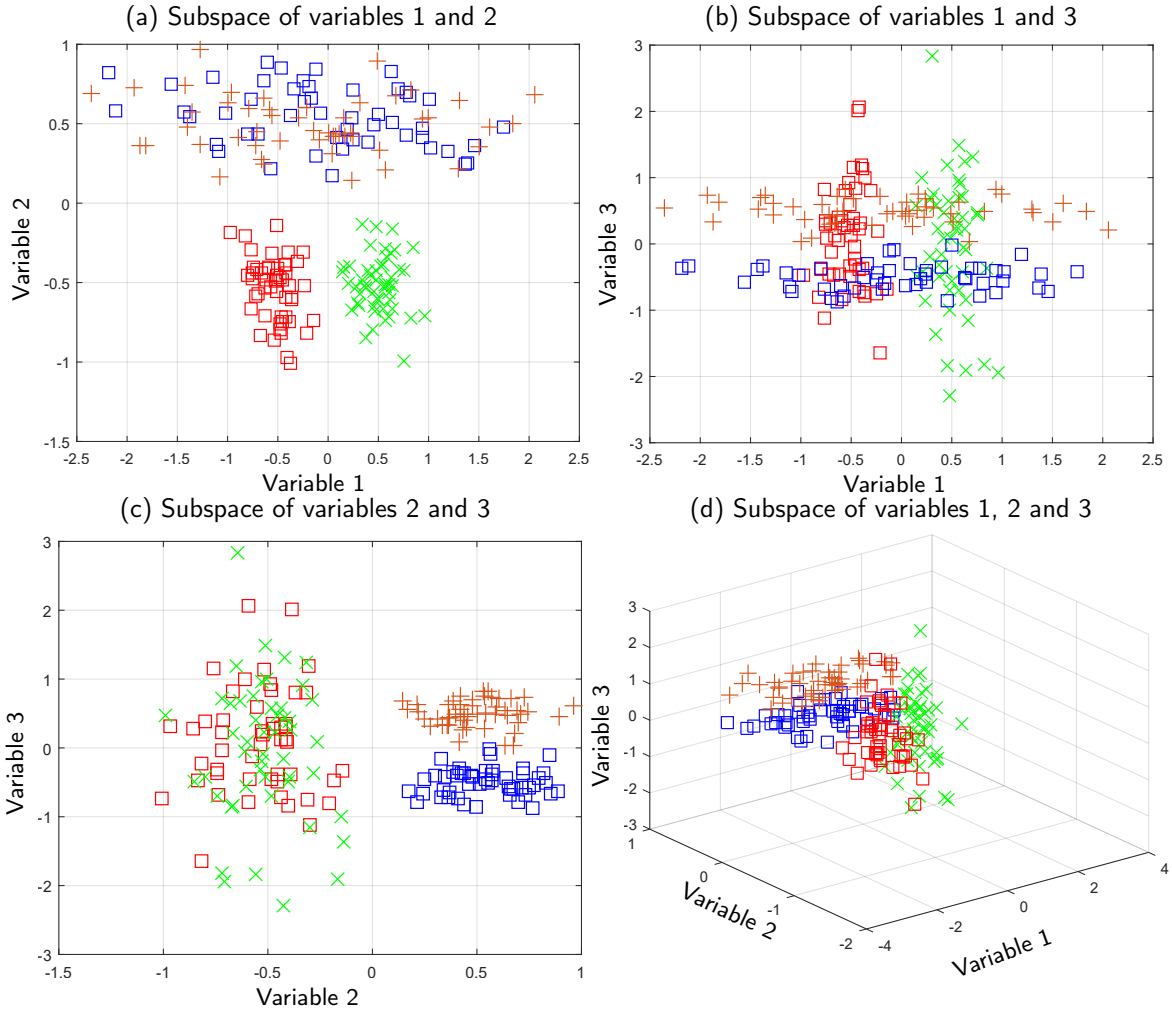
Traditional clustering methods assume that the variables are uncorrelated and equally relevant. The Euclidean distance as a dissimilarity measure restricts conventional algorithms to datasets with hyper-spherical clusters and linearly separable characteristics. However, in real problems, mainly in high-dimensional 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. Please see Figure 4.

Addressing such, several techniques for clustering high dimensional data have included both feature transformation and feature selection techniques. Feature transformation techniques summarize the dataset in fewer dimensions by creating combinations of the original attributes, which help uncover latent structure. However, since they preserve the relative distances between objects, they are less effective when many irrelevant attributes hide the clusters. Another observation is that these new features are combinations of the originals, making the new features challenging to interpret.

Feature selection approaches select only the most relevant dimensions to reveal clusters of objects similar to only a subset of their attributes. Despite their usefulness in many datasets, feature selection algorithms have difficulty finding groups in different subspaces. It is this type of data that motivated the evolution of Subspace clustering (SC) algorithms. The goal of SC is to locate clusters in different subspaces or a specific subspace of the original data space. SC is tolerant of missing values in input data. An object belongs to a particular subspace if the attribute values in this subspace are not missing, irrespective of the values of the rest of the attributes. This allows records with missing values to be used for clustering with more accurate results than replacing missing values with values taken from a distribution.

The two main classes of SC algorithms are Hard Subspace Clustering (HSC) and SSC.

Figure 4 – Visualization of different variables subspace. (a) Subspace of variables 1 and 2. (b) Subspace of variables 1 and 3. (c) Subspace of variables 2 and 3. (d) Subspace of variables 1, 2 and 3.



Source: Author (2022)

While the goal of HSC is to identify exact subspaces, SSC algorithms perform clustering in high-dimensional spaces by assigning a weight to each dimension to measure the contribution of individual dimensions to the formation of a particular cluster. A detailed review of hard subspace clustering algorithms can be found in Ref. (PARSONS; HAQUE; LIU, 2004). In SSC methods, each dimension contributes to every cluster with different contributions. The subspaces of different clusters can be identified by the weight values after clustering. In fact, SSC can be considered an extension of the conventional Feature Weighting Clustering (FWC) (HUANG et al., 2005; CHEUNG; ZENG, 2007; TSAI; CHIU, 2008; BOUGUILA, 2009), which employs a common weight vector for the whole dataset in the clustering procedure $\mathbf{V} = (v_1, \ldots, v_P)$. However, SSC assigns a weight vector to each cluster $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_C)$, with $\mathbf{v}_k = (v_{k1}, \ldots, v_{kP})$ that represents the P-dimensional vectors of relevance weights on the k-th cluster. From this perspective, SSC may thus be referred to as multiple features weighting clustering, which can

be divided into two main categories (DENG et al., 2010):

- Fuzzy Weighting Subspace Clustering (FWSC).
- Entropy Weighting Subspace Clustering (EWSC).

In the first type of algorithms, the weights are assigned to the features of different clusters with a fuzzy index, while the weights in the second type of algorithms are controllable by the entropy to a certain extent. Compared with traditional non-subspace clustering techniques, SSC has demonstrated promising performance in data clustering, especially for high-dimensional data.

Several SSC methods have been proposed (ZHU et al., 2014; WANG et al., 2016; CHEN et al., 2016). Wang et al. (WANG; WANG; WANG, 2004) introduced a technique applying a weighted Euclidean distance in the FCM formulation to improve the clustering performance. Later, Deng et al. (DENG et al., 2011) 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. (HANMANDLU et al., 2013) 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 (RODRÍGUEZ; CARVALHO, 2017) 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, hereafter, we adopt the notation Entropy Fuzzy Clustering Method with Local Product restriction and adaptive Euclidean distance (EFCM-LP2) for the clustering algorithm of Ref. (RODRÍGUEZ; CARVALHO, 2017).

2.5 FUZZY CLUSTERING WITH ENTROPY REGULARIZATION

Regularization is an old technique to solve ill-posed problems of functional equations (TIKHONOV, 1963) and has been applied to many real problems. Generally, regularization implies modifying a given problem that is singular in some sense into a regular problem. The singular problem is challenging to solve, while the other is easier to handle. The latter problem can be seen as a regularization of the original when the solution of the regularized problem approximates the original solution.

In hard clustering, the objects belong to one and only one group. However, in many problems, an item can belong to more than one cluster. For example, a university student can be both an enrolled student and an employee of the university. Relaxing this rigidity has constituted a domain of research in the framework of cluster analysis. Many authors have proposed a fuzzy setting as the appropriate approach to cope with this problem (BEZDEK, 2013). The notion of membership degree has replaced the crisp assignment of the classical techniques, thus

reflecting the imprecision in defining appropriate groups, especially when classifying complex objects.

The FCM method is the most known representative in this category. It imposes a fuzziness coefficient m to control the extent of membership sharing between fuzzy clusters. However, according to Li and Mukaidono (LI; MUKAIDONO, 1995), this parameter can be seen as an artificial device, without strong theoretical justification (LI; MUKAIDONO, 1995; COPPI; D'URSO, 2006). Addressing such, a new approach to fuzzy clustering was introduced by proposing the so-called Maximum Entropy Inference Method (LI; MUKAIDONO, 1995), formalized as follows:

$$maximize \left\{ -\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \right\} \tag{2.1}$$

$$\text{s.t.} \sum_{k=1}^{C} u_{ik} = 1 \quad ; \quad \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d_{ik} = J \tag{2.2}$$

where d_{ik} is a dissimilarity function and J is a given constraint loss function. According to its definition, the total fuzzy cluster entropy is a concave function, and therefore its inverse is a convex function. Therefore, maximizing the entropy is equivalent to minimizing its inverse function. Hence, according to the idea of the maximum entropy principle, combining the known constraint conditions with the maximum entropy term, an alternative expression of optimization objective function of maximum entropy clustering can be obtained by introducing Lagrange multipliers. The expressions for updating the prototypes and membership degree are given in Equation (2.3),

$$u_{ik} = \frac{\exp\frac{-d_{ik}}{2\sigma^2}}{\sum_{h=1}^{C}\exp\frac{-d_{ih}}{2\sigma^2}} \quad ; \quad g_{kj} = \frac{\sum_{i=1}^{N} u_{ik}x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{2.3}$$

where σ is the Lagrange multiplier for the second constraint. Please note that σ^2 resembles the concept of temperature in statistical physics. When applied to fuzzy clustering, such maximum entropy principle provides a new perspective to facing the problem of fuzzifying the clusterization of the units while ensuring the maximum compactness of the obtained clusters.

Based on the above idea, several maximum entropy clustering methods have been proposed. For example, Miyamoto and Mukaidono (SADAAKI; MASAO, 1997) offered a trade-off between fuzziness and compactness by introducing a unique objective function reformulating the maximum entropy method in terms of regularization of the FCM function. Since real datasets can have noise and outliers, Ref. (MIZUTANI; MIYAMOTO, 2005) introduced a kernel-based possibilistic C-means clustering with entropy regularization to get a robust solution in a noisy environment. Also, Rodríguez and de Carvalho (RODRÍGUEZ; CARVALHO, 2018a) proposed a soft subspace clustering algorithm based on adaptive City-Block distances and entropy regularization. Later, Tao et al. (TAO et al., 2019) presented a density-sensitive kernel maximum entropy clustering algorithm. It performs a mapping to high-dimensional feature space by adopting an appropriate kernel function to group non-Gaussian-distributed data.

In general, these methods 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. According to the entropy concentration principle, most possible states tend to finally concentrate near the one with the maximum entropy. Hence maximum entropy clustering algorithm is claimed to produce more accurate clustering results than other traditional clustering techniques (TAO et al., 2019). Since such models share similarities with Ref. (LI; MUKAIDONO, 1995), they offer a distinct physical meaning and well-defined mathematical characteristics, making them easy to understand (LI; MUKAIDONO, 1995; SADAAKI; MASAO, 1997; COPPI; D'URSO, 2006). They can produce desirable results that accurately reveal the known data's underlying cluster structure and achieve the least estimation for the incomplete data and the noise data to minimize the deviation. Furthermore, maximum entropy clustering methods have gained much attention due to their low sensitivity to initialization and high clustering performance on large-scale data (TAO et al., 2019).

2.6 CLUSTER VALIDITY

Cluster analysis is used for grouping a dataset into clusters of similar individuals. The partitions generated by a clustering approach define for all data elements to which group they belong. But clustering is an unsupervised process with no predefined classes and no examples that can show that the clusters found by the clustering algorithms are valid. Regardless of whether the structure exists, any clustering algorithm can always provide a division given a dataset. Besides, different methods typically lead to different clusters; even for the same algorithm, identifying the parameters or the presentation order of the input patterns may impact the final results. Therefore, effective evaluation standards and criteria are essential to provide a degree of confidence for the clustering results. Such evaluations should be objective and have no preferences for any algorithm. They are also helpful for answering questions like how many clusters are hidden in the data or why we choose some algorithm instead of another (XU; WUNSCH, 2005).

Cluster validity indices estimate the quality of partitions produced by clustering algorithms (XU; WUNSCH, 2008). Generally, there are two types of validation techniques, which are based on external criteria and internal criteria (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2002). The first one is based on external criteria. This implies that we evaluate the results of a clustering algorithm based on a previously specified structure imposed on a dataset and reflects our intuition about the clustering structure of the dataset. Internal criteria are not dependent on external information (prior knowledge). On the contrary, they examine the clustering structure directly from the original data. The indices used in this work to validate the clustering algorithms are described below.

2.6.1 Adjusted Rand Index

The Adjusted Rand Index (ARI) (HUBERT; ARABIE, 1985) is an external validation metric that measures the correspondence between two partitions of the same data. It takes its values on the interval $[-1, 1]$, in which 1 indicates perfect agreement between partitions. In contrast, values near 0 or negatives correspond to cluster agreement found by chance. Let $Q = \{Q_1, \ldots, Q_m\}$ be the a priori partition into m classes and $\mathcal{P} = \{P_1, \ldots, P_C\}$ be the partition into C clusters provided by the clustering algorithm. The ARI is defined as:

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

where $n_{i,j}$ is the number of agreements between class Q_i and the cluster P_j, n_i (or n_j) is the number of data points in class Q_i (or cluster P_j) and N is the size of the whole dataset.

2.6.2 Hullermeier index

The internal validation metric Hullermeier (HUL) (Equation (2.4)) is a fuzzy extension of the Rand index comparing two fuzzy partitions (HULLERMEIER et al., 2011). 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. (HULLERMEIER et al., 2011).

$$HUL = 1 - \left[\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}|\|\mathbf{V}_i - \mathbf{V}_j\| - \|\mathbf{U}_i - \mathbf{U}_j\||}{\binom{N}{2}}\right] \tag{2.4}$$

2.6.3 Partition Coefficient

The internal validation metric Partition Coefficient (PC) (BEZDEK, 1973) indicates the average relative amount of shared membership between pairs of fuzzy subsets in $\mathbf{U}$, when combining into a single number, the average contents of pairs of fuzzy algebraic products. Please see Equation (2.5). The index values range in $[\frac{1}{C}, 1]$, where C is the number of clusters. The closer the value of PC to $\frac{1}{C}$, the fuzzier the clustering is. The lower value is obtained when $u_{ik} = \frac{1}{C}, \forall k, i$.

$$V_{PC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}u_{ik}^2 \tag{2.5}$$

2.6.4 Partition Entropy

The internal validation metric Partition Entropy (PE) (BEZDEK, 1975) measures the amount of fuzziness in a given fuzzy partition $\mathbf{U}$ and is defined as in Equation (2.6). Its values range in $[0, \log(C)]$, such that the closer the value to 0, the crisper the clustering is. The index value close to the upper bound indicates the absence of any clustering structure or the inability of the algorithm to extract it.

$$V_{PE} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} (u_{ik}) \log(u_{ik}) \tag{2.6}$$

3 REAL-VALUED DATA CLUSTERING WITH ENTROPY REGULARIZATION

3.1 INTRODUCTION

This chapter presents a literature review of the methods most closely related to this work. New soft subspace clustering algorithms are proposed based on Euclidean, City-block, and Mahalanobis distances and entropy regularization. These methods are iterative three-step algorithms that provide a fuzzy partition, a representative for each fuzzy 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 its application to noisy image texture segmentation, demonstrate the usefulness of these adaptive clustering methods.

3.2 LITERATURE FUZZY CLUSTERING METHODS BASED ON ENTROPY REGULARIZATION

The literature holds several maximum entropy clustering algorithms to search for global regularity and obtain the smoothest reconstructions from the available data. Ref. (SADAAKI; MASAO, 1997) proposes a variant of the FCM algorithm considering entropy regularization, hereafter named EFCM. 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. The minimization of the following objective function is implicated:

$$J_{EFCM} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})d(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \tag{3.1}$$

$$\text{s.t.} \quad \sum_{k=1}^{C} u_{ik} = 1 \quad \text{and} \quad u_{ik} \in [0,1] \tag{3.2}$$

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 comparison, two variants of EFCM were considered in the experimental section, named EFCM-2 and EFCM-1, with $d(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{P}(x_{ij} - g_{kj})^2$ and $d(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{P}|x_{ij} - g_{kj}|$, respectively. The first term in the Equation (3.1) 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.

Case 1: If the dissimilarity function between objects and prototypes is based on the Euclidean distance, then, the prototype $\mathbf{g}_k = (g_{kj}, \ldots, g_{kP})$ of the cluster k that minimizes the clustering criterion is calculated as follows:

$$g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{3.3}$$

Case 2: If the dissimilarity function is based on the City-Block distance, then the minimization of the clustering criterion with respect to g_{kj} leads to the minimization of $\sum_{i=1}^{N} |y_i - a z_i|$, where $y_i = u_{ik}\, x_{ij}$, $z_i = u_{ik}$ and $a = g_{kj}$. Since there is no algebraic solution for this problem, but an algorithmic solution (JAJUGA, 1991) is known, we used Algorithm 1 to solve it.

Algorithm 1 Prototype computation

1: Determine $b_i = \frac{y_i}{z_i} (i = 1, \ldots, N)$;
2: Rearrange the z_i's according to ascending order of b_i's and get $\tilde{z}_1, \ldots, \tilde{z}_N$;
3: Minimize $\sum_{l=1}^{r} |\tilde{z}_l| - \sum_{l=r+1}^{N} |\tilde{z}_l|$ with respect to r;
4: If the minimum is negative, take $a = b_r$. If the minimum is positive, take $a = b_{r+1}$. If the minimum is equal to zero, take $b_r \leq a \leq b_{r+1}$;

The membership degrees u_{ik} are computed according to Equation (3.4). Table 2 specifies the assignment rules to obtain the fuzzy partition according to the different distance functions.

$$u_{ik} = \frac{\exp\left\{-\frac{d(\mathbf{x}_i, \mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C} \exp\left\{-\frac{d(\mathbf{x}_i, \mathbf{g}_w)}{T_u}\right\}} \tag{3.4}$$

Table 2 – Rules for computing the fuzzy partition according to the distance functions.

Distance function	Rules for u_{ik}
$\sum_{j=1}^{P} (x_{ij} - g_{kj})^2$	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P} (x_{ij} - g_{kj})^2}{T_u}\right\}}{\sum_{w=1}^{C} \exp\left\{-\frac{\sum_{j=1}^{P} (x_{ij} - g_{wj})^2}{T_u}\right\}}$
$\sum_{j=1}^{P} \lvert x_{ij} - g_{kj} \rvert$	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P} \lvert x_{ij} - g_{kj} \rvert}{T_u}\right\}}{\sum_{w=1}^{C} \exp\left\{-\frac{\sum_{j=1}^{P} \lvert x_{ij} - g_{wj} \rvert}{T_u}\right\}}$

Source: Author (2022)

Algorithm 2 summarizes the steps of the algorithm EFCM. Note that the minimization of the objective function is performed iteratively in two steps (representation and assignment).

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 (DENG et al., 2010). Addressing such, Ref. (HANMANDLU et al., 2013) proposed an Entropy Fuzzy Clustering Method with Local Sum restriction and adaptive Euclidean distance (EFCM-LS2) for image segmentation that minimizes the following objective function:

Algorithm 2 Algorithm EFCM

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

Output: The vector of prototypes $\mathbf{G}$ and the matrix of membership degrees $\mathbf{U}$.

1: **Initialization**:

Set $t = 0$;

Randomly select C distinct prototypes $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$ to obtain the vector of prototypes $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, ..., \mathbf{g}_C^{(t)})$;

Compute the membership degrees $\mathbf{U}^{(t)} = (u_{ik}^{(t)})_{\substack{1 \leq i \leq N \\ 1 \leq k \leq C}}$ by Equation (3.4);

Compute the initial value of J_{EFCM} according to Equation (3.1);

2: **repeat**

Set $t = t + 1$; $J_{OLD} = J_{EFCM}$;

3: **Step 1: representation**:

Compute g_{kj} according to Equation (3.3) if the dissimilarity function is based on the Euclidean distance, otherwise apply Algorithm 1;

4: **Step 3: assignment**:

Compute the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq C}}$ according to Equation (3.4);

5: Compute J_{EFCM} according to Equation (3.1) and set $J_{NEW} = J_{EFCM}$;

6: **until** $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$

$$
\begin{aligned}
J_{EFCM-LS2} = & \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \sum_{j=1}^{P} (v_{kj})(x_{ij} - g_{kj})^2 \\
& + T_u \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \ln(u_{ik}) + T_v \sum_{k=1}^{C} \sum_{j=1}^{P} (v_{kj}) \ln(v_{kj})
\end{aligned}
\tag{3.5}
$$

where $u_{ik} \in [0, 1]$, $\sum_{k=1}^{C} u_{ik} = 1$ and $v_{kj} \in [0, 1]$, $\sum_{j=1}^{P} v_{kj} = 1$. The weighting parameters T_u and T_v control the degree of fuzziness and the relevance weights of the variables, where $T_u > 0$ and $T_v > 0$. When increasing T_u, the fuzziness of the clusters increases. Additionally, when increasing T_v, the relevance of the variables tends to be similar.

The method starts with an initial partition and computes the prototypes, the relevance weights of the variables, and the cluster partition iteratively in three steps until a stopping criterion is satisfied. The minimum of the prototypes, the relevance of the variables, and fuzzy partition are obtained when Equations (3.6) to (3.8). The method is executed according to Algorithm 3.

$$
g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{3.6}
$$

$$
v_{kj} = \frac{exp\{-\frac{\sum_{i=1}^{N}(u_{ik})(x_{ij} - g_{kj})^2}{T_v}\}}{\sum_{w=1}^{P} exp\{-\frac{\sum_{i=1}^{N}(u_{ik})(x_{iw} - g_{kw})^2}{T_v}\}} \tag{3.7}
$$

$$u_{ik} = \frac{exp\{-\frac{\sum_{j=1}^{P}(v_{kj})(x_{ij}-g_{kj})^2}{T_u}\}}{\sum_{w=1}^{C} exp\{-\frac{\sum_{j=1}^{P}(v_{wj})(x_{iw}-g_{kw})^2}{T_u}\}} \tag{3.8}$$

Algorithm 3 Algorithm EFCM-LS2

Input: The dataset D, the number C of clusters, the parameter $T_u > 0$, 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 $\mathbf{U}$ and the relevance weight vectors $\mathbf{V}$.

1: **Initialization**:

Set $t = 0$;

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

Randomly select C distinct prototypes $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$ to obtain the vector of prototypes $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, ..., \mathbf{g}_C^{(t)})$;

Initialize the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1\leq k\leq C\\ 1\leq j\leq P}}$ with $v_{kj} = 1/P$, $\forall k, j$;

Compute the initial value of $J_{EFCM-LS2}$ by Equation (3.5);

2: **repeat**

Set $t = t + 1$; $J_{OLD} = J_{EFCM-LS2}$;

3: **Step 1: representation**:

For $k = 1, \ldots, C; j = 1, \ldots, P$, compute the component g_{kj} of the prototype $\mathbf{g}_k = (g_{k1}, ..., g_{kP})$ according to Equation (3.6);

4: **Step 2: weighting**:

Compute the component v_{kj} of the vector of relevance weights $\mathbf{v}_k = (v_{k1}, ..., v_{kP})$ according to Equation (3.7);

5: **Step 3: assignment**:

Compute the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1\leq i\leq N\\ 1\leq j\leq C}}$ according to Equation (3.8);

6: Compute $J_{EFCM-LS2}$ by Equation (3.5) and set $J_{NEW} = J_{EFCM-LS2}$;

7: **until** $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$

Later, Rodríguez and de Carvalho (RODRÍGUEZ; CARVALHO, 2017) proposed a FWSC with fuzzy entropy that minimizes the following objective function:

$$J_{EFCM-LP2} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(v_{kj})(x_{ij} - g_{kj})^2 + T_u\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \tag{3.9}$$

subject to $u_{ik} \in [0, 1]$, $v_{kj} > 0$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\prod_{j=1}^{P} v_{kj} = 1$. 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. T_u is a weighting parameter that specifies the fuzziness degree, such that $T_u > 0$. The advantage of this method is that it requires less parameter configuration. The method EFCM-LP2 sets an initial fuzzy partition and alternates three steps until reach a suitable stopping criterion. The minimum of the prototypes, relevance of the variables and fuzzy partition are obtained from Equations (3.10) to (3.12). The method is executed according to Algorithm 4.

$$g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \quad (3.10) \qquad v_{kj} = \frac{\left\{\prod_{w=1}^{P} \sum_{i=1}^{N} (u_{ik})(x_{iw} - g_{kw})^2\right\}^{\frac{1}{P}}}{\sum_{i=1}^{N} (u_{ik})(x_{ij} - g_{kj})^2} \quad (3.11)$$

$$u_{ik} = \frac{\exp\{-\frac{\sum_{j=1}^{P}(v_{kj})(x_{ij}-g_{kj})^2}{T_u}\}}{\sum_{w=1}^{C} \exp\{-\frac{\sum_{j=1}^{P}(v_{wj})(x_{ij}-g_{wj})^2}{T_u}\}} \quad (3.12)$$

Algorithm 4 Algorithm EFCM-LP2

Input: The dataset D, the number C of clusters, the parameter $T_u > 0$, 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 $\mathbf{U}$ and the relevance weight vectors $\mathbf{V}$.

1: **Initialization**:

Set $t = 0$;

Randomly select C distinct prototypes $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$ to obtain the vector of prototypes $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, ..., \mathbf{g}_C^{(t)})$;

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

Set the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1 \le k \le C \\ 1 \le j \le P}}$ with $v_{kj} = 1$, $\forall k, j$;

Compute the initial value of $J_{EFCM-LP2}$ by Equation (3.9);

2: **repeat**

Set $t = t + 1$; $J_{OLD} = J_{EFCM-LP2}$;

3: **Step 1: representation**:

For $k = 1, \ldots, C; j = 1, \ldots, P$, compute the component g_{kj} of the prototype $\mathbf{g}_k = (g_{k1}, ..., g_{kP})$ according to Equation (3.10);

4: **Step 2: weighting**:

Compute the component v_{kj} of the vector of relevance weights $\mathbf{v}_k = (v_{k1}, ..., v_{kP})$ according to Equation (3.11);

5: **Step 3: assignment**:

Compute the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1 \le i \le N \\ 1 \le j \le C}}$ according to Equation (3.12);

6: Compute $J_{EFCM-LP2}$ by Equation (3.9) and set $J_{NEW} = J_{EFCM-LP2}$;

7: **until** $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$

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. Lastly, the above methods don't work well in noisy environments since the dissimilarity function is based on the Euclidean distance. So, robust measures of dissimilarities are needed.

3.3 PROPOSED FUZZY REAL-VALUED DATA CLUSTERING WITH AUTOMATIC VARIABLE SELECTION AND ENTROPY REGULARIZATION

This section presents new fuzzy clustering algorithms based on feature-weight learning. The proposals measure the heterogeneity of the fuzzy partition as the sum of the heterogeneity in each fuzzy cluster, where the distance-based term defines the shape and size of the groups and encourages agglomeration. Additionally, it employs an entropy term that serves as a regulating factor during the minimization process. The algorithms can be divided into three main categories: fuzzy weighting subspace clustering (FWSC), entropy weighting subspace clustering (EWSC) and feature weighting clustering (FWC).

Initially, a FWSC algorithms were introduced. The first proposal takes into account the correlation of the variables in the clustering process. In this case, the method is defined by a local covariance matrix introduced by Gustafson and Kessel (GUSTAFSON; KESSEL, 1979), which changes in each iteration of the algorithm and is different from one cluster to another. In this case, the algorithm is named Entropy Fuzzy Clustering Method with Local Mahalanobis distance (EFCM-Mk), and minimizes Equation (3.13) such that $u_{ik} \in [0,1]$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\det(\mathbf{M}_k) = 1$.

$$\begin{aligned} J_{EFCM-Mk} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{\mathbf{M}_k}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}_k(\mathbf{x}_i - \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{3.13}$$

The adaptive distance defined by a global covariance matrix that changes in each algorithm iteration and is the same for all clusters was also considered. In this case, the proposed Entropy Fuzzy Clustering Method with Global Mahalanobis distance (EFCM-M) minimizes the following objective function:

$$\begin{aligned} J_{EFCM-M} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{3.14}$$

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. T_u is a weighting parameter specifying the fuzziness degree. The higher it is, the fuzzier the cluster will be in the end.

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 $\mathbf{v} = (v_1, ..., v_P)$, in which $v_j > 0$ and $\prod_{j=1}^{P} v_j = 1$. This FWC approach is named Entropy Fuzzy Clustering Method with Global Product restriction (EFCM-GP) and its objective function is defined as in Equation (3.15), such that $u_{ik} \in [0,1]$, $v_j > 0$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\prod_{j=1}^{P} v_j = 1$. As before, T_u specifies the fuzziness degree.

$$\begin{aligned} J_{EFCM-GP} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \ln(u_{ik}) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \sum_{j=1}^{P}(v_j) d(x_{ij}, g_{kj}) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \ln(u_{ik}) \end{aligned} \tag{3.15}$$

The method EFCM-GP is named Entropy Fuzzy Clustering Method with Global Product restriction and adaptive Euclidean distance (EFCM-GP2) when $d_{\mathbf{v}}$ is based on the global adaptive Euclidean distance. In this case, the objective function becomes:

$$J_{EFCM-GP2} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \sum_{j=1}^{P}(v_j)(x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \ln(u_{ik}) \tag{3.16}$$

Furthermore, EFCM-GP is defined as Entropy Fuzzy Clustering Method with Global Product restriction and adaptive City-Block distance (EFCM-GP1) when $d_{\mathbf{v}}$ is the global adaptive City-Block distance. In this case, the objective function becomes:

$$J_{EFCM-GP1} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \sum_{j=1}^{P}(v_j)|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \ln(u_{ik}) \tag{3.17}$$

Unlike the previous cases, the weights of the variables can also be controlled by entropy. In this regard, it is proposed an FWC 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 $\sum_{j=1}^{P} v_j = 1$). This variant is named Entropy Fuzzy Clustering Method with Global Sum restriction (EFCM-GS), and the corresponding algorithm involves the minimization of Equation (3.18), with $\mathbf{v} = (v_1, \ldots, v_P)$ and subject to: $u_{ik} \in [0,1]$, $v_j \in [0,1]$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\sum_{j=1}^{P} v_j = 1$. 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.

$$\begin{aligned} J_{EFCM-GS} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + T_v \sum_{j=1}^{P}(v_j)\ln(v_j) \quad (3.18) \\ &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(v_j) d(x_{ij}, g_{kj}) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + T_v \sum_{j=1}^{P}(v_j)\ln(v_j) \end{aligned}$$

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

$$J_{EFCM-GS2} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(v_j)(x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + T_v \sum_{j=1}^{P}(v_j)\ln(v_j) \quad (3.19)$$

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

$$J_{EFCM-GS1} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(v_j)|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + T_v \sum_{j=1}^{P}(v_j)\ln(v_j) \quad (3.20)$$

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 $\mathbf{v} = (v_1, ..., v_P)$, where v_j is the relevance weight of the j-th variable estimated globally.

Additionally, a robust variable-wise dissimilarity with relevance weight of the variables locally estimated is also considered. In this case, the product of the weights is equal to one (HUANG et al., 2005), and the City-Block distance compares objects and prototypes. The dissimilarity function is parameterized by the vector of relevance weights $\mathbf{v}_k = (v_{k1}, ..., v_{kP})$, in which $v_{kj} > 0$ and $\prod_{j=1}^{P} v_{kj} = 1$, and it is associated with the k-th fuzzy cluster $(k = 1, ..., C)$. This FWSC approach is named Entropy Fuzzy Clustering Method with Local Product restriction and adaptive City-Block distance (EFCM-LP1) and defines the objective function Equation (3.21), such that $u_{ik} \in [0, 1]$, $v_{kj} > 0$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\prod_{j=1}^{P} v_{kj} = 1$. The weighting parameter T_u controls the degree of fuzziness of the clusters. In addition to the matrix of

membership degrees and the vector of prototypes, the method returns the matrix of relevance weights $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_C) = (v_{kj})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$, where v_{kj} is the relevance weight of the j-th variable in the fuzzy cluster k and $\mathbf{v}_k = (v_{k1}, \ldots, v_{kP})$.

$$J_{EFCM-LP1} = \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \sum_{j=1}^{P} (v_{kj}) |x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \ln(u_{ik}) \tag{3.21}$$

The sum of the weights equal to one (HUANG et al., 2005) is also considered. The dissimilarity function is parameterized by the vector of relevance weights $\mathbf{v}_k = (v_{k1}, ..., v_{kP})$, in which $v_{kj} \geq 0$ and $\sum_{j=1}^{P} v_{kj} = 1$, and it is associated with the k-th fuzzy cluster $(k = 1, ..., C)$. This EWSC approach is named Entropy Fuzzy Clustering Method with Local Sum restriction and adaptive City-Block distance (EFCM-LS1) and defines the objective function Equation (3.22), such that $u_{ik} \in [0, 1]$, $v_{kj} \in [0, 1]$, $\sum_{k=1}^{C} u_{ik} = 1$ and $\sum_{j=1}^{P} v_{kj} = 1$. 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.

$$\begin{aligned} J_{EFCM-LS1} = \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \sum_{j=1}^{P} (v_{kj}) |x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) \ln(u_{ik}) \\ + T_v \sum_{k=1}^{C} \sum_{j=1}^{P} (v_{kj}) \ln(v_{kj}) \end{aligned} \tag{3.22}$$

Note that the proposed fuzzy clustering algorithms with the product constraint for the variables require less parameter setting than the other approaches.

3.3.1 Optimization steps

This section provides the optimization steps of the algorithms aiming to compute the prototypes, the fuzzy partition, 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).

3.3.1.1 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 plays an

essential role in computing the prototypes. This work 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 Equation (3.14)) concerning $\mathbf{g}_k$ and obtain Equation (3.23). Then, by solving Equation (3.23), $\mathbf{g}_k$ is defined as Equation (3.24).

$$\frac{\partial J_{EFCM-M}}{\partial \mathbf{g}_k} = -2 \sum_{i=1}^{N} \mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) = 0 \qquad (3.23) \qquad \mathbf{g}_k = \frac{\sum_{i=1}^{N} u_{ik} \mathbf{x}_i}{\sum_{i=1}^{N} u_{ik}} \qquad (3.24)$$

Similarly, if the dissimilarity function is $(\mathbf{x}_i - \mathbf{g}_k)^T \mathbf{M}_k (\mathbf{x}_i - \mathbf{g}_k)$ (Equation (3.13)), $\mathbf{g}_k$ is computed as in Equation (3.24).

Case 2: If the dissimilarity function is based on the global adaptive Euclidean distance, the partial derivative of $J_{EFCM-GS2}$ (see Equation (3.19)) is taken concerning g_{kj} and Equation (3.25) is obtained. Then, Equation (3.26) is obtained by solving Equation (3.25).

$$\frac{\partial J_{EFCM-GS2}}{\partial g_{kj}} = -2 \sum_{i=1}^{N} (u_{ik})(x_{ij} - g_{kj}) = 0 \quad (3.25) \quad g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \quad (3.26)$$

Following a similar reasoning, the prototype g_{kj} of the k-th cluster that minimizes the clustering criterion $J_{EFCM-GP2}$ (see Equation (3.16)) is computed as in Equation (3.26).

Case 3: If the dissimilarity function is the global adaptive City-Block distance, then the minimization problem of Equations (3.17) and (3.20) to (3.22) with respect to g_{kj} leads to the minimization of $\sum_{i=1}^{N} |y_i - a z_i|$, where $y_i = u_{ik}\, x_{ij}$, $z_i = u_{ik}$ and $a = g_{kj}$. Since there is no algebraic solution for this problem, but an algorithmic solution (JAJUGA, 1991) is known, we used Algorithm 1 to solve it.

3.3.1.2 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 $\mathbf{G}$ and the matrix of membership degrees $\mathbf{U}$ 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* $J_{EFCM-Mk}$ *(Equation (3.13)) under* $\det(\mathbf{M}_k) = 1$ *is updated according to the following expression:*

$$\mathbf{M}_k = [\det(C_k)]^{\frac{1}{P}} C_k^{-1} \text{ with } \quad C_k = \sum_{i=1}^{N} (u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T \qquad (3.27)$$

(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} *(Equation (3.14)) under* $\det(\mathbf{M}) = 1$ *is updated according to Equation* (3.28).

$$\mathbf{M} = [\det(Q)]^{\frac{1}{P}} Q^{-1}, \quad Q = \sum_{k=1}^{C} C_k \text{ and } \quad C_k = \sum_{i=1}^{N} (u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T \tag{3.28}$$

(c) If the adaptive distance function is given by $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{P} (v_j) d(x_{ij}, g_{kj})$, *the components* $v_j (j = 1, ..., P)$ *of the vector of weights* $\boldsymbol{v} = (v_1, \ldots, v_P)$ *minimizing the criterion* $J_{EFCM-GS}$ *(Equation (3.18)) under* $v_j \in [0, 1] \; \forall \; j$, *and* $\sum_{j=1}^{P} v_j = 1$ *are computed as follows:*

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) d(x_{ij}, g_{kj})}{T_v}\}}{\sum_{w=1}^{P} exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) d(x_{iw}, g_{kw})}{T_v}\}} \tag{3.29}$$

When the dissimilarity function is based on the Euclidean and City-Block distances, v_j *is defined as Equations* (3.30) *and* (3.31), *respectively.*

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})(x_{ij} - g_{kj})^2}{T_v}\}}{\sum_{w=1}^{P} exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})(x_{iw} - g_{kw})^2}{T_v}\}} \tag{3.30}$$

$$v_j = \frac{exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) |x_{ij} - g_{kj}|}{T_v}\}}{\sum_{w=1}^{P} exp\{-\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) |x_{iw} - g_{kw}|}{T_v}\}} \tag{3.31}$$

(d) If the adaptive distance function is given by $d_{\mathbf{v}}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{P} (v_j) d(x_{ij}, g_{kj})$, *the components* $v_j (j = 1, ..., P)$ *of the vector of weights* $\boldsymbol{v} = (v_1, \ldots, v_P)$ *minimizing the criterion* $J_{EFCM-GP}$ *(Equation (3.15)) under* $v_j > 0 \; \forall \; j$ *and* $\prod_{j=1}^{P} v_j = 1$ *are computed according to Equation* (3.32).

$$v_j = \frac{\left\{\prod_{w=1}^{P} \left[\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) d(x_{iw}, g_{kw})\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) d(x_{ij}, g_{kj})} \tag{3.32}$$

We define Equation (3.32) *as Equations* (3.33) *and* (3.34) *when the dissimilarity function is based on the Euclidean and City-Block distances, respectively.*

$$v_j = \frac{\left\{\prod_{w=1}^{P} \left[\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})(x_{iw} - g_{kw})^2\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})(x_{ij} - g_{kj})^2} \tag{3.33}$$

$$v_j = \frac{\left\{\prod_{w=1}^{P} \left[\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) |x_{iw} - g_{kw}|\right]\right\}^{\frac{1}{P}}}{\sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik}) |x_{ij} - g_{kj}|} \tag{3.34}$$

(e) If the adaptive distance function is given by $\sum_{j=1}^{P} (v_{kj}) |x_{ij} - g_{kj}|$, *the components* $v_{kj} (k = 1, ...C, j = 1, ..., P)$ *of the vector of weights* $\boldsymbol{v}_k = (v_{k1}, \ldots, v_{kP})$ *minimizing the*

criterion $J_{EFCM-LP1}$ (Eq. (3.21)) under $v_{kj} > 0 \; \forall \; k, j$ and $\prod_{j=1}^{P} v_{kj} = 1 \; \forall \; k$ are computed as follows:

$$v_{kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|\right]\right\}^{\frac{1}{P}}}{\sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}|} \tag{3.35}$$

(f) If the adaptive distance function is given by $\sum_{j=1}^{P}(v_{kj})|x_{ij} - g_{kj}|$, the components $v_{kj}(k = 1, ...C, j = 1, ..., P)$ of the vector of weights $\boldsymbol{v}_k = (v_{k1}, \ldots, v_{kP})$ minimizing the criterion $J_{EFCM-LS1}$ (Eq. (3.22)) under $v_{kj} \in [0, 1] \; \forall \; k, j$ and $\sum_{j=1}^{P} v_{kj} = 1 \; \forall \; k$ are computed as follows:

$$v_{kj} = \frac{exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\}}{\sum_{w=1}^{P} exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{3.36}$$

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

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) + T_u\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + \beta\left[1 - \det(\mathbf{M})\right] \tag{3.37}$$

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 $\mathbf{y}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{M}} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T - \beta\det(\mathbf{M})\mathbf{M}^{-1} = 0 \tag{3.38}$$

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

An extremum value of J_{EFCM-M} is reached when $\mathbf{M} = (\det(Q))^{\frac{1}{P}}Q^{-1}$. This extremum value is $J_{EFCM-M}((\det(Q))^{\frac{1}{P}}Q^{-1}) = trace[Q(\det(Q))^{\frac{1}{P}}Q^{-1}] = P\det(Q))^{\frac{1}{P}}$. 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: $\mathbf{P}\mathbf{P}^T = \mathbf{P}^T\mathbf{P} = \mathbf{I}$, $\Lambda = diag(\varsigma_1, ..., \varsigma_P)$, and $\varsigma_j(j = 1, ..., P)$ are the eigenvalues of Q. Thus $J_{EFCM-M}(\mathbf{I}) = trace[\mathbf{P}\Lambda\mathbf{P}^T] = trace[\Lambda] = \sum_{j=1}^{P}\varsigma_j$. Moreover, $\det(Q) = \det(\mathbf{P}\Lambda\mathbf{P}^T) = \det(\Lambda) = \prod_{j=1}^{P}\varsigma_j$. As it is well known that the arithmetic mean is greater than the geometric mean, i.e., $(1/P)(\varsigma_1 + ... + \varsigma_P) > \{\varsigma_1 \times .. \times \varsigma_P\}^{1/P}$ (the equality holds only if $\varsigma_1 = ... = \varsigma_P$, it follows that $J_{EFCM-M}(\mathbf{I}) > J_{EFCM-M}(\det(Q))^{\frac{1}{P}}Q^{-1})$. Thus, we conclude that this extreme is a minimum.

Note that the matrix C_k is related to the fuzzy covariance matrix in the k-th 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 $\mathbf{M}_k = [\det(C_k)]^{\frac{1}{P}} C_k^{-1}$ with $C_k = \sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T$.

(c) We want to minimize $J_{EFCM-LS1}$ with respect to $v_{kj}, (k = 1, ...C, j = 1, ..., P)$ under $v_{kj} \in [0, 1]\ \forall j$ and $\sum_{j=1}^{P} v_{kj} = 1$. We use the Lagrangian multiplier to solve the unconstrained minimization problem in Equation (3.22).

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \sum_{j=1}^{P}(v_{kj})|x_{ij} - g_{kj}| + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \\ + T_v \sum_{k=1}^{C}\sum_{j=1}^{P}(v_{kj})\ln(v_{kj}) - \sum_{k=1}^{C}\gamma_k\left[\sum_{j=1}^{P} v_{kj} - 1\right] \tag{3.39}$$

Taking the partial derivative of $\mathcal{L}$ in Equation (3.39) with respect to v_{kj} and setting the gradient to zero we have Equation (3.40). Then, from Equation (3.40), is obtained Equation (3.41).

$$\frac{\partial \mathcal{L}}{\partial v_{kj}} = \sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}| + T_v(\ln(v_{kj}) + 1) - \gamma_k = 0 \tag{3.40}$$

$$v_{kj} = \exp\{\frac{\gamma_k}{T_v} - 1\}\exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\} \tag{3.41}$$

Substituting Equation (3.40) in $\sum_{w=1}^{P} v_{kj} = 1$, we have

$$\sum_{w=1}^{P} v_{kw} = \sum_{w=1}^{P}\exp\{\frac{\gamma_k}{T_v} - 1\}\exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\} = 1 \tag{3.42}$$

It follows that

$$\exp\{\frac{\gamma_k}{T_v} - 1\} = \frac{1}{\sum_{w=1}^{P}\exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{3.43}$$

Substituting Equation (3.43) in Equation (3.40), we obtain

$$v_{kj} = \frac{\exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}|}{T_v}\}}{\sum_{w=1}^{P}\exp\{-\frac{\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|}{T_v}\}} \tag{3.44}$$

Also we have that

$$\frac{\partial J_{EFCM-LS1}}{\partial v_{kj}} = \sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}| + T_v(\ln(v_{kj}) + 1); \frac{\partial^2 J_{EFCM-LS1}}{\partial v_{kj}} = \frac{T_v}{v_{kj}} \tag{3.45}$$

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

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

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

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

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

(e) We want to minimize $J_{EFCM-GP}$ with respect to v_j, $(k = 1, ..., C)$, under $v_j > 0 \; \forall \; j$ and $\prod_{j=1}^{P} v_j = 1$. We use the Lagrangian multiplier to solve the unconstrained minimization problem in Equation (3.15).

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{k=1}^{C}\sum_{j=1}^{P}(u_{ik})(v_j)d(x_{ij},g_{kj}) + T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) - \gamma\left[\prod_{j=1}^{P} v_j - 1\right] \quad (3.46)$$

Taking the partial derivative of $\mathcal{L}$ in Equation (3.46) with respect to v_j and setting the gradient to zero, we have Equation (3.47). Then, from Equation (3.47), Equation (3.48) is obtained.

$$\frac{\partial \mathcal{L}}{\partial v_j} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj}) - \frac{\gamma}{v_j} = 0 \quad (3.47) \qquad v_j = \frac{\gamma}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj})} \quad (3.48)$$

Substituting Equation (3.48) in $\prod_{j=1}^{P} v_j = 1$, we have Equation (3.49). Then after some algebra, Equation (3.50) is obtained.

$$\prod_{h=1}^{P} v_h = \prod_{h=1}^{P}\frac{\gamma}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})} = 1 \quad (3.49) \qquad \gamma = \{\prod_{h=1}^{P}\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})\}^{\frac{1}{P}} \quad (3.50)$$

Substituting Equation (3.50) in Equation (3.48) we obtain

$$v_j = \frac{\{\prod_{h=1}^{P}\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ih},g_{kh})\}^{\frac{1}{P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d(x_{ij},g_{kj})} \quad (3.51)$$

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

(f) Following reasoning similar as in part (e), we conclude that

$$v_{kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})|x_{iw} - g_{kw}|\right]\right\}^{\frac{1}{P}}}{\sum_{i=1}^{N}(u_{ik})|x_{ij} - g_{kj}|} \tag{3.52}$$

3.3.1.3 Assignment step

This step provides the solution to compute the matrix $\mathbf{U}$ of membership degree. In the assignment step, the prototype $\mathbf{G}$ and the matrix $\mathbf{M}$ 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* $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ *is updated according to Equation* (3.53)*, with* $\mathbf{u}_i = (u_{ik}, \ldots, u_{iC})$*, such that* $u_{ik}(i = 1, \ldots, N; k = 1, \ldots, C)$ *represents the membership degree of object* e_i *in the* k*-th fuzzy cluster, under* $u_{ik} \in [0, 1]$ *and* $\sum_{k=1}^{C} u_{ik} = 1$*.*

$$u_{ik} = \frac{\exp\left\{-\frac{\Delta(\mathbf{x}_i, \mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\Delta(\mathbf{x}_i, \mathbf{g}_w)}{T_u}\right\}} \tag{3.53}$$

The distance function Δ *compares the* i*-th object and the fuzzy cluster prototype* k*. Table 3 specifies the assignment rules to obtain the fuzzy partition according to the different adaptive distance functions.*

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:

$$\mathcal{L} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) + T_u\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) - \sum_{i=1}^{N}\lambda_i\left[\sum_{k=1}^{C}u_{ik} - 1\right] \tag{3.54}$$

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

$$\frac{\partial\mathcal{L}}{\partial u_{ik}} = (\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k) + T_u(\ln(u_{ik}) + 1) - \lambda_i = 0 \tag{3.55}$$

From Equation (3.55) is obtained:

$$u_{ik} = \exp\{\frac{\lambda_i}{T_u} - 1\}\exp\{-\frac{(\mathbf{x}_i - \mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{g}_k)}{T_u}\} \tag{3.56}$$

Subject to $\sum_{w=1}^{C} u_{iw} = 1$

Table 3 – Assignment rules for the fuzzy partition according to the distance functions.

Objective functions	Rules for u_{ik}
Equation (3.14)	$\frac{\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\right\}}$
Equation (3.13)	$\frac{\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}_k(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}_w(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\right\}}$
Equations (3.19) and (3.16)	$\frac{\exp\left\{-\frac{\sum_{j=1}^{P}(v_j)(x_{ij}-g_{kj})^2}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}(v_j)(x_{ij}-g_{wj})^2}{T_u}\right\}}$
Equations (3.20) and (3.17)	$\frac{\exp\left\{-\frac{\sum_{j=1}^{P}(v_j)\lvert x_{ij}-g_{kj}\rvert}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}(v_j)\lvert x_{ij}-g_{wj}\rvert}{T_u}\right\}}$
Equations (3.21) and (3.22)	$\frac{\exp\left\{-\frac{\sum_{j=1}^{P}(v_{kj})\lvert x_{ij}-g_{kj}\rvert}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}(v_{wj})\lvert x_{ij}-g_{wj}\rvert}{T_u}\right\}}$

Source: Author (2022)

$$\sum_{w=1}^{C}\exp\{\frac{\lambda_i}{T_u}-1\}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\}=1 \tag{3.57}$$

From Equation (3.57) we obtain Equation (3.58). Then, substituting Equation (3.58) in Equation (3.56) we have Equation (3.59).

$$\exp\{\frac{\lambda_i}{T_u}-1\}=\frac{1}{\sum_{w=1}^{C}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\}} \tag{3.58}$$

$$u_{ik}=\frac{\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)}{T_u}\}}{\frac{1}{\sum_{w=1}^{C}\exp\{-\frac{(\mathbf{x}_i-\mathbf{g}_w)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_w)}{T_u}\}}} \tag{3.59}$$

Additionally, we know that:

$$\frac{\partial J_{EFCM-M}}{\partial u_{ik}}=(\mathbf{x}_i-\mathbf{g}_k)^T\mathbf{M}(\mathbf{x}_i-\mathbf{g}_k)+T_u(\ln(u_{ik})+1) \quad \text{and} \quad \frac{\partial^2 J_{EFCM-M}}{\partial u_{ik}}=\frac{T_u}{u_{ik}} \tag{3.60}$$

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

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

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. □

Note that the proposed maximum entropy clustering algorithms share some similarities with the Gaussian method proposed by Rui-Ping Li et al. (LI; MUKAIDONO, 1995) regarding the membership degree computing. Please see Section 2.5. Therefore, we can state that the proposals in this work have a distinct physical meaning and well-defined mathematical characteristics (TAO et al., 2019; LI; MUKAIDONO, 1995).

3.3.2 Proposed fuzzy clustering algorithms

Algorithm 5 summarizes the proposed clustering algorithms.

3.3.3 Convergence properties

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 $\mathbf{U}^* = (\mathbf{u}_1^*, \dots, \mathbf{u}_N^*)$ and a vector of prototypes $\mathbf{G}^* = (\mathbf{g}_1^*, \dots, \mathbf{g}_C^*)$ where:

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

– $\mathbb{L}$ is the representation space of the prototypes such that $\mathbf{g}_k \in \mathbb{L}\,(k = 1, \dots, C)$ and $\mathbf{G} \in \mathbb{L}^C = \mathbb{L} \times \dots \times \mathbb{L}$. In this paper $\mathbb{L} = \mathbb{R}^P$.

– $\mathbb{U}$ is the space of the fuzzy partition membership degrees such that $\mathbf{u}_i \in \mathbb{U}\,(i = 1, \dots, N)$. In this paper $\mathbb{U} = \{\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_C) \in [0,1] \times \dots \times [0,1] = [0,1]^C : \sum_{k=1}^{C} u_{ik} = 1$ and $u_{ik} \geq 0\}$ and $\mathbf{U} \in \mathbb{U}^N = \mathbb{U} \times \dots \times \mathbb{U}$.

– $\mathbb{M}$ defines the space of positive definite symmetric matrix with determinant equal to 1, such that $\mathbf{M} \in \mathbb{M}$ and $\mathbf{M}_k \in \mathbb{M}^C = \mathbb{M} \times \dots \times \mathbb{M}$.

Algorithm 5 The proposed algorithms.

Input: The dataset 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**:

Set $t = 0$;

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

2: **repeat**

3: Set $t = t + 1$;

4: **Step 1: representation**:

Computing the component g_{kj} of the prototype $\mathbf{g}_k = (g_{k1}, ..., g_{kP})$ according to Section 3.3.1.1;

5: **Step 2: weighting**:

Computing the weights of the variables as shown in Section 3.3.1.2;

6: **Step 3: assignment**:

Computing the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq C}}$ according to Equation (3.53).

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

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

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

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

Additionally, EFCM-LP1 EFCM-LS1 provides a fuzzy partition $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_N^*)$, 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:

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

– Ξ is the space of vectors of weights such that $\mathbf{v}_k \in \Xi, (k = 1, \ldots, C)$. In this paper, $\Xi = \{\mathbf{v} = (v_1, \ldots, v_P) \in \mathbb{R}^P : v_{kj} > 0 \text{ and } \prod_{j=1}^{P} v_{kj} = 1\}$ or $\Xi = \{\mathbf{v} = (v_1, \ldots, v_P) \in \mathbb{R}^P : v_{kj} \in [0, 1] \text{ and } \sum_{j=1}^{P} v_{kj} = 1\}$, and $\mathbf{V} \in \Xi^C = \Xi \times \cdots \times \Xi$.

Similarly to Ref. (DIDAY; SIMON, 1976), the convergence properties of the proposed algorithms can be studied from the series:

- $v^{(t)}_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathbb{M} \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-M} = J_{EFCM-M}(v^{(t)}_{EFCM-M}) = J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v^{(t)}_{EFCM-Mk}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \mathbb{M}^C \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-Mk} = J_{EFCM-Mk}(v^{(t)}_{EFCM-Mk}) = J_{EFCM-Mk}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}_k, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v^{(t)}_{EFCM-GS}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-GS} = J_{EFCM-GS}(v^{(t)}_{EFCM-GS}) = J_{EFCM-GS}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v^{(t)}_{EFCM-GP}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-GP} = J_{EFCM-GP}(v^{(t)}_{EFCM-GP}) = J_{EFCM-GP}(\mathbf{G}^{(t)}, \mathbf{v}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v^{(t)}_{EFCM-LP1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi^C \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-LP1} = J_{EFCM-LP1}(v^{(t)}_{EFCM-LP1}) = J_{EFCM-LP1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;
- $v^{(t)}_{EFCM-LS1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi^C \times \mathbb{U}^N$ and $u^{(t)}_{EFCM-LS1} = J_{EFCM-LS1}(v^{(t)}_{EFCM-LS1}) = J_{EFCM-LS1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

From the initial terms: $v^{(0)}_{EFCM-M}(\mathbf{G}^{(0)}, \mathbf{M}^{(0)}, \mathbf{U}^{(0)})$, $v^{(0)}_{EFCM-Mk}(\mathbf{G}^{(0)}, \mathbf{M}^{(0)}_k, \mathbf{U}^{(0)})$, $v^{(0)}_{EFCM-GS}(\mathbf{G}^{(0)}, \mathbf{v}^{(0)}, \mathbf{U}^{(0)})$,$v^{(0)}_{EFCM-GP}(\mathbf{G}^{(0)}, \mathbf{v}^{(0)}, \mathbf{U}^{(0)})$, $v^{(0)}_{EFCM-LP1}(\mathbf{G}^{(0)}, \mathbf{V}^{(0)}, \mathbf{U}^{(0)})$ and $v^{(0)}_{EFCM-LS1}(\mathbf{G}^{(0)}, \mathbf{V}^{(0)}, \mathbf{U}^{(0)})$, the algorithms EFCM-M, EFCM-Mk, EFCM-GS, EFCM-GP, EFCM-LP1 and EFCM-LS1 compute the different terms of the series, $v^{(t)}_{EFCM-M}$, $v^{(t)}_{EFCM-Mk}$, $v^{(t)}_{EFCM-GS}$, $v^{(t)}_{EFCM-GP}$, $v^{(t)}_{EFCM-LP1}$, and $v^{(t)}_{EFCM-LS1}$, until the respective convergence (to be demonstrated) when the objective functions J_{EFCM-M}, $J_{EFCM-Mk}$, $J_{EFCM-GS}$, $J_{EFCM-GP}$, $J_{EFCM-LP1}$ 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 $u^{(t)}_{EFCM-LP1} = J_{EFCM-LP1}(v^{(t)}_{EFCM-LP1}) = J_{EFCM-LP1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)})$, $t = 0, 1, \ldots$, *decreases at each iteration and converges;*

vi 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 $J_{EFCM-M}(\mathbf{G}^{(t)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{C}\sum_{i=1}^{N}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{g}^{(t)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$ and $J_{EFCM-M}(\mathbf{G}^{(t+1)}, \mathbf{M}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{C}\sum_{i=1}^{N}(u^{(t)}_{ik})\, d_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u^{(t)}_{ik}) \ln(u^{(t)}_{ik})$, and according to Section 3.3.1.1,

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

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

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

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

$$\mathbf{U}^{(t+1)} = (\mathbf{u}^{(t+1)}_1, \ldots, \mathbf{u}^{(t+1)}_N) = \underbrace{\arg\min}_{\mathbf{U}=(\mathbf{u}_1,\ldots,\mathbf{u}_N)\in\mathbb{U}^N} \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{\mathbf{M}^{(t+1)}}(\mathbf{x}_i, \mathbf{g}^{(t+1)}_k) + T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \ln(u_{ik}) \tag{3.61}$$

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 $u^{(t)}_{EFCM-Mk}, t = 0, 1, \ldots$, $u^{(t)}_{EFCM-GS}, t = 0, 1, \ldots$, $u^{(t)}_{EFCM-GP}, t = 0, 1, \ldots$, $u^{(t)}_{EFCM-LP1}, t = 0, 1, \ldots$ 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. □

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 $v^{(t)}_{EFCM-LP1} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}, \mathbf{U}^{(t)}), t = 0, 1, \ldots,$ *converges.*

vi. 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 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 $v^{(t)}_{EFCM-Mk}, t = 0, 1, \ldots,$ $v^{(t)}_{EFCM-GS}, t = 0, 1, \ldots,$ $v^{(t)}_{EFCM-GP}, t = 0, 1, \ldots,$ $v^{(t)}_{EFCM-LP1}, t = 0, 1, \ldots$ 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. □

3.3.4 Complexity analysis

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

- Complexity time of $O(T \times \max\{N \times C \times P^2, P^3\})$ for EFCM-M.
- Complexity time of $O(T \times C \times \max\{N \times P^2, P^3\})$ for EFCM-Mk.
- Complexity time of $O(T \times N \times C \times P)$ for EFCM-GS2 and EFCM-GP2.
- Finally, complexity time of $O(T \times C \times N \times P \times \log(N))$ for EFCM-GS1, EFCM-GP1, EFCM-LP1 and EFCM-LS1.

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

3.4 EXPERIMENTAL RESULTS

This section evaluates the performance and usefulness of the proposed algorithms by applying them to suitable synthetic and real datasets. A summary of proposed methods is shown in Table 4. Also, research questions 1, 2 and 3 are answered. Proposed methods were compared with the four previous most related fuzzy clustering models: EFCM-2 and EFCM-1 (SADAAKI; MASAO, 1997), EFCM-LS2 (HANMANDLU et al., 2013) and EFCM-LP2 (RODRÍGUEZ; CARVALHO, 2017) algorithms.

3.4.1 Experimental setting

Optimal values for T_u and T_v were found using a grid search strategy. Following a procedure similar to Ref. (SCHWÄMMLE; JENSEN, 2010), for each dataset, the value of T_u varied

Table 4 – Summary of proposed methods.

Algorithms	Description
EFCM-M	Proposed method based on Global Mahalanobis distance
EFCM-Mk	Proposed method based on Local Mahalanobis distance
EFCM-GP2	Proposed entropy fuzzy method with global product restriction and adaptive Euclidean distance
EFCM-GP1	Proposed entropy fuzzy method with global product restriction and adaptive City-Block distance
EFCM-LP1	Proposed entropy fuzzy method with local product restriction and adaptive City-Block distance
EFCM-GS2	Proposed entropy fuzzy method with global sum restriction and adaptive Euclidean distance
EFCM-GS1	Proposed entropy fuzzy method with global sum restriction and adaptive City-Block distance
EFCM-LS1	Proposed entropy fuzzy method with local sum restriction and adaptive City-Block distance

Source: Author (2022)

from 0.01 to 100 (with step 0.01). 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 (with step 10). Fixing the value of T_v, the procedure abovementioned to compute T_u was followed. Subsequently, the pair (T_u, T_v) with the maximum distance was chosen. Note that the optimal values are calculated without supervision. The maximum number of iterations T and stop condition parameter ε were set to 100 and 10^{-5}, respectively.

Before running the algorithms, each dataset was normalized. The characteristics had a zero mean and a standard deviation of one. 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 the measures HUL and ARI.

3.4.2 Synthetic dataset experiments with different configurations

In the first experiment, four synthetic datasets described by two-dimensional vectors generated randomly from a normal distribution were created. The synthetic datasets were generated, having classes of different sizes and shapes as in Ref. (CARVALHO; TENÓRIO; JUNIOR, 2006). Each 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 5.

Four different data configurations were considered. Firstly, the class covariance matrices are diagonal and nearly equal (Figure 5 (a)). In the second configuration, the class covariance

Table 5 – 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	$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$	$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}$

Source: Author (2022)

matrices are diagonal but unequal (Figure 5 (b)). The class covariance matrices are not diagonal but almost the same for the third configuration (Figure 5 (c)). Finally, we consider that in addition to not being diagonal, the class covariance matrices are also unequal (Figure 5 (d)). Table 6 shows a detailed review of each synthetic data configuration.

Table 6 – Synthetic data configurations.

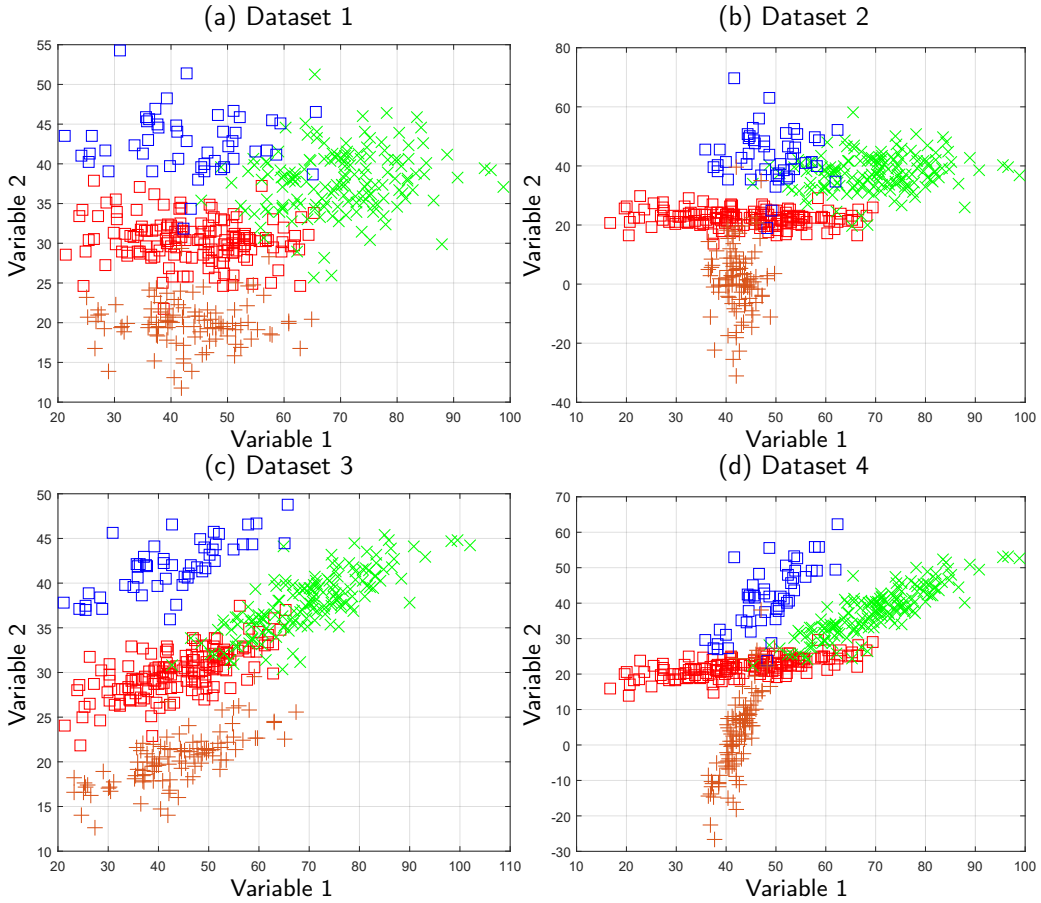
	Configuration 1				Configuration 2				Configuration 3				Configuration 4			
Classes	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
μ_1	45	70	45	42	45	70	50	42	45	70	45	42	45	70	50	42
μ_2	30	38	42	20	22	38	42	2	30	38	42	20	22	38	42	2
σ_1^2	100	81	100	81	144	81	36	9	100	81	100	81	144	81	36	9
σ_2^2	9	16	16	9	9	36	81	144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

Source: Author (2022)

Fifty replications of each synthetic dataset were carried out in a framework of a Monte Carlo experiment. For each dataset and algorithm, 50 random initializations were performed. 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 (FRIEDMAN, 1937) 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 perform equivalently is rejected under the Friedman test. In that case, the Nemenyi post-hoc test (NEMENYI, 1963) 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 the two algorithms are significantly different if the distance of the average ranks exceeds the critical distance. The objective is to determine whether at least one method is significantly better than at least one other method at the $\alpha = 0.05$ level.

Figure 5 – Clusters drawn from different data configurations. (a) The class covariance matrices are diagonal and nearly equal. (b) The class covariance matrices are diagonal but unequal. (c) The class covariance matrices are not diagonal but almost the same. (d) The class covariance matrices are neither diagonal nor equal.



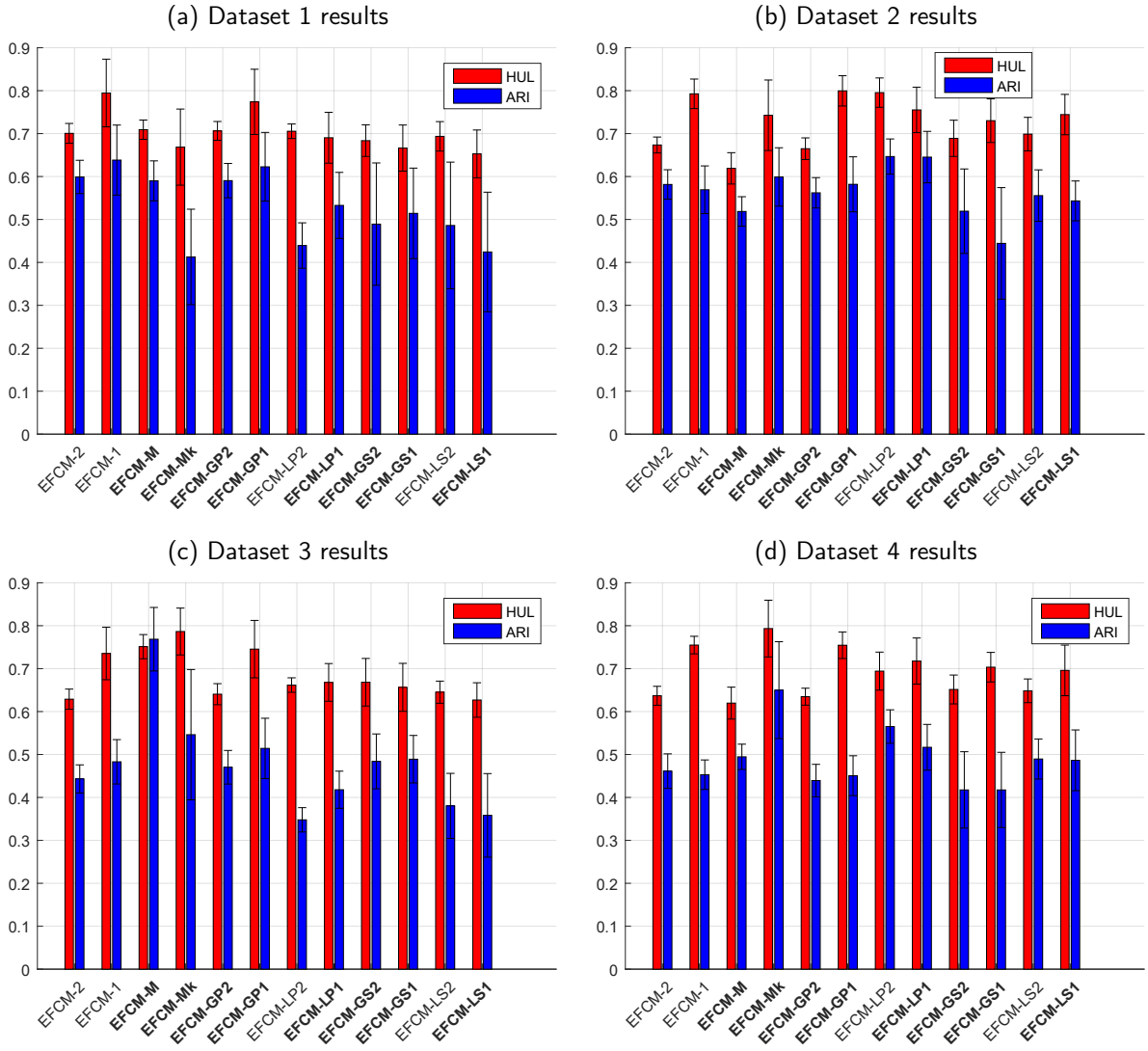
Source: Author (2022)

3.4.2.1 Results

Figure 6 shows the values of the mean and the standard deviation for HUL and ARI for all methods and data configurations. Numeric values can be observed in Table 26 (Appendix B). Proposed methods are highlighted in bold.

In data configuration 1 (the cluster covariance matrices are diagonal and almost the same), the best results according to the HUL index were obtained by EFCM-1, EFCM-GP1, and EFCM-M algorithms with values of 0.7946, 0.7739, and 0.7091, respectively. The best performance for ARI was presented by the EFCM-1 algorithm. 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, in this data configuration, almost all the methods with global adaptive distance outperformed their respective variants based on local

Figure 6 – Mean and standard deviation for the data configurations. (a) Dataset 1 results. (b) Dataset 2 results. (c) Dataset 3 results. (d) Dataset 4 results



Source: Author (2022)

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 ARI index, but the latter surpasses EFCM-GS2 regarding HUL index.

Data configuration 2 presents cluster covariance matrices that are diagonal but unequal. The best result was 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 obtained the worst performance for the HUL index and EFCM-GS1, EFCM-M and EFCM-GS2 for the ARI index. For this configuration, almost all the local adaptive distance methods presented better results than their respective variants based on global adaptive distance.

When the cluster covariance matrices are not diagonal but almost the same (data configuration 3), the first, second, and third best performance for HUL were presented by EFCM-GP1, EFCM-Mk, and EFCM-M, respectively. For the ARI index, the best results were achieved, respectively, by EFCM-M, EFCM-Mk, and EFCM-GP1. The worst result was presented, respectively, by the algorithms EFCM-LS1 and EFCM-GS1 showed the worst results for HUL, and EFCM-LP2 and EFCM-LS2 for ARI. Finally, for this data configuration and concerning the ARI index, the methods with global adaptive distance outperformed their respective variants based on local adaptive distance. Besides, as expected, the proposal EFCM-M was the best among all methods.

For the fourth data configuration, where 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, for this data configuration and concerning the ARI index, the methods with local adaptive distance outperformed their respective variants based on global adaptive distance.

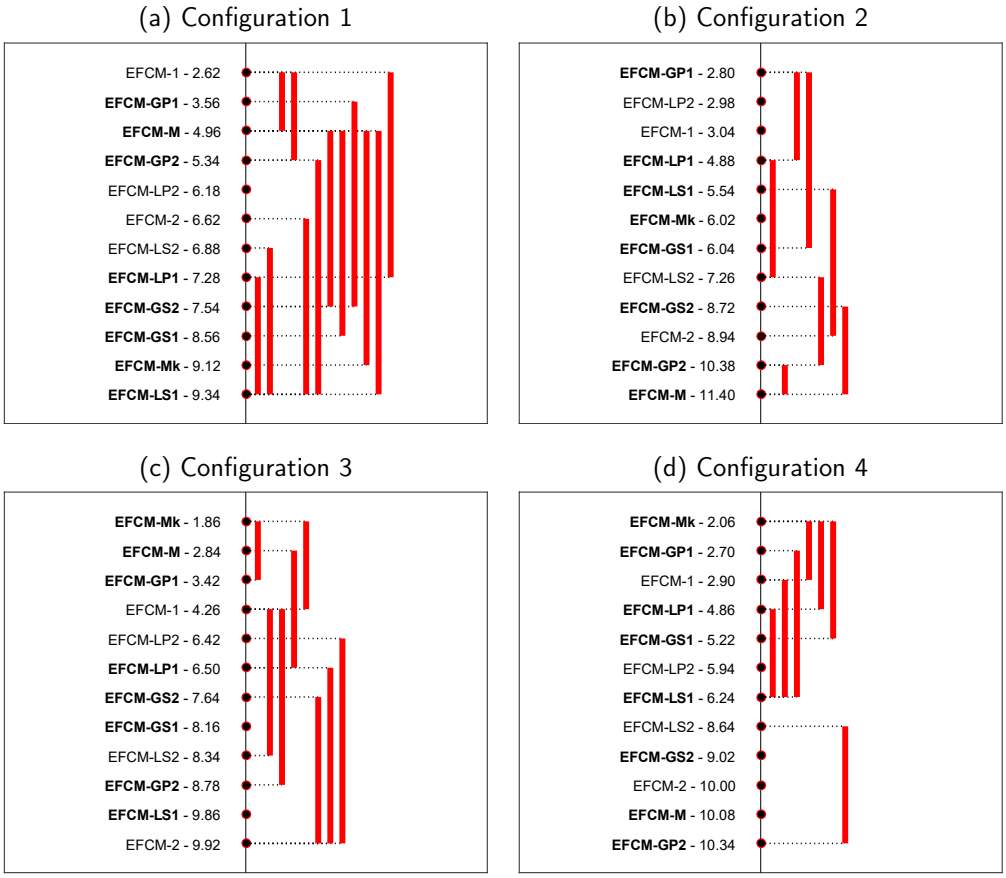
Figures 7 and 8 show the comparison of the algorithms against each other with the Nemenyi test. For the models joined by the horizontal lines, there is no evidence of statistical significant differences. The average performance rank is also presented. In this regard, concerning HUL index, Figure 7 shows 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 EFCM-M. In configuration 2, the best results were presented by EFCM-GP1, 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, Figure 8 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 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.

3.4.3 Synthetic datasets with different percentages of outliers

An outlier in a dataset is defined informally as an observation that is considerably different from the remainders as if a distinct mechanism generates it (HE; XU; DENG, 2003). Outliers occur due to mechanical faults, changes in system behavior, fraudulent behavior, and human errors. Manage outliers in the clustering process is an important issue in data mining with

Figure 7 – Comparison of the algorithms against each other through the Nemenyi test for HUL.



Source: Author (2022)

numerous applications, including credit card fraud detection, the discovery of criminal activities in electronic commerce, weather prediction, marketing, and customer segmentation.

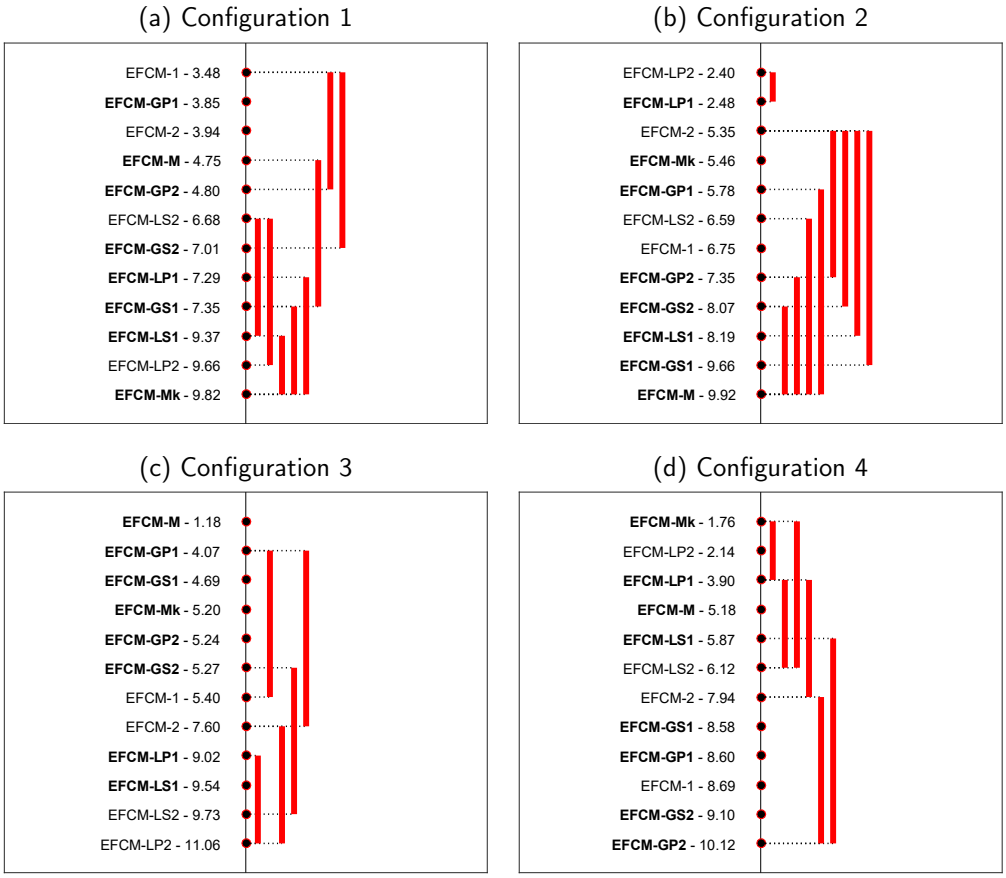
This experiment was developed to verify the behavior of the proposed methods in the presence of outliers. For this purpose, a synthetic dataset with 80 objects described by two-dimensional vectors was generated randomly from a normal distribution according to the parameters shown in Table 7 (Figure 9 (a)). Two different percentages of outliers (10% and 20%) have been added to the dataset to assess the robustness of the algorithms. Please see Figure 9 (b) and (c). Table 7 shows the configuration of the parameters for generating the outliers.

Table 7 – Parameter settings for different percentages of outliers.

	μ_1	μ_2	σ_1^2	σ_2^2
Class 1	0	0	0.05	0.05
Class 2	0.8	0.8	0.05	0.05
Outliers	0.8	1	5	5

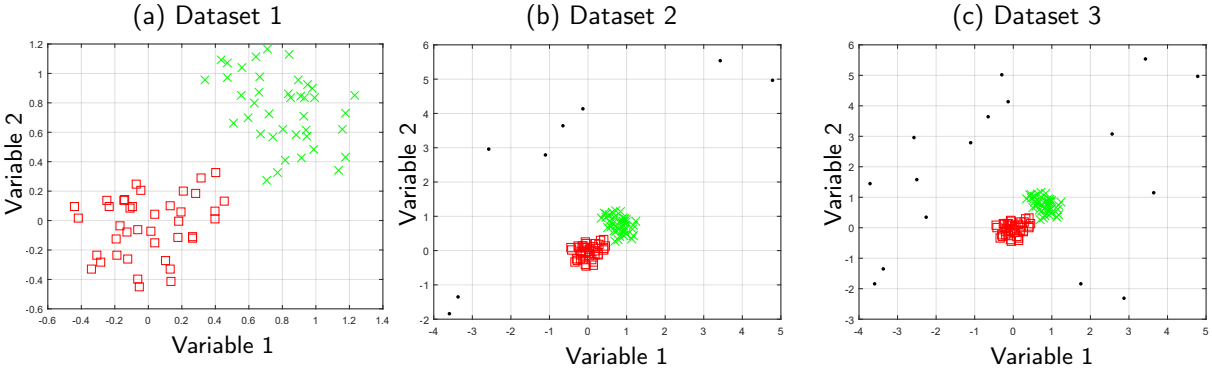
Source: Author (2022)

Figure 8 – Comparison of the algorithms against each other through the Nemenyi test for ARI.



Source: Author (2022)

Figure 9 – Synthetic dataset with different percentage of outliers. (a) Data with 0% of outliers, (b) data with 10% of outliers and (c) data with 20% of outliers.



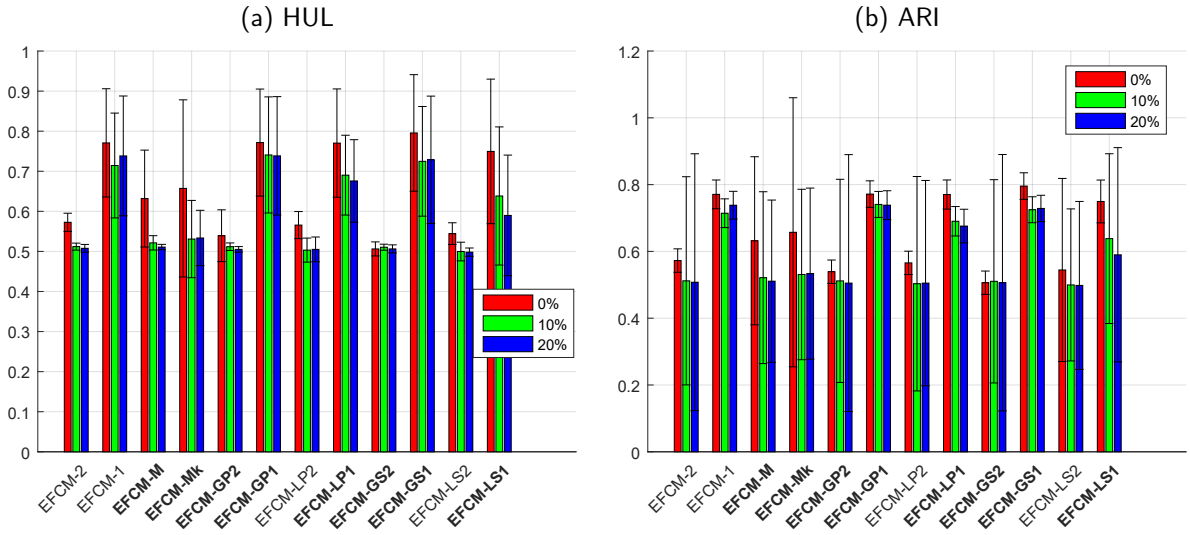
Source: Author (2022)

Fifty replications of the synthetic dataset were carried out in a framework of a Monte Carlo experiment. For each dataset, 50 random initializations of the clustering algorithm were performed. 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.

3.4.3.1 Results

Figure 10 shows the results for HUL and ARI with different percentages of outliers according to the mean and the standard deviation. Numeric values can be observed in Table 27 (Appendix B).

Figure 10 – Mean and standard deviation for different percentages of outliers according to the indices (a) HUL and (b) ARI, respectively.



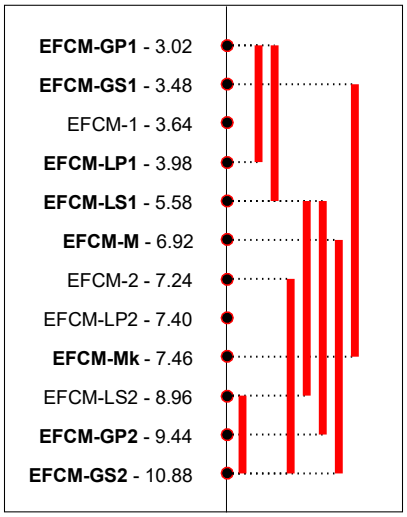
Source: Author (2022)

Figure 10 shows that for 0% of outliers, methods with the Euclidean distance presented the best results compared with those based on the Mahalanobis and City-Block distances. However, concerning misclassification measured in HUL and ARI with different percents of outliers, algorithms based on the City-Block distance outperform the other approaches, being able to identify the presence of clusters even in a noisy environment, and the performance clustering degrades very slowly as the percentage of outliers increases.

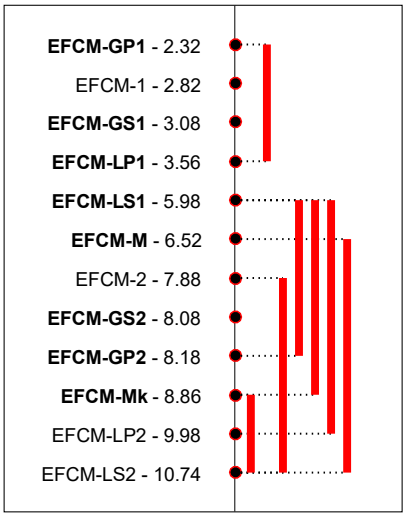
Figure 11 shows that for HUL, the City-Block distance-based methods achieved the highest average performance ranking, regardless of the percentage of outliers. Additionally, almost all algorithms showed similar results when analyzing the values for ARI with 0% percent of outliers, see Figure 12. However, 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.

Figure 11 – Comparison of the algorithms with each other with the Nemenyi test for HUL with different percentages of outliers.

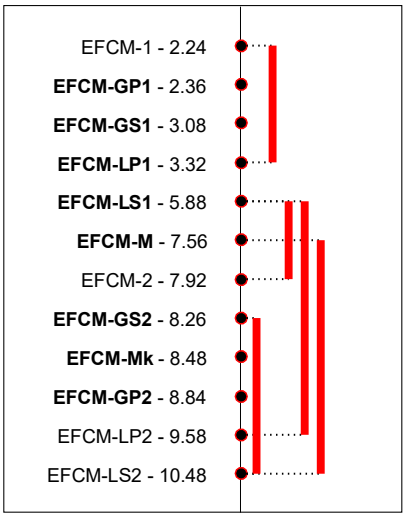
(a) Average performance ranking for 0% of outliers



(b) Average performance ranking for 10% of outliers



(c) Average performance ranking for 20% of outliers



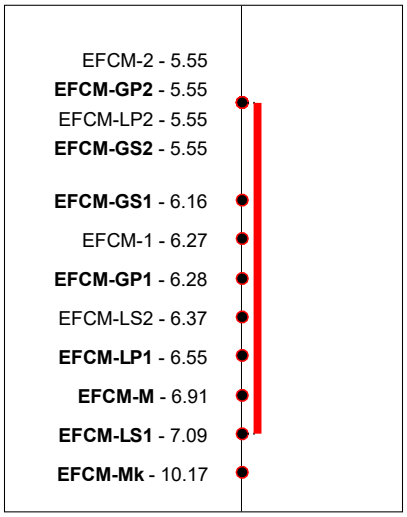
Source: Author (2022)

3.4.4 Real datasets

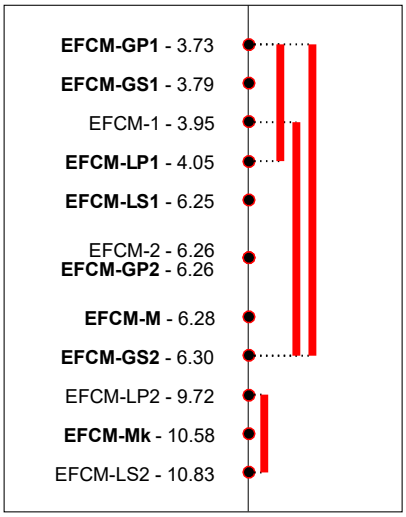
The previous and proposed algorithms were also applied on 15 real datasets available at the UCI machine learning repository (BACHE; LICHMAN, 2013): Automobile (Auto), Balance Scale, Haberman's Survival, Statlog (Heart), Image Segmentation, Ionosphere, Iris Plants, Mnist, Thyroid Gland, User Knowledge 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 N, P, and C represent the number of patterns, variables, and a priori classes, respectively. As can be seen, several sample sizes, number of

Figure 12 – Comparison of the algorithms with each other with the Nemenyi test for ARI with different percentages of outliers.

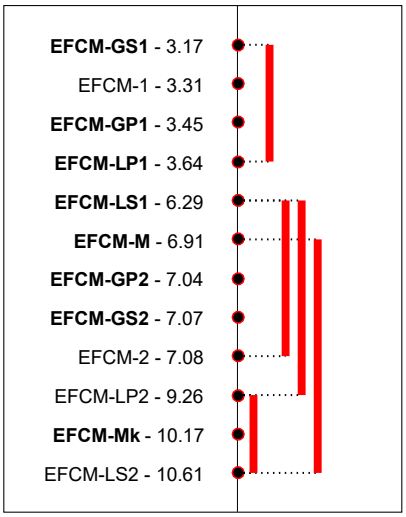
(a) Average performance ranking for 0% of outliers



(b) Average performance ranking for 10% of outliers



(c) Average performance ranking for 20% of outliers



Source: Author (2022)

attributes, and number of classes were considered.

The hyper-parameter values were chosen according to the same procedure used in the experiments with the synthetic datasets. 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. The HUL and ARI indices were considered to assess the misclassification.

Table 8 – Summary of the real datasets.

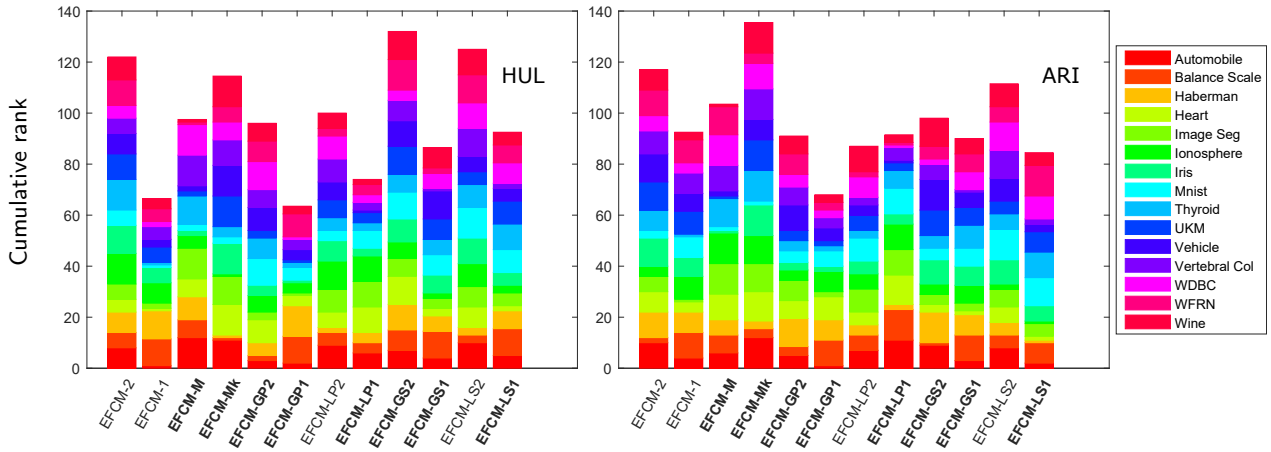
Dataset	N	P	C	Dataset	N	P	C	Dataset	N	P	C
Automobile	205	25	6	Balance Scale	625	4	3	Haberman	306	3	2
Heart	270	13	2	Image Segmentation	2310	19	7	Ionosphere	351	33	2
Iris plants	150	4	3	Mnist	14 780	784	2	Thyroid gland	215	5	3
UKM	403	5	4	Vehicle	846	18	4	Vertebral column	310	6	3
WDBC	569	30	2	WFRN	5456	4	4	Wine	178	13	3

Source: Author (2022)

3.4.4.1 Results

Table 28 gives the numeric values provided by the algorithms on real datasets and the performance rank in parenthesis. Besides, Figure 13 presents the cumulative rank according to both indices computed from Table 28. It is observed that the algorithm EFCM-GP1 offers the best performance for real datasets, regardless of the index considered. In this regard, the proposal with the global adaptive City-Block distance showed to be a promising approach for resolving real clustering problems. Moreover, the algorithms EFCM-1 and EFCM-LP1 achieved the second and third-best results for the HUL index, while EFCM-LS1 and EFCM-LP2 for ARI. Finally, the algorithms EFCM-GS2 and EFCM-Mk reached the worst results for HUL and ARI, respectively.

Figure 13 – Clustering results on real data for HUL and ARI according to Table 28.

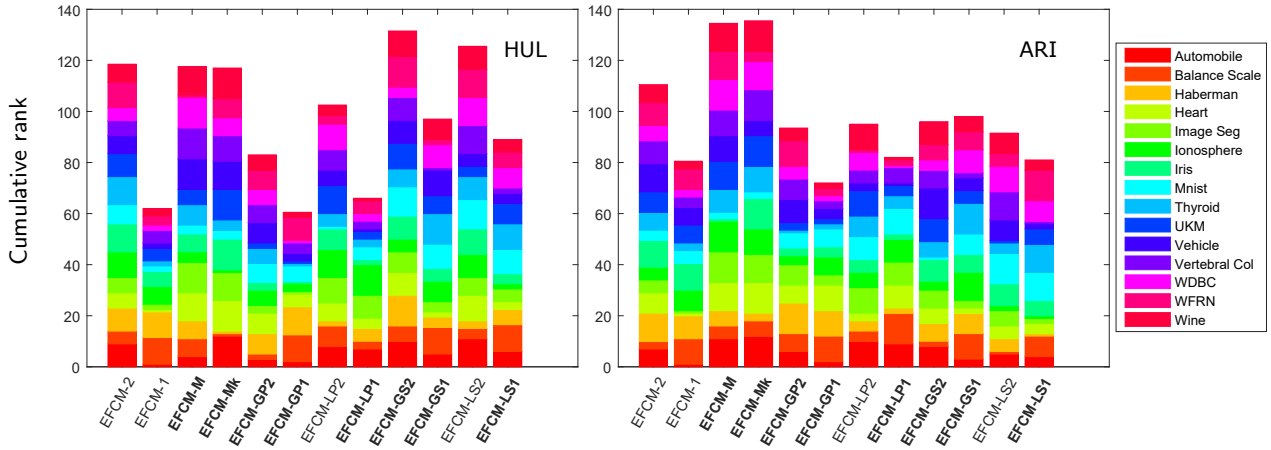


Source: Author (2022)

Table 29 gives the results provided by the algorithms on real datasets and the performance rank of each algorithm (in parenthesis) based on the mean and the standard deviation of the algorithm executions. Furthermore, Figure 14 shows the cumulative rank according to both indices computed from Table 29. It can be observed that the proposed algorithm EFCM-GP1 also achieves the best clustering results for both indices. Therefore, the method is robust when selecting the best result according to the objective function and all other cases. The algorithm

EFCM-GS2 obtained the worst results for HUL. The lowest values for ARI were presented by EFCM-M and EFCM-Mk.

Figure 14 – Clustering results on real data for HUL and ARI according to Table 29.



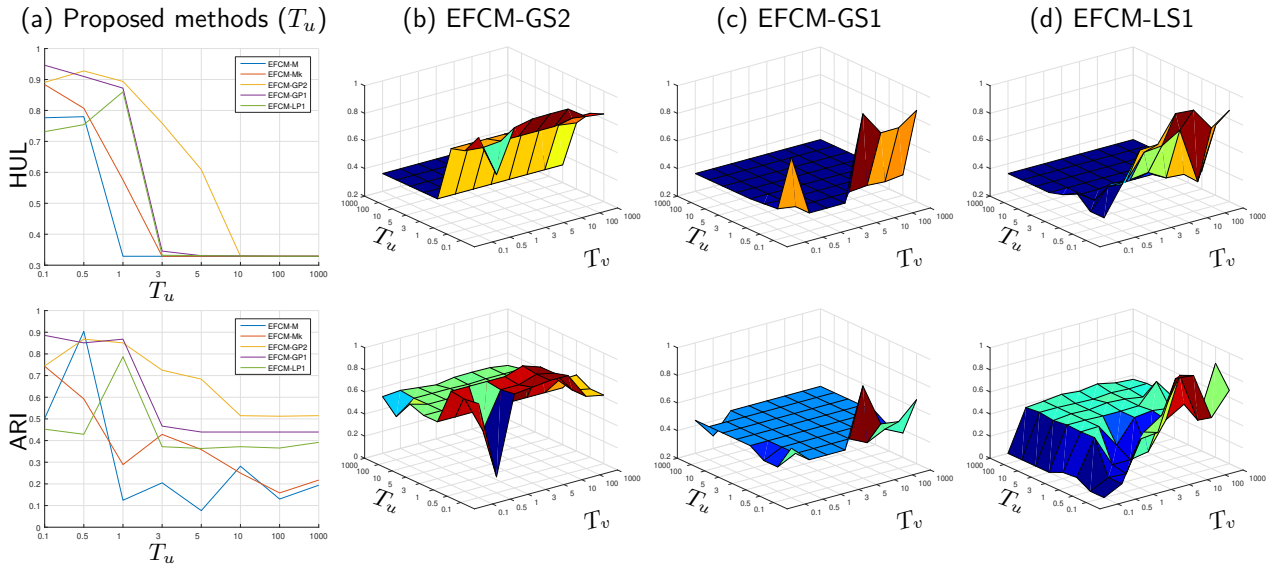
Source: Author (2022)

3.4.5 Analysis of the hyper-parameters 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 Figure 15. 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 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 Figure 16. 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. Besides, small values reveal distinct relevance, see Table 9. 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 $T_v = 1000$.

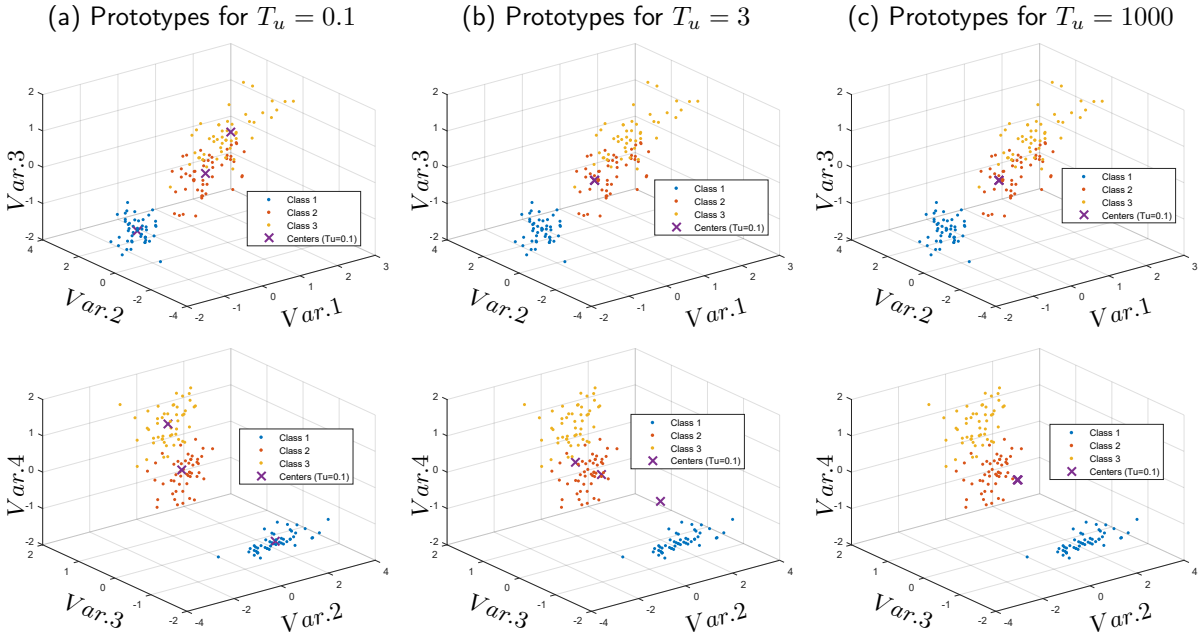
In general, even though selecting the optimal value for the hyper-parameters in an unsupervised manner is a difficult task, it was noted that acceptable solutions could be obtained by avoiding values that lead to nearby centroids. Addressing such, the optimal values for T_u and T_v were chosen without supervision, as described in Section 3.4. In this sense, a new perspective was provided for fuzzifying the clusterization of the units while ensuring maximum cluster compactness.

Figure 15 – Effect of the hyper-parameters T_u and T_v on the Iris dataset. (a) Performance of EFCM-M, EFCM-Mk, EFCM-GP2, EFCM-GP1 and EFCM-LP1 for HUL and ARI with different values of T_u. Figures (b), (c) and (d) shows the performances of the algorithms EFCM-GS2, EFCM-GS1 and EFCM-LS1 varying T_u and T_v.



Source: Author (2022)

Figure 16 – Prototypes obtained using the proposed method EFCM-GS2 on the Iris dataset for (a) $T_u = 0.1$, (b) $T_u = 3$ and (c) $T_u = 1000$.



Source: Author (2022)

3.4.6 Brodatz texture images for segmentation

Image segmentation is a challenging but essential task in several imaging analyses or computer vision applications (PAL; PAL, 1993). Texture image segmentation plays an essential

Table 9 – 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) $T_v = 1000$.

Variables	$T_v = 0.1$	$T_v = 3$	$T_v = 1000$
1	0	0	0.2485
2	0	0	0.2420
3	1	0.9303	0.2554
4	0	0.0697	0.2541

Source: Author (2022)

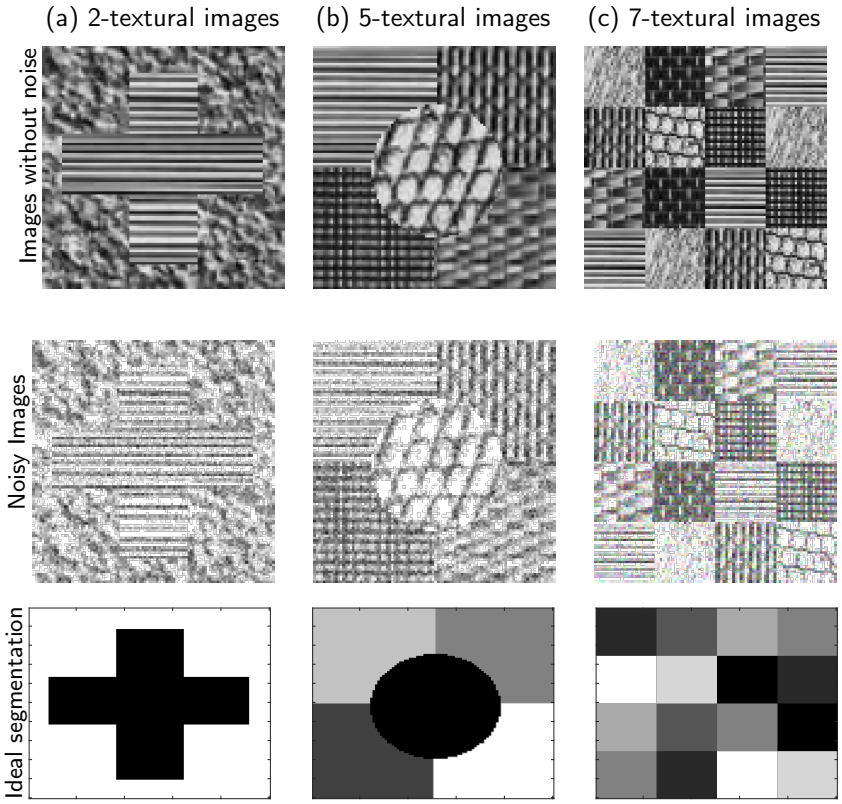
role in the human visual system for recognition and interpretation (AKBARIZADEH; RAHMANI, 2017; QIAN et al., 2017; DU et al., 2018). It aims to segment a texture image into several regions with different texture features, providing surface characteristics for analyzing many types of images, including 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 (RANDEN,), and are employed to demonstrate the performance of the algorithms for datasets with high dimensionality. The first and second rows of Figure 17 show the six texture images with and without Gaussian noise (mean 0, variance 0.3). Moreover, the last row of Fig. 17 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 and D49), five-textural images (D21, D22, D49, D53, and D55), and seven-textural images (D3, D6, D21, D49, D53, D56, and D93).

3.4.6.1 Experimental setting

The features of the texture images were extracted using the Gabor filter as in Ref. (KYRKI; KAMARAINEN; KÄLVIÄINEN, 2004). 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. The best results were selected according to their respective objective function. In the experiments, the number of clusters is assumed to be equal to 2, 5, and 7 in images with two, five, and seven textures, respectively.

Figure 17 – Two, five, and seven-textural images with and without Gaussian noise.



Source: Author (2022)

3.4.6.2 Results

Tables 10 and 11 present the obtained clustering for the algorithms on images without and with noise, respectively, for the best result according to their objective functions. Besides, Figures 18 to 20 show the unsupervised segmentation results using each algorithms on 2, 4 and 7 textural images 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. Besides, 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, EFCM-GP1 and EFCM-GP2 obtained the best

Table 10 – HUL and ARI index for 2, 5 and 7 textural image without noise for the best solution according to the objective function.

	2-Textural Image		5-Textural Image		7-Textural Image	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5547 (6)	0.7438 (4)	0.6484(5)	0.7572 (3)	0.7577 (3)	0.4716 (5)
EFCM-1	**0.7853** (1)	0.7222 (7)	0.5780 (6)	0.7502 (4)	0.7225 (6)	0.5191 (2)
EFCM-M	0.5398 (10)	0.0079 (10)	0.2990 (12)	0.1906 (10)	**0.8765** (1)	**0.5231** (1)
EFCM-Mk	0.5365 (11)	-0.0236 (11)	0.5558 (8)	0.1792 (11)	0.6561 (10)	0.2157 (10)
EFCM-GP2	0.5710 (5)	0.7438 (4)	0.6873 (3)	0.7740 (2)	0.8063 (2)	0.4981 (4)
EFCM-GP1	0.6980 (3)	0.7571 (2)	0.6487 (4)	**0.7826** (1)	0.6939 (7)	0.5051 (3)
EFCM-LP2	0.5506 (7)	0.7266 (6)	0.6964 (2)	0.4811 (8)	0.6581 (9)	0.2632 (9)
EFCM-LP1	0.7005 (2)	**0.7740** (1)	0.5580 (7)	0.6054 (6)	0.6862 (8)	0.4319 (8)
EFCM-GS2	0.5399 (9)	0.7438 (4)	0.5160 (10)	0.4860 (7)	0.7403 (5)	0.4666 (6)
EFCM-GS1	0.5925 (4)	0.7114 (8)	0.5053 (11)	0.1230 (12)	0.6003 (12)	0.1081 (11)
EFCM-LS2	0.5359 (12)	-0.0430 (12)	**0.7629** (1)	0.4379 (9)	0.7404 (4)	0.4353 (7)
EFCM-LS1	0.5420 (8)	0.1129 (9)	0.5368 (9)	0.6319 (5)	0.6267 (11)	0.0743 (12)

Source: Author (2022)

Table 11 – HUL and ARI index for 2, 5 and 7 textural image with Gaussian noise for the best solution according to the objective function.

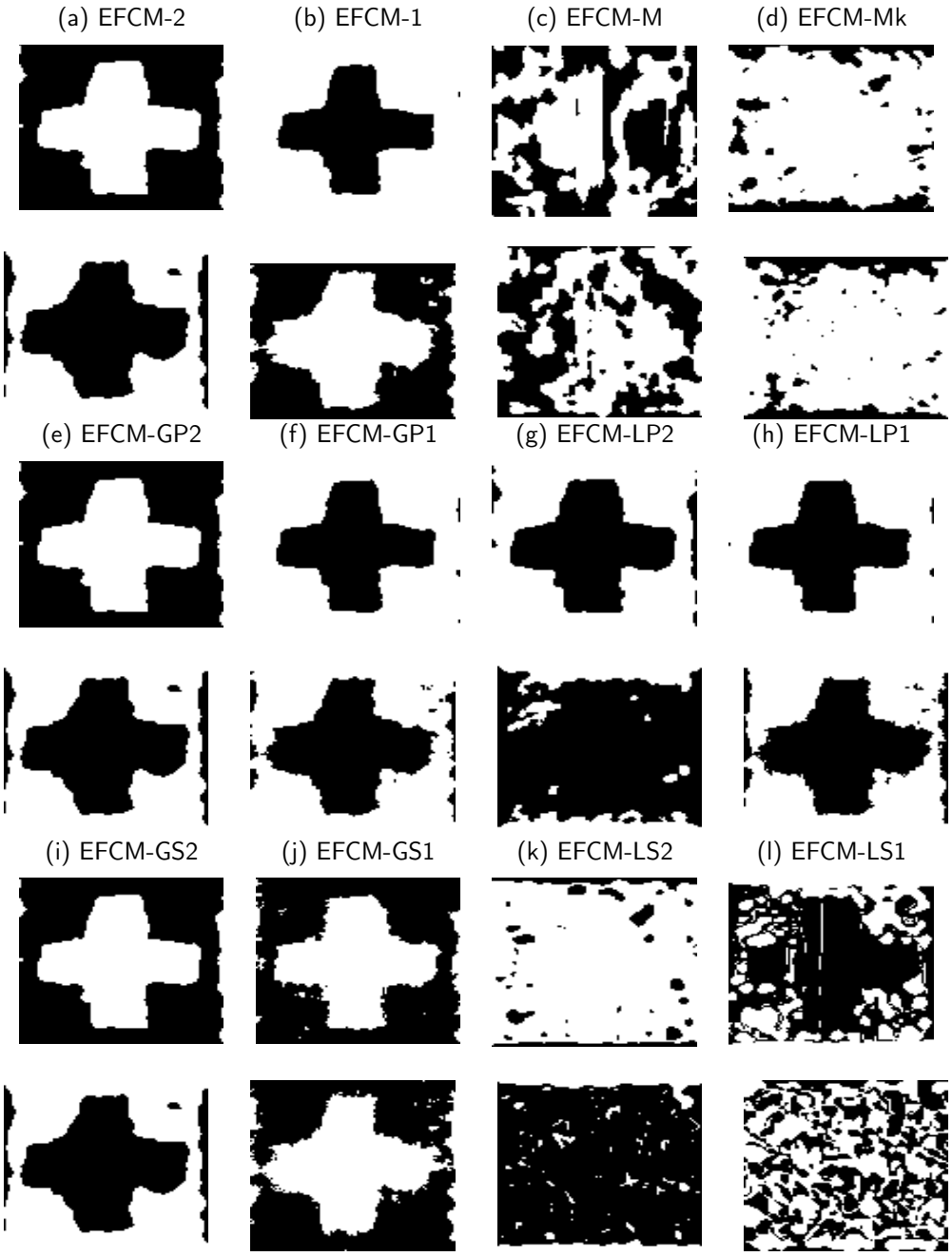
	2-Textural Image		5-Textural Image		7-Textural Image	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5694 (4)	0.5863 (7)	0.4795 (9)	0.4523 (6)	0.6437 (4)	0.2894 (6)
EFCM-1	0.630 (2)	0.6328 (2)	0.6164 (5)	0.5925 (4)	0.5830 (8)	0.3274 (2)
EFCM-M	0.5392 (8)	0.0434 (8)	0.2021 (12)	0.0678 (12)	0.1484 (12)	0.0390 (12)
EFCM-Mk	0.5195 (12)	0.0015 (9)	0.4318 (10)	0.1805 (10)	0.5412 (9)	0.1373 (9)
EFCM-GP2	0.5403 (5)	0.5866 (5.5)	**0.7592** (1)	**0.6730** (1)	**0.6964** (1)	0.3219 (3)
EFCM-GP1	**0.6581** (1)	**0.6427** (1)	0.6589 (4)	0.6515 (2)	0.6808 (2)	**0.4451** (1)
EFCM-LP2	0.5344 (10)	-0.0015 (11)	0.6910 (3)	0.3515 (9)	0.6139 (7)	0.2076 (8)
EFCM-LP1	0.5759 (3)	0.5960 (4)	0.7028 (2)	0.6102 (3)	0.6608 (3)	0.3000 (4)
EFCM-GS2	0.5399 (7)	0.5866 (5.5)	0.4844 (8)	0.4692 (5)	0.4767 (10)	0.2912 (5)
EFCM-GS1	0.5403 (6)	0.6045 (3)	0.5269 (7)	0.1018 (11)	0.6418 (5)	0.0814 (10)
EFCM-LS2	0.5361 (9)	-0.0375 (12)	0.4309 (11)	0.4246 (8)	0.4745 (11)	0.2873 (7)
EFCM-LS1	0.5215 (11)	-0.0001 (10)	0.5923 (6)	0.4499 (7)	0.6264 (6)	0.0592 (11)

Source: Author (2022)

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.

Tables 12 and 13 show the mean of the metrics indices for the 10 executions of the

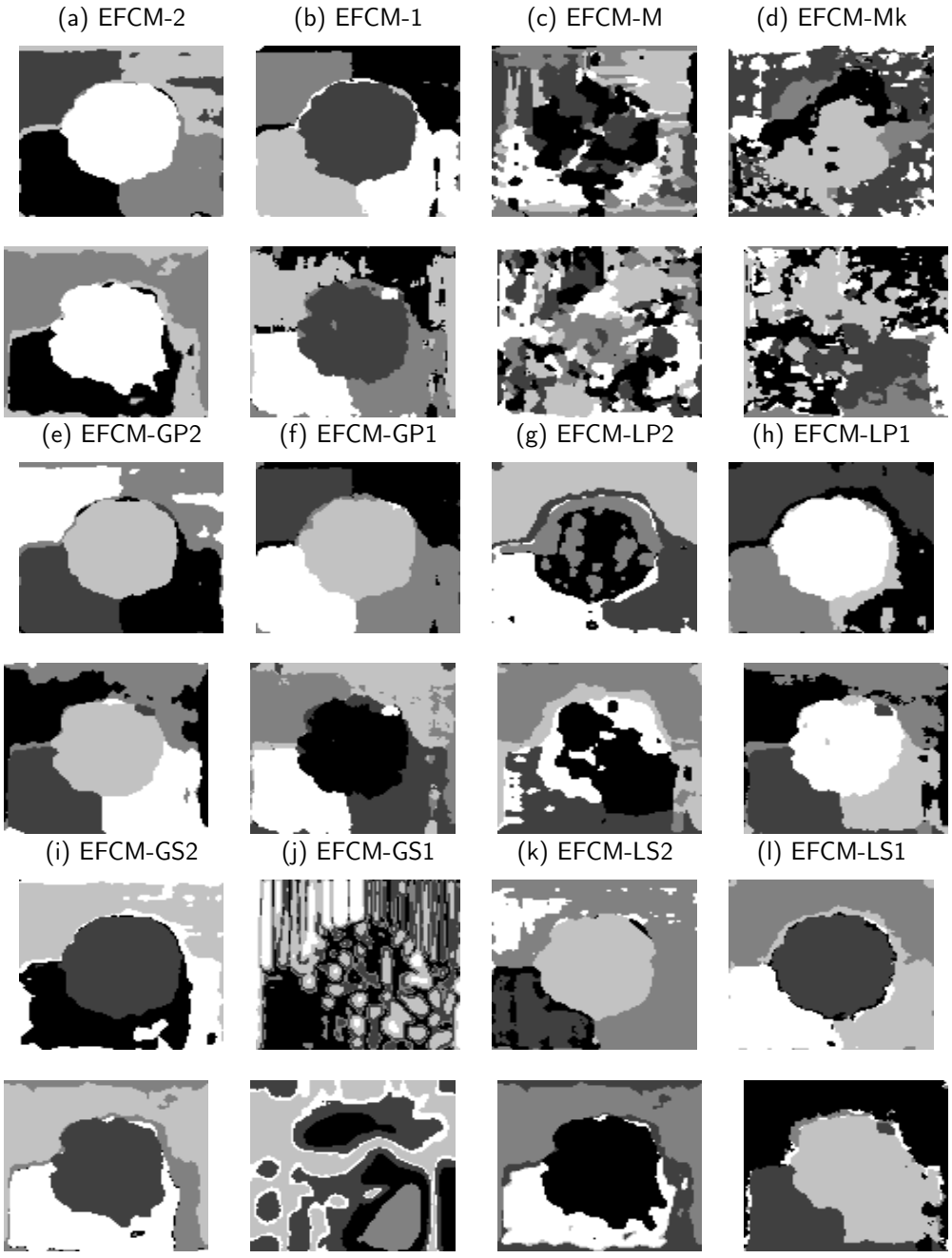
Figure 18 – Segmentation results for the 2-textural image with and without Gaussian noise. The first, third, and fifth rows show the segmentation results for the original 2-textural image. The second, fourth, and sixth rows show the obtained segmentation for the 2-textural image with noise.



Source: Author (2022)

algorithms for HUL and ARI on images with and without and noise. In the case of the 2-textural image without noise, we can see that EFCM-1 and EFCM-LP1 present the best clustering results, showing stability when often converging to the same solution. Besides, EFCM-LS2 and EFCM-GP1 as in Table 10 also achieve the best results for 5-textural image. However, for the 7-textural image, the proposed algorithm EFCM-GP2 obtains the best segmentation on

Figure 19 – Segmentation results of each algorithm for the 5-textural image with and without Gaussian noise. The first, third and fifth rows show the segmentation results for the original 5-textural image. The second, fourth and sixth rows present the obtained segmentation for the 5-textural image with Gaussian noise.

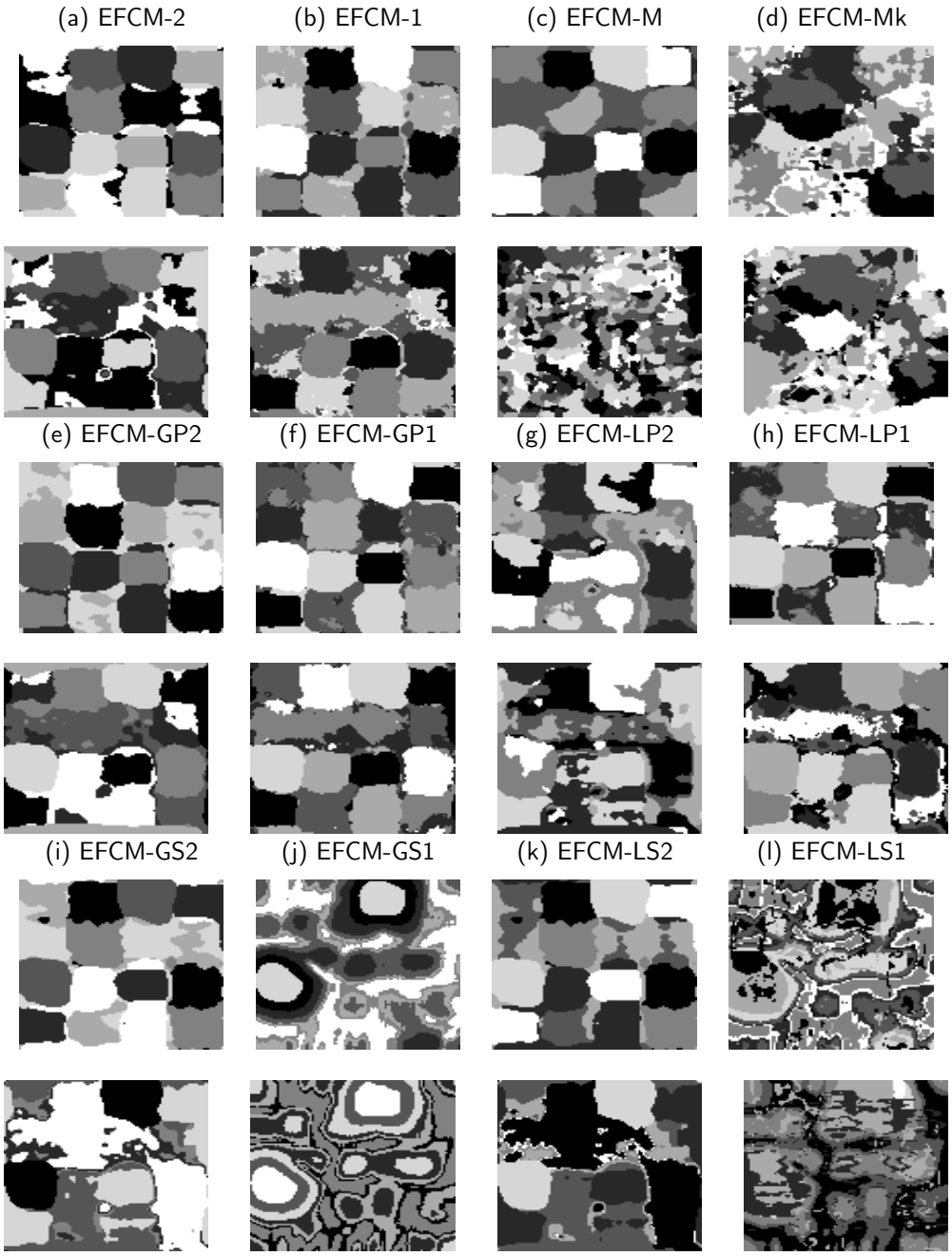


Source: Author (2022)

average.

On the other hand, for images with noise, as in Table 11, methods based on City-Block distance degrade their performance more slowly. For 2-textural image, EFCM-GP1 obtains the best segmentation as in Table 11. In images with 5 and 7 textures, on average, algorithms EFCM-GP2 and EFCM-GP1 present the best clustering results for HUL and ARI, respectively.

Figure 20 – Segmentation results of each algorithm for the 7-textural image with and without Gaussian noise. The first, third and fifth rows show the segmentation results for the original 7-textural image. The second, fourth and sixth rows present the obtained segmentation for the 7-textural image with Gaussian noise.



Source: Author (2022)

3.5 CONCLUSIONS

New fuzzy clustering algorithms for real-valued data based on suitable adaptive Euclidean, Mahalanobis, and City-Block distances and entropy regularization have been proposed. Moreover, adaptive distances have been introduced that change at each algorithm iteration. They

Table 12 – Mean of the HUL and ARI index for 2, 5 and 7 textural image without noise.

	2-Textural Image		5-Textural Image		7-Textural Image	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5464 (6.0) (0.0049)	0.7438 (3.0) (0.0001)	0.6146 (5.0) (0.0215)	0.6680 (4.0) (0.0954)	0.7543 (2.0) (0.0031)	0.4736 (3.0) (0.0020)
EFCM-1	**0.7853** (1.0) (0.0000)	0.7222 (7.0) (0.0000)	0.5733 (6.0) (0.0050)	0.7115 (3.0) (0.0577)	0.6723 (7.0) (0.0465)	0.4538 (6.0) (0.0642)
EFCM-M	0.5398 (10.0) (0.0000)	0.0151 (10.0) (0.0189)	0.2152 (12.0) (0.0295)	0.1374 (11.0) (0.0563)	0.6026 (12.0) (0.2303)	0.3419 (8.0) (0.1099)
EFCM-Mk	0.5384 (11.0) (0.0012)	-0.0260 (11.0) (0.0013)	0.5419 (8.0) (0.0100)	0.1846 (10.0) (0.0033)	0.6415 (10.0) (0.0082)	0.2728 (10.0) (0.0224)
EFCM-GP2	0.5733 (4.0) (0.0021)	0.7438 (4.5) (0.0000)	0.6724 (3.0) (0.0138)	0.7465 (2.0) (0.0251)	**0.8070** (1.0) (0.0004)	**0.4975** (1.0) (0.0007)
EFCM-GP1	0.6822 (3.0) (0.0499)	0.7515 (2.0) (0.0177)	0.6317 (4.0) (0.0232)	**0.7685** (1.0) (0.0288)	0.6930 (5.0) (0.0015)	0.4971 (2.0) (0.0054)
EFCM-LP2	0.5442 (7.0) (0.0037)	0.7257 (6.0) (0.0020)	0.6917 (2.0) (0.0098)	0.4823 (8.0) (0.0063)	0.6672 (9.0) (0.0100)	0.3035 (9.0) (0.0337)
EFCM-LP1	0.6844 (2.0) (0.0508)	**0.7631** (1.0) (0.0345)	0.5671 (7.0) (0.0125)	0.6238 (5.0) (0.0174)	0.6867 (6.0) (0.0004)	0.4408 (7.0) (0.0047)
EFCM-GS2	0.5400 (9.0) (0.0000)	0.7438 (4.5) (0.0000)	0.5160 (10.0) (0.0000)	0.4859 (7.0) (0.0001)	0.7410 (3.0) (0.0015)	0.4676 (4.0) (0.0069)
EFCM-GS1	0.5538 (5.0) (0.0213)	0.6898 (8.0) (0.0148)	0.5041 (11.0) (0.0039)	0.1217 (12.0) (0.0031)	0.6697 (8.0) (0.0733)	0.1584 (11.0) (0.0534)
EFCM-LS2	0.5359 (12.0) (0.0000)	-0.0430 (12.0) (0.0000)	**0.7285** (1.0) (0.0258)	0.4122 (9.0) (0.0371)	0.7408 (4.0) (0.0007)	0.4643 (5.0) (0.0102)
EFCM-LS1	0.5420 (8.0) (0.0000)	0.1129 (9.0) (0.0000)	0.5377 (9.0) (0.0062)	0.5945 (6.0) (0.0422)	0.6371 (11.0) (0.0059)	0.1219 (12.0) (0.0170)

Source: Author (2022)

can either be the same for all clusters or different 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 the Friedman test, the Nemenyi post-hoc test was

Table 13 – Mean of HUL and ARI for 2, 5 and 7 textural image with Gaussian noise.

	2-Textural Image		5-Textural Image		7-Textural Image	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5510 (4.0) (0.0088)	0.5856 (7.0) (0.0031)	0.4972 (9.0) (0.0216)	0.4928 (6.0) (0.0519)	0.6439 (3.0) (0.0001)	0.2933 (5.0) (0.0019)
EFCM-1	0.6302 (2.0) (0.0000)	0.6328 (2.0) (0.0000)	0.6165 (5.0) (0.0001)	0.6070 (4.0) (0.0052)	0.5822 (8.0) (0.0005)	0.3259 (2.0) (0.0032)
EFCM-M	0.5386 (8.0) (0.0003)	0.0178 (8.0) (0.0155)	0.2022 (12.0) (0.0001)	0.0993 (12.0) (0.0310)	0.1484 (12.0) (0.0000)	0.0685 (11.0) (0.0212)
EFCM-Mk	0.5196 (12.0) (0.0004)	0.0001 (9.0) (0.0038)	0.4243 (11.0) (0.0062)	0.1988 (10.0) (0.0201)	0.5362 (9.0) (0.0063)	0.1421 (9.0) (0.0032)
EFCM-GP2	0.5450 (5.0) (0.0025)	0.5866 (5.5) (0.0000)	**0.7563** (1.0) (0.0063)	0.6525 (2.0) (0.0417)	**0.6975** (1.0) (0.0020)	0.3239 (3.0) (0.0067)
EFCM-GP1	**0.6581** (1.0) (0.0000)	**0.6427** (1.0) (0.0000)	0.6585 (4.0) (0.0002)	**0.6586** (1.0) (0.0053)	0.6045 (7.0) (0.0268)	**0.3559** (1.0) (0.0318)
EFCM-LP2	0.5363 (9.0) (0.0011)	-0.0100 (11.0) (0.0037)	0.6920 (3.0) (0.0027)	0.3753 (9.0) (0.0378)	0.6153 (6.0) (0.0020)	0.2085 (8.0) (0.0151)
EFCM-LP1	0.5659 (3.0) (0.0105)	0.5920 (3.0) (0.0062)	0.7030 (2.0) (0.0003)	0.6120 (3.0) (0.0027)	0.6632 (2.0) (0.0027)	0.3200 (4.0) (0.0236)
EFCM-GS2	0.5399 (7.0) (0.0000)	0.5866 (5.5) (0.0000)	0.5128 (8.0) (0.0168)	0.5254 (5.0) (0.0432)	0.4739 (10.0) (0.0028)	0.2908 (6.0) (0.0007)
EFCM-GS1	0.5403 (6.0) (0.0002)	0.5920 (4.0) (0.0098)	0.5269 (7.0) (0.0001)	0.1018 (11.0) (0.0000)	0.6417 (4.0) (0.0002)	0.0814 (10.0) (0.0000)
EFCM-LS2	0.5361 (10.0) (0.0000)	-0.0375 (12.0) (0.0000)	0.4297 (10.0) (0.0044)	0.4133 (7.0) (0.0091)	0.4646 (11.0) (0.0129)	0.2907 (7.0) (0.0029)
EFCM-LS1	0.5215 (11.0) (0.0000)	-0.0001 (10.0) (0.0000)	0.5900 (6.0) (0.0009)	0.4124 (8.0) (0.0154)	0.6263 (5.0) (0.0042)	0.0587 (12.0) (0.0047)

Source: Author (2022)

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. A sensitivity analysis of such hyper-parameters was performed to measure their influence on the quality of the clustering. Furthermore, an unsupervised process to compute their optimal value was introduced.

The experimental results on synthetic datasets 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 clustering algorithms with global distances achieved the highest performances for both indices.

With these results, the research questions of this work were answered as follows:

1. Will Mahalanobis distance-based approaches really be more appropriate in certain situations than other clustering methods?

    - The commonly used Euclidean distance metric restricts conventional algorithms to datasets with hyper-spherical clusters and linearly separable characteristics. However, a distance metric should identify essential features and discriminate relevant and irrelevant features. The Mahalanobis distance is a measure between two objects in the space defined by relevant features. Since it accounts for unequal variances and correlations between features, it will adequately evaluate the distance by assigning different weights or important factors to the characteristics of data points. Only when the features are uncorrelated can the distance under a Mahalanobis distance metric be identical to that under the Euclidean distance metric. Besides, geometrically, a Mahalanobis distance metric can adjust the geometrical distribution of data so that the distance between similar data points is small.
    - As expected, the experiments on synthetic and real datasets showed that the Mahalanobis distance-based methods outperform other approaches for data with correlated characteristics.

2. How can the detrimental effect of outliers be avoided?

    - The City-Block distance is the sum of the absolute differences across dimensions. Generally, it yields results similar to the Euclidean distance. However, it has the advantage that the effect of outlier is minimized as the differences are not squared. Then, using such robust dissimilarity function allowed us to minimize the detrimental effect of outliers present in the data.

- Experiments with different amounts of outliers and noise showed that methods based on the City-Block distance degrade their performance more slowly than those with the Euclidean and Mahalanobis distances. Furthermore, the clustering results obtained showed smoothness and lower error variance.

3. How to simplify the definition of the optimization problem to reduce the number of parameters?

   - Fuzzy clustering algorithms with adaptive distances locate groups in different subspaces of the same dataset by assigning a weight to each dimension to measure the contribution of individual features to the formation of the clusters. Despite the popularity of such a technique, tuning hyper-parameters is difficult in general due to the lack of ground truth for validation. However, the success of most clustering methods depends heavily on the correct choice of the involved hyper-parameters.
   - New fuzzy clustering algorithms have been proposed where the product of the relevance of the variables is equal to one. As can be seen in the analysis of the hyper-parameter configuration section, such restriction allows us to adjust fewer hyper-parameters in the model and obtain good clustering results faster than other approaches.

4. Can approaches with the same set of relevant variables to all groups improve clustering results?

   - Feature weighting clustering employs a common weight vector for the whole dataset in the clustering procedure. However, soft subspace clustering is distinct in that different weight vectors are assigned to different clusters. In this work, both approaches have been considered.
   - Experiments show that methods in which the set of relevant variables is the same for all groups are appropriate when the internal dispersion of the clusters is almost the same. Furthermore, such methods presented the best results on real datasets and for texture image segmentation.

4 INTERVAL-VALUED DATA CLUSTERING WITH AUTOMATIC VARIABLES WEIGHTING

4.1 INTRODUCTION

This chapter starts with a review of interval-valued data clustering. New clustering algorithms for interval-valued data are proposed based on adaptive Euclidean and City-Block distances. The proposals consider the joint relevance of the variables for lower and upper boundaries, since in some cases, a boundary may play a minor role compared to the other. The adaptive distances change at each iteration of the algorithms and can be different from one cluster to another. The algorithms optimize an objective function alternating three steps for obtaining the representatives of each group, a fuzzy partition, and the relevance weights of the variables. Experiments on synthetic and real datasets corroborate the robustness and usefulness of the proposed methods.

4.2 INTERVAL-VALUED DATA CLUSTERING

There are two common representations of the objects upon which clustering can be based: relational data and usual or symbolic feature data. When a relationship represents each pair of objects, we have relational data. Alternatively, when each object is described by a vector of quantitative or qualitative values, the vectors expressing the items are called a feature dataset. When each complex object is defined by a vector of sets of categories, intervals, or weight histograms, the set of vectors representing the objects is called a symbolic feature dataset mainly studied in SDA. This work focuses on interval-valued data that is a particular type of symbolic data used in applications such as daily interval stock prices, ranges of fluctuations of some physical measurements, or monthly temperature in meteorological stations (Table 14). According to Ref. (D'URSO et al., 2017), there are several real cases where the empirical information is imprecise, being described by intervals. In particular, the following situations can be distinguished:

- Interval-valued data may occur due to a lack of knowledge, i.e., when the true value of a variable is unknown, and only an interval of values including the true value is available. Thus, the information available is imprecise and can therefore not be accurately represented using a single value.
- Interval-valued data may arise as a result of aggregating huge databases, which can not be analyzed in their original form.
- The data are intrinsically interval-valued, i.e, the phenomena are naturally explained by using intervals.

Table 14 – Minimal and maximal monthly temperatures recorded at 60 meteorological stations in China (1998).

Stations	Monthly temperature ($[min : max]$)				
	January	February	...	November	December
AnQing	$[1.8, 7.1]$	$[2.1, 7.2]$	...	$[7.8, 17.9]$	$[4.3, 11.8]$
⋮	⋮	⋮	⋮	⋮	⋮
ZhiJiang	$[2.7, 8.4]$	$[2.7, 8.7]$	...	$[8.2, 20]$	$[5.1, 13.3]$

Source: Author (2022)

An interval-valued data can be formalized as: $e_i\,(1 \leq i \leq N)$ is the i-th object represented by a vector $\mathbf{x}_i = (x_{i1}, ..., x_{iP})$, where $x_{ij} = [a_{ij}, b_{ij}]$, with $a_{ij} \leq b_{ij}$, which is the interval value taken by the j-th variable $(1 \leq j \leq P)$. The lower and upper bounds of the interval are denoted as a_{ij} and b_{ij}, respectively. Each object is represented geometrically by a hyperrectangle in $\mathbb{R}^P$ having 2^P vertices. The 2^P vertices correspond to all the possible (lower bound, upper bound) combinations. Alternatively, an interval valued data can be represented in terms of its midpoint (centers) $m_{ij} = \frac{a_{ij}+b_{ij}}{2}, i = 1, \ldots, N; j = 1, \ldots, P$, and of its radius (spreads) $r_{ij} = \frac{b_{ij}-a_{ij}}{2}, i = 1, \ldots, N; j = 1, \ldots, P$. In this way, the lower and upper bounds of the interval-valued data can be obtained as $m_{ij}-r_{ij}$ and $m_{ij}+r_{ij}$, respectively. Since the examined variables are interval-valued, each prototype $\mathbf{g}_k = (g_{k1}, ..., g_{kP})$ is a vector of P intervals with $g_{kj} = [\alpha_{kj}, \beta_{kj}]\,(1 \leq j \leq P; 1 \leq k \leq C)$.

Several clustering methods are available to manage symbolic data. For example, Irpino and Verde (IRPINO; VERDE, 2008) proposed a Wasserstein-based distance approach and showed its properties in the context of clustering techniques. For avoiding the disruptive effects of possible outliers, D'Urso et al. (D'URSO; GIOVANNI; MASSARI, 2015) suggested a fuzzy C-medoids method with a trimming rule. The clustering procedure is applied to the data after discarding a fixed fraction of outlying data. The percentage of data discarded in the clustering process and, thus, not considered in the optimization problem is determined by combining a validity criterion with the trimming algorithm. Later, Leski et al. (LESKI et al., 2016) introduced a clustering approach combining the fuzzy C-medoids clustering with the robust ordered statistics using Huber's M-estimator. Other clustering approaches proposed in the literature to manage symbolic data can be found in Refs. (CARVALHO, 2007; IRPINO; VERDE, 2008; CARVALHO; LECHEVALLIER, 2009; CARVALHO; LECHEVALLIER, 2009; LESKI et al., 2016; CARVALHO; SIMÕES, 2017; CHEN; BILLARD, 2019; SOUZA; SOUZA; AMARAL, 2020).

An interesting work is the one introduced by Ref. (D'URSO et al., 2017). They proposed the use of a robust metric based on the exponential distance in a framework of the fuzzy C-medoids clustering model for interval-valued data as follows:

$$J_{ExpFCMd-ID} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m \left[1 - \exp\{-\beta\left(\|\mathbf{m}_i - \overline{\mathbf{m}}_k\|^2 + \|\mathbf{r}_i - \overline{\mathbf{r}}_k\|^2\right)\}\right] \tag{4.1}$$

where $u_{ik} \in [0,1]$, $\sum_{k=1}^{C}(u_{ik}) = 1$ and the objects and prototypes are considered by the means of their midpoint and radius. The fuzziness of membership for each object is controlled by $m \in [1,\infty]$. Besides, β is a suitable parameter (positive constant) determined according to the variability of the data. The method starts with an initial partition and computes the prototypes and the fuzzy partition iteratively in two steps until a stopping criterion is satisfied as shown in Algorithm 6.

Algorithm 6 Proposed ExpFCMd-ID algorithm.

Input: The dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the number C of clusters $(2 \leq C \leq N)$, the parameters $m(1 < m < \infty)$, β and T (maximum number of iterations).

Output: The vector of prototypes $\mathbf{G}$; the matrix of membership degrees $\mathbf{U}$;

1: **Initialization**:

Set $t = 0$;

Pick initial medoids $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$ to obtain the vector of prototypes $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, ..., \mathbf{g}_C^{(t)})$, such that $\mathbf{g}_k^{(t)} = (\overline{\mathbf{m}}_k^{(t)}, \overline{\mathbf{r}}_k^{(t)})$;

2: **repeat**

Store the current medoids $\mathbf{G}_{old}^{(t)} = \mathbf{G}^{(t)}$;

3: **Step 1: assignment**:

Compute the components of $\mathbf{U}^{(t)}$ as follows:

$$u_{ik} = \frac{1}{\sum_{h=1}^{C}\left[\frac{1-\exp\{-\beta(\|\mathbf{m}_i-\overline{\mathbf{m}}_k\|^2+\|\mathbf{r}_i-\overline{\mathbf{r}}_k\|^2)\}}{1-\exp\{-\beta(\|\mathbf{m}_i-\overline{\mathbf{m}}_h\|^2+\|\mathbf{r}_i-\overline{\mathbf{r}}_h\|^2)\}}\right]^{\frac{1}{m-1}}};$$

4: **Step 2: representation**:

Select the new medoids $\mathbf{g}_k^{(t)} = (\overline{\mathbf{m}}_k^{(t)}, \overline{\mathbf{r}}_k^{(t)})$

for k=1 to C **do**

$$q = \underset{1\leq i\leq N}{\arg\max} \sum_{j=1}^{N}(u_{jk})^m \left[1 - \exp\{-\beta\left(\|\mathbf{m}_i - \mathbf{m}_j\|^2 + \|\mathbf{r}_i - \mathbf{r}_j\|^2\right)\}\right]$$

$$\mathbf{g}_k = \mathbf{x}_q$$

5: Set t=t+1;

6: **until** $\mathbf{G}_{old}^{(t)} = \mathbf{G}^{(t)}$ or $t > T$

Another method closely related to this work is the one proposed by Ref. (CARVALHO; SIMÕES, 2017), where a robust fuzzy interval-valued data clustering are introduced as shown in Equation (4.2).

$$J_{AIFCM} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m d_{\mathbf{V}}(\mathbf{x}_i, \mathbf{g}_k) \tag{4.2}$$

The parameter $m \in [1, +\infty]$ controls the fuzziness of membership for each object. The function $d_{\mathbf{V}}$ is a suitable adaptive variable-wise dissimilarity between the vectors of intervals $\mathbf{x}_i$ and $\mathbf{g}_k$, parameterized by the vectors of relevance weights of the variables $\mathbf{V}$. The weights are estimated globally for all clusters and locally for each cluster. Ref. (CARVALHO; SIMÕES, 2017)

considers adaptive dissimilarity measures based on the City-Block and Hausdorff distances as shown in Table 15.

Table 15 – Suitable variable-wise dissimilarity for the algorithm AIFCM based on the City-Block and Hausdorff distances.

Global	Local
$\sum_{j=1}^{P}(v_j)[\lvert a_{ij}-\alpha_{kj}\rvert+\lvert b_{ij}-\beta_{kj}\rvert]$	$\sum_{j=1}^{P}(v_{kj})[\lvert a_{ij}-\alpha_{kj}\rvert+\lvert b_{ij}-\beta_{kj}\rvert]$
$\sum_{j=1}^{P}(v_j)\max\{\lvert a_{ij}-v_{kj}\rvert,\lvert b_{ij}-\beta_{kj}\rvert\}$	$\sum_{j=1}^{P}(v_{kj})\max\{\lvert a_{ij}-\alpha_{kj}\rvert,\lvert b_{ij}-\beta_{kj}\rvert\}$

Source: Author (2022)

The algorithm returns the matrix of membership degrees, the vector of prototypes for each fuzzy cluster, and the vector of relevance weights of the variables. The minimization of the objective function is performed iteratively in three steps (representation, weighting, and assignment). In the first step, the matrix of prototypes $\mathbf{G}$ are computed following the algorithm proposed in Refs. (KARST, 1958; JAJUGA, 1991). The relevance weights of the variables are computed as follows.

Case 1: If the dissimilarity function is based on the City-Block and Hausdorff distances and globally takes into account the weighting of the variables such that $v_j > 0$ and $\prod_{j=1}^{P} v_j = 1$, then the vectors of relevance weights of the variables $\mathbf{V} = (v_1, \ldots, v_P)$ that minimize the adequacy criterion have their components computed as shown in Equation (4.3) and Equation (4.4).

$$v_j = \frac{\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m(\lvert a_{iw}-\alpha_{kw}\rvert+\lvert b_{iw}-\beta_{kw}\rvert)\right]\}^{\frac{1}{P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m(\lvert a_{ij}-\alpha_{kj}\rvert+\lvert b_{ij}-\beta_{kj}\rvert)} \tag{4.3}$$

$$v_j = \frac{\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m\max\{\lvert a_{iw}-\alpha_{kw}\rvert,\lvert b_{iw}-\beta_{kw}\rvert\}\right]\}^{\frac{1}{P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^m\max\{\lvert a_{ij}-\alpha_{kj}\rvert,\lvert b_{ij}-\beta_{kj}\rvert\}} \tag{4.4}$$

Case 2: If the dissimilarity function is based on the City-Block and Hausdorff distances and locally takes into account the weighting of the variables such that $v_{kj} > 0$ and $\prod_{j=1}^{P} v_{kj} = 1$, then the vector of relevance weights for the cluster k, $\mathbf{v}_k = (v_{k1}, \ldots, v_{kP})$ that minimize the adequacy criterion have their components computed as shown in Equation (4.5) and Equation (4.6).

$$v_{kj} = \frac{\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})^m(\lvert a_{iw}-\alpha_{kw}\rvert+\lvert b_{iw}-\beta_{kw}\rvert)\right]\}^{\frac{1}{P}}}{\sum_{i=1}^{N}(u_{ik})^m(\lvert a_{ij}-\alpha_{kj}\rvert+\lvert b_{ij}-\beta_{kj}\rvert)} \tag{4.5}$$

$$v_{kj} = \frac{\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})^m\max\{\lvert a_{iw}-\alpha_{kw}\rvert,\lvert b_{iw}-\beta_{kw}\rvert\}\right]\}^{\frac{1}{P}}}{\sum_{i=1}^{N}(u_{ik})^m\max\{\lvert a_{ij}-\alpha_{kj}\rvert,\lvert b_{ij}-\beta_{kj}\rvert\}} \tag{4.6}$$

The membership degrees u_{ik} are computed according to eq. (4.7). Table 16 specifies the assignment rules to obtain the fuzzy partition according to the different distance functions. Algorithm 7 summarizes the steps of the algorithm AIFCM.

$$u_{ik} = \left[\sum_{h=1}^{C} \left(\frac{d_{\mathbf{V}}(\mathbf{x}_i, \mathbf{g}_k)}{d_{\mathbf{V}}(\mathbf{x}_i, \mathbf{g}_w)} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{4.7}$$

Table 16 – Rules to compute the fuzzy partition for $AIFCM$ according to the distance functions.

Global adaptive distance	
City-Block	Hausdorff
$u_{ik} = \left[\sum_{h=1}^{C} \left(\frac{\sum_{j=1}^{P} (v_j)[\lvert a_{ij} - \alpha_{kj} \rvert + \lvert b_{ij} - \beta_{kj} \rvert]}{\sum_{j=1}^{P} (v_j)[\lvert a_{ij} - \alpha_{wj} \rvert + \lvert b_{ij} - \beta_{wj} \rvert]} \right)^{\frac{1}{m-1}} \right]^{-1}$	$u_{ik} = \left[\sum_{h=1}^{C} \left(\frac{\sum_{j=1}^{P} (v_j) \max\{\lvert a_{ij} - \alpha_{kj} \rvert, \lvert b_{ij} - \beta_{kj} \rvert\}}{\sum_{j=1}^{P} (v_j) \max\{\lvert a_{ij} - \alpha_{wj} \rvert, \lvert b_{ij} - \beta_{wj} \rvert\}} \right)^{\frac{1}{m-1}} \right]^{-1}$
Local adaptive distance	
$u_{ik} = \left[\sum_{h=1}^{C} \left(\frac{\sum_{j=1}^{P} (v_{kj})[\lvert a_{ij} - \alpha_{kj} \rvert + \lvert b_{ij} - \beta_{kj} \rvert]}{\sum_{j=1}^{P} (v_{kj})[\lvert a_{ij} - \alpha_{wj} \rvert + \lvert b_{ij} - \beta_{wj} \rvert]} \right)^{\frac{1}{m-1}} \right]^{-1}$	$u_{ik} = \left[\sum_{h=1}^{C} \left(\frac{\sum_{j=1}^{P} (v_{kj}) \max\{\lvert a_{ij} - \alpha_{kj} \rvert, \lvert b_{ij} - \beta_{kj} \rvert\}}{\sum_{j=1}^{P} (v_{kj}) \max\{\lvert a_{ij} - \alpha_{wj} \rvert, \lvert b_{ij} - \beta_{wj} \rvert\}} \right)^{\frac{1}{m-1}} \right]^{-1}$

Source: Author (2022)

Although the above methods deal with symbolic data and consider the relevance of variables, they assign the same importance for the lower and upper boundaries of the variables. As a result, a boundary that plays a minor role in the clustering task can still significantly impact. Addressing such, Ref. (SOUZA; CARVALHO, 2004) proposed a clustering method considering the relevance of lower and upper boundaries. However, this approach manages the interval bounds independently, which can be seen as single-valued data clustering. This is a disadvantage since lower and upper boundaries are intrinsically related and must be treated as such. In this regard, Ref. (IRPINO; VERDE; CARVALHO, 2017) proposed a fuzzy clustering algorithm for data described by distributional variables. The method uses the L_2 Wasserstein distance between distributions as a dissimilarity measure. Then, a decomposition of the distance and the notion of adaptive distance are then introduced for automatically computing relevance weights associated with variables and their components. We extend such approach for interval-valued data clustering. On the other hand, such FCM methods are sensitive to initial cluster centers and noises, so they tend to deteriorate in some cases, especially with incomplete data (TAO et al., 2019; SING; ADHIKARI; BASU, 2015).

4.3 PROPOSED INTERVAL-VALUED DATA CLUSTERING WITH ENTROPY REGULARIZATION

This section introduces new FWC and FWSC algorithms for interval-valued data. Additionally, an entropy term is added, functioning as a regulating factor during the minimization process. The proposed approaches are based on adaptive Euclidean and City-Block distances that consider the joint relevance of the variables for the lower and upper boundaries. As a result, a boundary that plays a significant role will have a higher relevance weight. Local and

Algorithm 7 AIFCM

Input: The dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the number C of clusters $(2 \leq C \leq N)$, the parameter $m > 1$, the parameter T (maximum number of iterations) and, the threshold $\varepsilon > 0$ and $\varepsilon << 1$.

Output: The vector of prototypes $\mathbf{G}$ and the matrix of membership degrees $\mathbf{U}$.

1: **Initialization**:

Set $t = 0$;

Randomly select C distinct prototypes $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$ to obtain the vector of prototypes $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, ..., \mathbf{g}_C^{(t)})$;

Initialize $\mathbf{V} = (v_j^{(t)})_{1 \leq j \leq P}$ with $v_j^{(t)} = 1$ or $\mathbf{V} = (v_{kj}^{(t)})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ with $v_{kj}^{(t)} = 1$;

Compute the matrix of membership degrees $\mathbf{U}^{(t)} = (u_{ik}^{(t)})_{\substack{1 \leq i \leq N \\ 1 \leq k \leq C}}$ according to Equation (4.7);

Compute the initial value of the clustering criterion J_{AIFCM} according to Equation (4.2);

2: **repeat**

Set $t = t + 1$; $J_{OLD} = J_{AIFCM}$;

3: **Step 1: representation**:

Compute the matrix of prototypes $\mathbf{G}$ as proposed in Refs. (KARST, 1958; JAJUGA, 1991);

4: **Weighting step**:

Compute $\mathbf{V}^{(t)}$ from Equations (4.3) to (4.6);

5: **Step 3: assignment**:

Compute the elements u_{ij} of the matrix of membership degrees $\mathbf{U} = (u_{ij})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq C}}$ according to Equation (4.7).

6: Compute J_{AIFCM} according to Equation (4.2) and set $J_{NEW} = J_{AIFCM}$.

7: **until** $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$

global adaptive distances were considered, which change with each algorithm iteration and may differ from one group to another.

Each method provides the best fuzzy partition of E into C fuzzy clusters, represented by the best matrix of membership degrees of the objects into the clusters $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ with $\mathbf{u}_i = (u_{i1}, \ldots, u_{iC})$, the best matrix of prototypes $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_C)$ for the groups and either the best vectors or matrices of weights of relevance of the variables $\mathbf{V}$ or the best vectors or matrices of weights of relevance of the variables $\mathbf{V}_l$ and $\mathbf{V}_u$ for the lower and upper boundaries, respectively.

All proposals consider the joint relevance of the variables, assuming different weights of importance for the lower and upper boundaries. The adequacy criterion can be defined as follows:

$$J_{EIFCM} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i,\mathbf{g}_k) + T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \tag{4.8}$$

such that $\sum_{k=1}^{C} u_{ik} = 1$. The dissimilarity between objects and prototypes is measured by $d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i,\mathbf{g}_k)$, which is parameterized by the vectors or matrices of relevance weights of the variables $\mathbf{V}_l$ and $\mathbf{V}_u$ for the lower and upper boundaries, respectively. Moreover, T_u is a weighting parameter that specifies the degree of fuzziness, such that $T_u > 0$. By increasing it, the fuzziness of the clusters increases. Note that this kind of approach, when applied to fuzzy clustering, provides a new perspective to face the problem of fuzzifying the clusterization of the objects while ensuring the maximum compactness of the obtained clusters.

Initially, a FWC is introduced. In this case, the method EIFCM is named Entropy Fuzzy Clustering Method with Global Joint relevance of the interval-valued variables and adaptive Euclidean distance (EIFCM-GJ2) when the dissimilarity function is based on the Euclidean distance. The objective function can be rewritten as follows:

$$\begin{aligned} J_{EIFCM-GJ2} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\sum_{j=1}^{P}\left[(v_{l,j})(a_{ij}-\alpha_{kj})^2 + (v_{u,j})(b_{ij}-\beta_{kj})^2\right] \\ +T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{4.9}$$

However, if the dissimilarity function is based on the City-Block distance, the method EIFCM is named as Entropy Fuzzy Clustering Method with Global Joint relevance of the interval-valued variables and adaptive City-Block distance (EIFCM-GJ1). The objective function for this FWC approach is defined as Equation (4.10).

$$\begin{aligned} J_{EIFCM-GJ1} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\sum_{j=1}^{P}\left[(v_{l,j})|a_{ij}-\alpha_{kj}| + (v_{u,j})|b_{ij}-\beta_{kj}|\right] \\ +T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{4.10}$$

In both cases, the vector of positive weights are estimated globally for all clusters at once. Then, such vector of positive weights for lower and upper boundaries are denoted as $\mathbf{V}_l = (v_{l,1},\ldots,v_{l,P})$ and $\mathbf{V}_u = (v_{u,1},\ldots,v_{u,P})$, respectively, with $v_{l,j} > 0$ and $v_{u,j} > 0$. Besides, the weights of the lower and upper boundaries are jointly restricted such that $\prod_{j=1}^{P}(v_{l,j})\times(v_{u,j}) = 1$.

The alternative FWSC is also considered. In this case, the adaptive distance considers the local joint weighing of the boundaries, in which the matrices of positive weights for the lower and upper boundaries are denoted as $\mathbf{V}_l = (\mathbf{v}_{l,1},\ldots,\mathbf{v}_{l,C})$ and $\mathbf{V}_u = (\mathbf{v}_{u,1},\ldots,\mathbf{v}_{u,C})$, respectively. The adequacy criterion of Equation (4.8) for the Entropy Fuzzy Clustering Method

with Local Joint relevance of the interval-valued variables and adaptive Euclidean distance (EIFCM-LJ2) is defined as follows if the dissimilarity function is based on the Euclidean distance:

$$J_{EIFCM-LJ2} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\sum_{j=1}^{P}\left[(v_{l,kj})(a_{ij}-\alpha_{kj})^2 + (v_{u,kj})(b_{ij}-\beta_{kj})^2\right] + T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \tag{4.11}$$

Nevertheless, when the dissimilarity function is based on the City-Block distance, the method EIFCM is named as Entropy Fuzzy Clustering Method with Local Joint relevance of the interval-valued variables and adaptive City-Block distance (EIFCM-LJ1) and defined as Equation (4.12).

$$J_{EIFCM-LJ1} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\sum_{j=1}^{P}\left[(v_{l,kj})|a_{ij}-\alpha_{kj}| + (v_{u,kj})|b_{ij}-\beta_{kj}|\right] + T_u\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik}) \tag{4.12}$$

In this case, the $\mathbf{v}_{l,k} = (v_{l,k1},\ldots,v_{l,kP})$ and the $\mathbf{v}_{u,k} = (v_{u,k1},\ldots,v_{u,kP})$ measure the importance of each interval-valued variable on the k-th cluster, where $v_{l,kj} > 0$, $v_{u,kj} > 0$ and $\prod_{j=1}^{P} v_{l,kj} \times v_{u,kj} = 1$.

4.3.1 Optimization steps

In general, proposed clustering algorithms set an initial fuzzy partition and alternate three steps until a satisfying stopping criterion. They provide the prototypes for each fuzzy cluster, either the relevance weight for each variable or the relevance weights of the lower and upper boundaries of the variables and the fuzzy partition.

4.3.1.1 Representation step

The representation step computes the prototypes associated with each fuzzy cluster. During this step, the relevance weights of the variables, and the fuzzy partition are kept fixed. Note that the computation of the prototypes depends on the dissimilarity function used.

Case 1: The dissimilarity functions are based on the Euclidean distance. In this case, the partial derivative of Equation (4.9) is taken with respect to the boundaries α_{kj} and β_{kj}. This is done to compute the components $g_{kj} = [\alpha_{kj}, \beta_{kj}]$ of the prototype $\mathbf{g}_k = (g_{k1},\ldots,g_{kP})$ of the cluster k, which minimizes the clustering criterion, as follows:

$$\frac{\partial J_{EIFCM-GJ2}}{\partial \alpha_{kj}} = \sum_{i=1}^{N}(u_{ik})(v_j)(-2a_{ij} + 2\alpha_{kj}) = 0 \tag{4.13}$$

$$\frac{\partial J_{EIFCM-GJ2}}{\partial \beta_{kj}} = \sum_{i=1}^{N}(u_{ik})(v_j)(-2b_{ij} + 2\beta_{kj}) = 0 \tag{4.14}$$

After setting the partial derivatives to zero and after some algebra we obtain:

$$\alpha_{kj} = \frac{\sum_{i=1}^{N} u_{ik} a_{ij}}{\sum_{i=1}^{N} u_{ik}} \quad ; \quad \beta_{kj} = \frac{\sum_{i=1}^{N} u_{ik} b_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{4.15}$$

Following a similar procedure, the prototype for Equation (4.11) is computed according to Equation (4.15).

Case 2: If the dissimilarity function is based on the City-Block distance for both global and local cases, the boundaries of the intervals $g_{kj} = [\alpha_{kj}, \beta_{kj}]$ yields two minimization problems:

$$\sum_{i=1}^{N}(u_{ik})|a_{ij} - \alpha_{kj}| \longrightarrow Min \quad (a) \quad ; \quad \sum_{i=1}^{N}(u_{ik})|b_{ij} - \beta_{kj}| \longrightarrow Min \quad (b) \tag{4.16}$$

These two problems involve the minimization of $\sum_{i=1}^{N}|y_i - az_i|$, where $z_i = u_{ik}$, $y_i = u_{ik}a_{ij}$ and $a = \alpha_{kj}$ in minimization problem 4.16 (a). However, $y_i = u_{ik}b_{ij}$ and $a = \beta_{kj}$ in minimization problem 4.16 (b). Since these problems do not have an algebraic solution, an algorithmic solution (JAJUGA, 1991) is used to solve them by applying Algorithm 1.

4.3.1.2 Weighting step

The weighting step provides the relevance weights of the variables for lower and upper boundaries, globally for all clusters or locally for each cluster. During this step, the prototypes and the fuzzy partition remain fixed.

Proposition 5. *The relevance weights of the variables for each boundary are computed according to the adaptive distance used.*

Case 1: If the dissimilarity function is based on the Euclidean distance and globally considers the joint weighting of the boundaries such that $v_{l,j} > 0$, $v_{u,j} > 0$ *and* $\prod_{j=1}^{P} v_{l,j} \times v_{u,j} = 1$, *then the vectors of relevance weights of the lower and upper boundaries of the variables* $\mathbf{V}_l = (v_{l,1}, \ldots, v_{l,P})$ *and* $\mathbf{V}_u = (v_{u,1}, \ldots, v_{u,P})$ *that minimize the adequacy criterion of Equation* (4.9)*, have their components computed, respectively, as follows:*

$$v_{l,j} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{iw} - \alpha_{kw})^2\right]\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{iw} - \beta_{kw})^2\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{ij} - \alpha_{kj})^2} \tag{4.17}$$

$$v_{u,j} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{iw}-\alpha_{kw})^2\right]\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{iw}-\beta_{kw})^2\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{ij}-\beta_{kj})^2} \quad (4.18)$$

Case 2: If the dissimilarity function is based on the City-Block distance and globally considers the joint weighting of the boundaries such that $v_{l,j} > 0$, $v_{u,j} > 0$ *and* $\prod_{j=1}^{P} v_{l,j} \times v_{u,j} = 1$, *then the vectors of relevance weights of the lower and upper boundaries of the variables* $\mathbf{V}_l = (v_{l,1}, \ldots, v_{l,P})$ *and* $\mathbf{V}_u = (v_{u,1}, \ldots, v_{u,P})$ *that minimize the adequacy criterion of Equation* (4.10), *have their components computed, respectively, as follows:*

$$v_{l,j} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|a_{iw}-\alpha_{kw}|\right]\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|b_{iw}-\beta_{kw}|\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|a_{ij}-\alpha_{kj}|} \quad (4.19)$$

$$v_{u,j} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|a_{iw}-\alpha_{kw}|\right]\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|b_{iw}-\beta_{kw}|\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})|b_{ij}-\beta_{kj}|} \quad (4.20)$$

Case 3: If the dissimilarity function is based on the Euclidean distance and locally considers the joint weighting of the boundaries such that $v_{l,kj} > 0$, $v_{u,kj} > 0$ *and* $\prod_{j=1}^{P} v_{l,kj} \times v_{u,kj} = 1$, *then the vectors of relevance weights for the cluster* k, $\mathbf{v}_{l,k} = (v_{l,k1}, \ldots, v_{l,kP})$ *and* $\mathbf{v}_{u,k} = (v_{u,k1}, \ldots, v_{u,kP})$ *that minimize the adequacy criterion of Equation* (4.11), *have their components computed as follows:*

$$v_{l,kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})(a_{iw}-\alpha_{kw})^2\right]\left[\sum_{i=1}^{N}(u_{ik})(b_{iw}-\beta_{kw})^2\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}(u_{ik})(a_{ij}-\alpha_{kj})^2} \quad (4.21)$$

$$v_{u,kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})(a_{iw}-\alpha_{kw})^2\right]\left[\sum_{i=1}^{N}(u_{ik})(b_{iw}-\beta_{kw})^2\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}(u_{ik})(b_{ij}-\beta_{kj})^2} \quad (4.22)$$

Case 4: If the dissimilarity function is based on the City-Block distance and locally considers the joint weighting of the boundaries such that $v_{l,kj} > 0$, $v_{u,kj} > 0$ *and* $\prod_{j=1}^{P} v_{l,kj} \times v_{u,kj} = 1$, *then the vectors of relevance weights for the cluster* k, $\mathbf{v}_{l,k} = (v_{l,k1}, \ldots, v_{l,kP})$ *and* $\mathbf{v}_{u,k} = (v_{u,k1}, \ldots, v_{u,kP})$ *that minimize the adequacy criterion of Equation* (4.12), *have their components computed as follows:*

$$v_{l,kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})|a_{iw}-\alpha_{kw}|\right]\left[\sum_{i=1}^{N}(u_{ik})|b_{iw}-\beta_{kw}|\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}(u_{ik})|a_{ij}-\alpha_{kj}|} \quad (4.23)$$

$$v_{u,kj} = \frac{\left\{\prod_{w=1}^{P}\left[\sum_{i=1}^{N}(u_{ik})|a_{iw}-\alpha_{kw}|\right]\left[\sum_{i=1}^{N}(u_{ik})|b_{iw}-\beta_{kw}|\right]\right\}^{\frac{1}{2P}}}{\sum_{i=1}^{N}(u_{ik})|b_{ij}-\beta_{kj}|} \quad (4.24)$$

Proof. In case the dissimilarity function takes into account the globally joint weighting of the variables for lower and upper boundaries by minimizing Equation (4.9), the relevance of the variables can be computed according to Equations (4.17) and (4.18).

To obtain the weight of the variables for each boundary, the Lagrange multiplier δ is applied under constraint $\prod_{j=1}^{P} v_{l,j} \times v_{u,j} = 1$ as follows:

$$\mathcal{L} = J_{EIFCM-GJ2} - \delta(\prod_{j=1}^{P} v_{l,j} \times v_{u,j} - 1) \tag{4.25}$$

If we take the partial derivative of $\mathcal{L}$ with respect to $v_{l,j}$ and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial v_{l,j}} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{ij} - \alpha_{kj})^2 - \frac{\delta}{v_{l,j}} = 0 \tag{4.26}$$

Then, solving Equation (4.26) concerning $v_{l,j}$, Equation (4.27) is obtained. Following a similar reasoning, Equation (4.28) is computed.

$$v_{l,j} = \frac{\delta}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{ij} - \alpha_{kj})^2} \tag{4.27}$$

$$v_{u,j} = \frac{\delta}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{ij} - \beta_{kj})^2} \tag{4.28}$$

If $\prod_{w=1}^{P} v_{l,j} \times v_{u,j} = 1$ then:

$$\prod_{w=1}^{P}\left[\frac{\delta}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{iw} - \alpha_{kw})^2}\right]\left[\frac{\delta}{\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{iw} - \beta_{kw})^2}\right] = 1 \tag{4.29}$$

Solving Equation (4.29) we obtain that:

$$\delta = \left\{\prod_{h=1}^{P}\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{iw} - \alpha_{kw})^2\right]\left[\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{iw} - \beta_{kw})^2\right]\right\}^{\frac{1}{2P}} \tag{4.30}$$

Substituting Equation (4.30) in Equations (4.27) and (4.28) we obtain the expressions Equations (4.17) and (4.18).

We can rewrite Equation (4.9) as $J_{EIFCM-GJ2} = \sum_{j=1}^{P} v_{l,j}J_{l,j} + \sum_{i=1}^{P} v_{u,j}J_{u,j}$, where $J_{l,j} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(a_{ij} - \alpha_{kj})^2$, $J_{u,j} = \sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})(b_{ij} - \beta_{kj})^2$ and $T_u \sum_{i=1}^{P}\sum_{k=1}^{C}(u_{ik})\ln(u_{ik})$ is seen like a constant. An extremum value of $J_{EIFCM-GJ2}$ is reached when $v_{l,j}$ and $v_{u,j}$ are computed according to Equations (4.17) and (4.18). So the extremum can be expressed as:

$$J_{EIFCM-GJ2} = \sum_{j=1}^{P} v_{l,j}J_{l,j} + \sum_{i=1}^{P} v_{u,j}J_{u,j} = 2P\left\{\prod_{j=1}^{P} J_{l,j}J_{u,j}\right\}^{\frac{1}{2P}} = 2\sqrt{P\left\{\prod_{j=1}^{P} J_{l,j}\right\}^{\frac{1}{P}}}\sqrt{P\left\{\prod_{j=1}^{P} J_{u,j}\right\}^{\frac{1}{P}}} \tag{4.31}$$

Because the arithmetic mean is greater than the geometric mean, $\frac{1}{P}\sum_{j=1}^{P} J_{l,j} > \{\prod_{j=1}^{P} J_{l,j}\}^{\frac{1}{P}}$ and $\frac{1}{P}\sum_{j=1}^{P} J_{u,j} > \{\prod_{j=1}^{P} J_{u,j}\}^{\frac{1}{P}}$. Also, if $a, b \in \mathbb{R}^{+}$ then $(\sqrt{a} - \sqrt{b})^2 \geq 0$, and therefore $a + b \geq 2\sqrt{a}\sqrt{b}$. Thus, as $J_{EIFCM-GJ2}(1, \ldots, 1; 1, \ldots, 1) = \sum_{j=1}^{P} J_{l,j} + \sum_{i=1}^{P} J_{u,j}$, the following inequality holds:

$$\sum_{j=1}^{P} J_{l,j} + \sum_{i=1}^{P} J_{u,j} \geq 2\sqrt{\sum_{j=1}^{P} J_{l,j}}\sqrt{\sum_{i=1}^{P} J_{u,j}} \geq 2\sqrt{P\left\{\prod_{j=1}^{P} J_{l,j}\right\}^{\frac{1}{P}}}\sqrt{P\left\{\prod_{j=1}^{P} J_{u,j}\right\}^{\frac{1}{P}}} \quad (4.32)$$

In conclusion, we can say that this extreme is a minimum. The other cases follow a similar reasoning. □

In local adaptive distances and for a fixed variable, the closer objects are to the prototype of a given cluster concerning a given boundary, the higher is the relevance weight of this boundary on the cluster. Moreover, for the global adaptive distances and for a fixed variable, the closer objects are to the set of cluster prototypes concerning a given boundary, the higher is the relevance weight of this boundary. Note that proposed constraints provide high weights only for the most relevant boundaries. This characteristic represents an advantage over previous works, which can produce bounds with high relevance weights even if they are not.

4.3.1.3 Assignment step

This step computes the matrix of membership degrees that represents the fuzzy partition, in which the prototypes and the weights of the variables for lower and upper boundaries are kept fixed.

Proposition 6. *The component of the matrix* $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$*, where* $\mathbf{u}_i = (u_{i1}, \ldots, u_{iC})$ *that minimizes the adequacy criterion of Equation* (4.8) *is computed according to the following expression:*

$$u_{ik} = \frac{\exp\left\{-\frac{d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i,\mathbf{g}_k)}{T_u}\right\}}{\sum_{w=1}^{C}\exp\left\{-\frac{d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i,\mathbf{g}_w)}{T_u}\right\}} \quad (4.33)$$

where $d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i, \mathbf{g}_k)$ *is the distance function that compares the* i*-th object and the prototype of the cluster* k*. Table 17 specifies the assignment rules for obtaining the fuzzy partition according to the different dissimilarity functions.*

Proof. We want to minimize the clustering criterion Equation (4.8) with respect to u_{ik} under $u_{ik} \in [0, 1]$ and $\sum_{k=1}^{C} u_{ik} = 1$. Let the Lagrangian function be:

$$\mathcal{L} = J_{EIFCM} - \sum_{i=1}^{N} \lambda_i \left[\sum_{k=1}^{C} u_{ik} - 1\right] \quad (4.34)$$

Table 17 – Expressions to obtain the fuzzy partition according to the Euclidean and City-Block distances.

Objective functions	Assignment rule
Equation (4.9)	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,j})(a_{ij}-\alpha_{kj})^2+(v_{u,j})(b_{ij}-\beta_{kj})^2\right]}{T_u}\right\}}{\sum_{h=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,j})(a_{ij}-\alpha_{hj})^2+(v_{u,j})(b_{ij}-\beta_{hj})^2\right]}{T_u}\right\}}$
Equation (4.10)	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,j})\lvert a_{ij}-\alpha_{kj}\rvert+(v_{u,j})\lvert b_{ij}-\beta_{kj}\rvert\right]}{T_u}\right\}}{\sum_{h=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,j})\lvert a_{ij}-\alpha_{hj}\rvert+(v_{u,j})\lvert b_{ij}-\beta_{hj}\rvert\right]}{T_u}\right\}}$
Equation (4.11)	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,kj})(a_{ij}-\alpha_{kj})^2+(v_{u,kj})(b_{ij}-\beta_{kj})^2\right]}{T_u}\right\}}{\sum_{h=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,hj})(a_{ij}-\alpha_{hj})^2+(v_{u,hj})(b_{ij}-\beta_{hj})^2\right]}{T_u}\right\}}$
Equation (4.12)	$u_{ik} = \frac{\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,kj})\lvert a_{ij}-\alpha_{kj}\rvert+(v_{u,kj})\lvert b_{ij}-\beta_{kj}\rvert\right]}{T_u}\right\}}{\sum_{h=1}^{C}\exp\left\{-\frac{\sum_{j=1}^{P}\left[(v_{l,hj})\lvert a_{ij}-\alpha_{hj}\rvert+(v_{u,hj})\lvert b_{ij}-\beta_{hj}\rvert\right]}{T_u}\right\}}$

Source: Author (2022)

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

$$\frac{\partial \mathcal{L}}{\partial u_{ik}} = d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i, \mathbf{g}_k) + T_u(\ln(u_{ik}) + 1) - \lambda_i = 0 \tag{4.35}$$

From Equation (4.35), Equation (4.36) is obtained.

$$u_{ik} = \exp\{\frac{\lambda_i}{T_u} - 1\}\exp\{-\frac{d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i, \mathbf{g}_k)}{T_u}\} \tag{4.36}$$

If $\sum_{w=1}^{C} u_{ih} = 1$ then:

$$\sum_{w=1}^{C}\exp\{\frac{\lambda_i}{T_u} - 1\}\exp\{-\frac{d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i, \mathbf{g}_w)}{T_u}\} = 1 \tag{4.37}$$

From Equation (4.37) we have that:

$$\exp\{\frac{\lambda_i}{T_u} - 1\} = \frac{1}{\sum_{w=1}^{C}\exp\{-\frac{d_{(\mathbf{v}_l,\mathbf{v}_u)}(\mathbf{x}_i, \mathbf{g}_w)}{T_u}\}} \tag{4.38}$$

Substituting Equation (4.38) in Equation (4.36) we have Equation (4.33).

Additionally, we know that:

$$\frac{\partial J_{EIFCM}}{\partial u_{ik}} = d_{(\mathbf{v}_l,\mathbf{v}_u}(\mathbf{x}_i, \mathbf{g}_k) + T_u(\ln(u_{ik}) + 1) \tag{4.39}$$

Therefore, $\frac{\partial^2 J_{EIFCM}}{\partial u_{ik}} = \frac{T_u}{u_{ik}}$. Then, the Hessian matrix of J_{EIFCM} according to $\mathbf{U}$ is:

$$\partial^2 J_{EIFCM}(\mathbf{U}) = \begin{bmatrix} \frac{T_u}{u_{11}} & \dots & 0 \\ & \dots\dots & \\ 0 & \dots & \frac{T_u}{u_{NC}} \end{bmatrix}$$

Since $T_u > 0$ and $u_{ik} > 0$, the Hessian matrix $\partial^2 J_{EIFCM}(\mathbf{U})$ is positive definite, so that we can conclude that this extremum is a minimum. □

The proposed maximum entropy clustering algorithms share some similarities with the Gaussian method proposed by Rui-Ping Li et al. (LI; MUKAIDONO, 1995) regarding the membership degree computing. Please see Section 2.5. Then, we can say that the proposals in this work have a distinct physical meaning and well-defined mathematical characteristics, making them easy to understand (TAO et al., 2019; LI; MUKAIDONO, 1995).

4.3.2 Proposed algorithms

Algorithm 8 summarizes the proposed fuzzy clustering approaches. To simplify the presentation and discussion of the experimental results, hereafter, it is adopted the following notations for the proposed methods.

- EIFCM-GJ2 when the dissimilarity function is based on the Euclidean distance and globally takes into account the joint weighting of the lower and the upper boundaries of the variables Equation (4.9).
- EIFCM-GJ1 when the dissimilarity function is based on the City-Block distance and globally takes into account the joint weighting of the lower and the upper boundaries of the variables Equation (4.10).
- EIFCM-LJ2 when the dissimilarity function is based on the Euclidean distance and locally takes into account the joint weighting of the lower and the upper boundaries of the variables Equation (4.11).
- EIFCM-LJ1 when the dissimilarity function is based on the City-Block distance and locally takes into account the joint weighting of the lower and the upper boundaries of the variables Equation (4.12).

Algorithm 8 Proposed algorithms.

Input: The dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the number C of groups, $T_u > 0$, T and $\varepsilon > 0$.

Output: The vector of prototypes $\mathbf{G}$; the matrix of membership degrees $\mathbf{U}$; the vector or the matrix of relevance weights $\mathbf{V}_l$ and $\mathbf{V}_u$.

1: **Initialization**:

Set $t = 0$;

Randomly select C distinct prototypes $\mathbf{g}_k^{(t)} \in D\,(k = 1, ..., C)$;

Initialize $\mathbf{V}_l = (v_{l,j}^{(t)})_{1 \leq j \leq P}$ with $v_{l,j}^{(t)} = 1$ and $\mathbf{V}_u = (v_{u,j}^{(t)})_{1 \leq j \leq P}$ with $v_{u,j}^{(t)} = 1$ or $\mathbf{V}_l = (v_{l,kj}^{(t)})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ with $v_{l,kj}^{(t)} = 1$ and $\mathbf{V}_u = (v_{u,kj}^{(t)})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ with $v_{u,kj}^{(t)} = 1$, $\forall k, j$;

Compute the membership degrees $\mathbf{U}^{(t)} = (u_{ik}^{(t)})_{\substack{1 \leq i \leq N \\ 1 \leq k \leq C}}$ by Equation (4.33);

Compute J according to the choice of adequacy criterion (Equations (4.9) to (4.12));

2: **repeat**

Set $t = t + 1$; $J_{OLD} = J$;

3: **Step 1: representation**:

Compute g_{kj} as shown in Section 4.3.1.1;

4: **Step 2: weighting**:

Compute $\mathbf{V}_l^{(t)}$ and $\mathbf{V}_u^{(t)}$ from as shown in Section 4.3.1.2;

5: **Step 3: assignment**:

Compute the components of $\mathbf{U}^{(t)}$ according to Equation (4.33);

6: Compute J according to the choice of adequacy criterion (acc. to Equations (4.9) to (4.12)) and set $J_{NEW} = J$.

7: **until** $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$

4.3.3 Convergence properties

The proposals provide a matrix of prototypes for each fuzzy cluster $\mathbf{G}$, a fuzzy partition $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ of E, and the best weights of the variables $\mathbf{V}_l$ and $\mathbf{V}_u$ for the lower and upper boundaries, respectively, such that:

$$J_{EIFCM-GJ2}(\mathbf{G}^{(*)}, \mathbf{V}_l^{(*)}, \mathbf{V}_u^{(*)}, \mathbf{U}^{(*)}) = \min\{J_{EIFCM-GJ2}(\mathbf{G}, \mathbf{V}_l, \mathbf{V}_u, \mathbf{U})\} : \mathbf{G} \in \mathbb{L}^C, (\mathbf{V}_l, \mathbf{V}_u) \in \Xi, \mathbf{U} \in \mathbb{U}^N$$

$$J_{EIFCM-GJ1}(\mathbf{G}^{(*)}, \mathbf{V}_l^{(*)}, \mathbf{V}_u^{(*)}, \mathbf{U}^{(*)}) = \min\{J_{EIFCM-GJ1}(\mathbf{G}, \mathbf{V}_l, \mathbf{V}_u, \mathbf{U})\} : \mathbf{G} \in \mathbb{L}^C, (\mathbf{V}_l, \mathbf{V}_u) \in \Xi, \mathbf{U} \in \mathbb{U}^N$$

$$J_{EIFCM-LJ2}(\mathbf{G}^{(*)}, \mathbf{V}_l^{(*)}, \mathbf{V}_u^{(*)}, \mathbf{U}^{(*)}) = \min\{J_{EIFCM-LJ2}(\mathbf{G}, \mathbf{V}_l, \mathbf{V}_u, \mathbf{U})\} : \mathbf{G} \in \mathbb{L}^C, (\mathbf{V}_l, \mathbf{V}_u) \in \Xi^C, \mathbf{U} \in \mathbb{U}^N$$

$$J_{EIFCM-LJ1}(\mathbf{G}^{(*)}, \mathbf{V}_l^{(*)}, \mathbf{V}_u^{(*)}, \mathbf{U}^{(*)}) = \min\{J_{EIFCM-LJ1}(\mathbf{G}, \mathbf{V}_l, \mathbf{V}_u, \mathbf{U})\} : \mathbf{G} \in \mathbb{L}^C, (\mathbf{V}_l, \mathbf{V}_u) \in \Xi^C, \mathbf{U} \in \mathbb{U}^N$$

where

- $\mathbb{L}$ is the representation space of the prototypes such that $\mathbf{g_k} \in \mathbb{L}(k = 1, \ldots, C)$ and $\mathbf{G} \in \mathbb{L}^C = \mathbb{L} \times \cdots \times \mathbb{L}$. In this work, $\mathbb{L} = \Im^P = \Im \times \cdots \times \Im$ and $\mathbf{G} \in (\Im^P)^C = \Im^P \times \cdots \times \Im^P$, where $\Im$ defines an interval;

- For global methods, Ξ is the space of vectors of weights, such that $(\mathbf{V}_l, \mathbf{V}_u) \in \Xi$ with $\Xi = \{(\mathbf{V}_l, \mathbf{V}_u) : \mathbf{V}_l = (v_{l,1}, \ldots, v_{l,P}) \in \mathbb{R}^P, v_{l,j} > 0, \mathbf{V}_u = (v_{l,u}, \ldots, v_{u,P}) \in \mathbb{R}^P, v_{u,j} > 0$ and $\prod_{j=1}^{P}(v_{l,j}) \times (v_{u,j}) = 1\}$;

- For local approaches, Ξ is the space of vectors of weights, such that $(\mathbf{v}_{l,k}, \mathbf{v}_{u,k}) \in \Xi(k = 1, \ldots, C)$ with $\Xi = \{(\mathbf{v}_{l,k}, \mathbf{v}_{u,k}) : \mathbf{v}_{l,k} = (v_{l,k1}, \ldots, v_{l,kP}) \in \mathbb{R}^P, \mathbf{v}_{u,k} = (v_{u,k1}, \ldots, v_{u,kP}) \in \mathbb{R}^P, v_{l,kj} > 0, v_{u,kj} > 0$ and $\prod_{j=1}^{P}(v_{l,kj}) \times (v_{u,kj}) = 1\}$ and $(\mathbf{V}_l, \mathbf{V}_u) \in \Xi^C = \Xi \times \cdots \times \Xi$;

- $\mathbb{U}$ is the space of fuzzy partition membership such that $\mathbf{u}_i \in \mathbb{U}(i = 1, \ldots, N)$, and $\mathbb{U} = \{\mathbf{u} = (u_1, \ldots, u_C) \in [0,1] \times \cdots \times [0,1] = [0,1]^C : \sum_{k=1}^{C} u_k = 1\}$ and $\mathbf{U} \in \mathbb{U}^N = \mathbb{U} \times \cdots \times \mathbb{U}$.

Similarly to Ref. (DIDAY; SIMON, 1976), the convergence properties of the proposed algorithms can be studied from the series:

- $\nu_{EIFCM-GJ2}^{(t)} = (\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi \times \mathbb{U}^N$ and $u_{EIFCM-GJ2}^{(t)} = J_{EIFCM-GJ2}(\nu_{EIFCM-GJ2}^{(t)}) = J_{EIFCM-GJ2}(\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

- $\nu_{EIFCM-GJ1}^{(t)} = (\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi \times \mathbb{U}^N$ and $u_{EIFCM-GJ1}^{(t)} = J_{EIFCM-GJ1}(\nu_{EIFCM-GJ1}^{(t)}) = J_{EIFCM-GJ1}(\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

- $\nu_{EIFCM-LJ2}^{(t)} = (\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi^C \times \mathbb{U}^N$ and $u_{EIFCM-LJ2}^{(t)} = J_{EIFCM-LJ2}(\nu_{EIFCM-LJ2}^{(t)}) = J_{EIFCM-LJ2}(\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

- $\nu_{EIFCM-LJ1}^{(t)} = (\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^C \times \Xi^C \times \mathbb{U}^N$ and $u_{EIFCM-LJ1}^{(t)} = J_{EIFCM-LJ1}(\nu_{EIFCM-LJ1}^{(t)}) = J_{EIFCM-LJ1}(\mathbf{G}^{(t)}, \mathbf{V}_l^{(t)}, \mathbf{V}_u^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number;

From an initial term

- $\nu_{EIFCM-GJ2}^{(0)} = (\mathbf{G}^{(0)}, \mathbf{V}_l^{(0)}, \mathbf{V}_u^{(0)}, \mathbf{U}^{(0)})$;

- $\nu^{(0)}_{EIFCM-GJ1} = (\mathbf{G}^{(0)}, \mathbf{V}^{(0)}_l, \mathbf{V}^{(0)}_u, \mathbf{U}^{(0)})$;
- $\nu^{(0)}_{EIFCM-LJ2} = (\mathbf{G}^{(0)}, \mathbf{V}^{(0)}_l, \mathbf{V}^{(0)}_u, \mathbf{U}^{(0)})$;
- $\nu^{(0)}_{EIFCM-LJ1} = (\mathbf{G}^{(0)}, \mathbf{V}^{(0)}_l, \mathbf{V}^{(0)}_u, \mathbf{U}^{(0)})$

the algorithms EIFCM-GJ2, EIFCM-GJ1, EIFCM-LJ2 and EIFCM-LJ1 compute the different terms of the series $\nu^{(t)}_{EIFCM-GJ2}$, $\nu^{(t)}_{EIFCM-GJ1}$, $\nu^{(t)}_{EIFCM-LJ2}$ and $\nu^{(t)}_{EIFCM-LJ1}$, until the respective convergence, when the objective functions $J_{EIFCM-GJ2}$, $J_{EIFCM-GJ1}$, $J_{EIFCM-LJ2}$ and $J_{EIFCM-LJ1}$ reach stationary values.

Proposition 7.

i. The series $u^{(t)}_{EIFCM-GJ2} = J_{EIFCM-GJ2}(\nu^{(t)}_{EIFCM-GJ2}) = J_{EIFCM-GJ2}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *decreases at each iteration and converges;*

ii. The series $u^{(t)}_{EIFCM-GJ1} = J_{EIFCM-GJ1}(\nu^{(t)}_{EIFCM-GJ1}) = J_{EIFCM-GJ1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *decreases at each iteration and converges;*

iii. The series $u^{(t)}_{EIFCM-LJ2} = J_{EIFCM-LJ2}(\nu^{(t)}_{EIFCM-LJ2}) = J_{EIFCM-LJ2}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *decreases at each iteration and converges;*

iv. The series $u^{(t)}_{EIFCM-LJ1} = J_{EIFCM-LJ1}(\nu^{(t)}_{EIFCM-LJ1}) = J_{EIFCM-LJ1}(\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *decreases at each iteration and converges;*

Proof. The proof follows a similar reasoning as presented in Section 3.3.3. □

Proposition 8.

i. The series $\nu^{(t)}_{EIFCM-GJ2} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *converges;*

ii. The series $\nu^{(t)}_{EIFCM-GJ1} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *converges;*

iii. The series $\nu^{(t)}_{EIFCM-LJ2} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *converges;*

iv. The series $\nu^{(t)}_{EIFCM-LJ1} = (\mathbf{G}^{(t)}, \mathbf{V}^{(t)}_l, \mathbf{V}^{(t)}_u, \mathbf{U}^{(t)})$ *converges;*

Proof. The convergence proof is similar to the one presented in Section 3.3.3. □

4.3.4 Complexity analysis

The computational complexity of the proposed approaches depends on the dissimilarity measure used.

– If the algorithms are based on the Euclidean distance and entropy regularization, the computational complexity is $O(N \times P \times C \times T)$.

– If the algorithms are based on the City-Block distance and entropy regularization, the computational complexity is $O(N \times P \times C \times T \times \log(N))$.

An advantage of the proposed approaches with entropy regularization is that they have a simple implementation and less computational complexity compared to FCM methods, which makes them more applicable for large and high-dimensional data clustering.

4.4 EXPERIMENTAL RESULTS

This section presents the properties and usefulness of proposed algorithms through different experiments on synthetic and real interval-valued datasets. A summary of proposed methods is shown in Table 18. The performance of the proposed approaches was compared with six clustering algorithms: TrFCMdd-ID (D'URSO; GIOVANNI; MASSARI, 2015), ExpFCMd-ID (D'URSO et al., 2017), AIFCM-G2 (CARVALHO; LECHEVALLIER, 2009), AIFCM-G1 (CARVALHO; SIMÕES, 2017), AIFCM-L2 (CARVALHO; LECHEVALLIER, 2009), and AIFCM-L1 (CARVALHO; SIMÕES, 2017). Furthermore, the quality of the fuzzy and hard partitions was measured by HUL and ARI. Also, research questions 4, 5 and 6 are answered.

Table 18 – Summary of proposed methods.

Algorithms	Description
EIFCM-GJ2	Proposed entropy fuzzy clustering method with global joint relevance of the interval-valued variables and adaptive Euclidean distance
EIFCM-GJ1	Proposed entropy fuzzy clustering method with global joint relevance of the interval-valued variables and adaptive City-Block distance
EIFCM-LJ2	Proposed entropy fuzzy clustering method with local joint relevance of the interval-valued variables and adaptive Euclidean distance
EIFCM-LJ1	Proposed entropy fuzzy clustering method with local joint relevance of the interval-valued variables and adaptive City-Block distance

Source: Author (2022)

4.4.1 Experimental setting

In all experiments, the parameters ε was set to 10^{-5}, the maximum number of iterations T to 100, and for each dataset, the number C of clusters was set equal to the number of a priori classes. The interval-valued datasets were normalized (CARVALHO; BRITO; BOCK, 2006), such that the resulting transformed midpoints have zero mean and dispersion 1 in each dimension.

The optimal values for m and Tu were found using a grid search strategy. Following a procedure similar to Ref. (COPPI; D'URSO, 2006), PC and PE were considered for measuring

the degree of overlap within the clusters. The parameter m was varied between 1.1 to 300 with step 0.1, and T_u between 0.1 to 1000 with step 0.1. Then, an optimal value for the parameter is obtained when $PC \approx PE$. This approach allows us to maximize both the separation of clusters measured by PC and the flexibility of classification given by PE. Note that optimal values are computed in an unsupervised manner. As for algorithm ExpFCMd-ID, the parameter β was estimated following Ref. (D'URSO et al., 2017). Since this method depends on both m and β, finding the optimal combination of values takes longer than for the other algorithms.

4.4.2 Synthetic interval-valued datasets in which lower and upper boundaries have different relevance

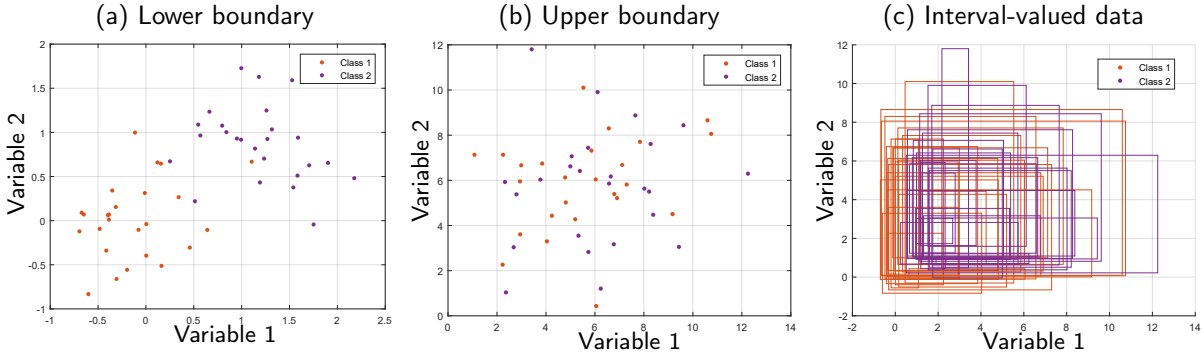
Some aspects of the proposed approaches are presented in this section through experiments on the synthetic interval-valued datasets. Initially, a dataset was created such that the lower and upper boundaries of the variables have different relevance for the clustering task. The objective is to show how previous and proposed approaches identify the structure of clusters under these conditions. The dataset has 25 objects for each of the two classes drawn from 2-dimensional Gaussian distributions with a specific mean vector and covariance matrix, as shown in Table 19. In this case, only the lower boundary of the variables is relevant for constructing the clusters. Figure 21 shows the obtained interval-valued dataset.

Table 19 – Parameter setting for lower and upper boundaries.

	Lower bound configuration				Upper bound configuration			
	μ_1	μ_2	σ_1^2	σ_2^2	μ_1	μ_2	σ_1^2	σ_2^2
Class 1	0	0	0.2	0.2	5	5	7	7
Class 2	1	1	0.2	0.2	6	6	7	7

Source: Author (2022)

Figure 21 – Plots for the interval-valued data lower and upper boundaries (a) and (b), respectively. (c) The synthetic interval-valued data.



Source: Author (2022)

All algorithms were executed in a Monte Carlo simulation framework with 20 replications. Then, 20 synthetic datasets were generated according to the settings described above. We run the algorithms 50 times for each dataset and select the best solution according to the adequacy criterion. The mean and standard deviation of HUL and ARI were computed based on the 20 Monte Carlo iterations.The robustness of the algorithms was compared according to the mean and standard deviation of the 50 runs in each dataset. Finally, the Friedman test is utilized to explore the statistical significance of the results obtained. Besides, the Nemenyi post-hoc test is used to determine which algorithms perform statistically differently. The objective is to determine whether at least one method is significantly better than at least one other method at the $\alpha = 0.05$ level.

4.4.2.1 Results

Table 20 shows the mean and standard deviation of the performance of each algorithm when the best solution is selected according to its adequacy criterion. In this case, proposed methods that consider the joint weights of the relevance of the intervals boundaries outperform other approaches. This is because the weighting of the variables is calculated according to the significance of each boundary. For a better understanding, Table 21 shows the importance of the variables for the proposed algorithms EIFCM-LJ1 and EIFCM-GJ2, which presented the best results for HUL and ARI, respectively. We can observe that they assign the highest relevance weights for the lower boundaries of variables 1 and 2. Besides, the upper boundaries of these variables were less relevant. Subsequently, the dissimilarity between objects and prototypes is amplified on the lower boundaries of the variables and reduced on the upper ones. Note that the literature methods showed a worse performance because they implicitly provide the same importance for both boundaries. Then, they cannot find the cluster structure when it appears in one of the boundaries. Nevertheless, among them, those based on the City-Block distance presented higher flexibility and better performance.

Figure 22 shows that, as expected, the proposed methods that consider the relevance of the variables for each boundary performed better than the other algorithms and the observed performance differences between these methods were statistically significant. The proposed method EIFCM-LJ2 don't show significant difference with the literature methods for HUL and $\alpha = 0.05$. Besides, among literature methods, those with City-Block distance achieved higher values as average. However, there is no consistent evidence to indicate statistical performance differences between them. In conclusion, separate weighting for the lower and upper boundaries is the most appropriate approach for problems in which the cluster structure is defined in one of the subspaces of the bounds of the variables.

Table 20 – Mean and standard deviation according to HUL and ARI

	HUL		ARI	
Algorithms	mean	(std)	mean	(std)
TrFCMdd-ID	0.5058	(0.0152)	0.0595	(0.0913)
ExpFCMd-ID	0.4999	(0.0070)	0.0327	(0.0598)
AIFCM-G2	0.5038	(0.0109)	0.0437	(0.0665)
AIFCM-G1	0.5121	(0.0148)	0.0915	(0.1145)
AIFCM-L2	0.5038	(0.0117)	0.0423	(0.0650)
AIFCM-L1	0.5115	(0.0146)	0.0879	(0.1030)
EIFCM-GJ2	0.6133	(0.0145)	**0.7100**	(0.1034)
EIFCM-GJ1	0.6163	(0.0339)	0.6508	(0.1679)
EIFCM-LJ2	0.6006	(0.0226)	0.6620	(0.1400)
EIFCM-LJ1	**0.6218**	(0.0268)	0.6919	(0.1562)

Source: Author (2022)

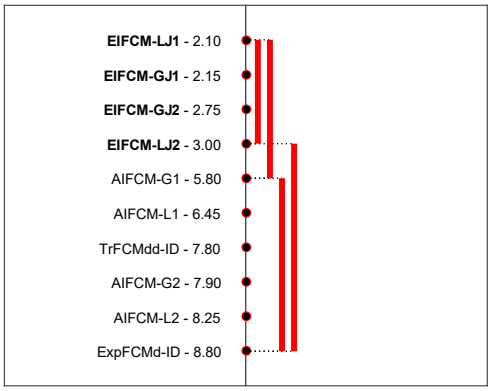
Table 21 – Weights of the variables obtained by the proposed algorithms EIFCM-GJ2 and EIFCM-LJ1 for the first Monte Carlo replica.

	EIFCM-GJ2		EIFCM-LJ1			
	$\mathbf{V}_l$	$\mathbf{V}_u$	$\mathbf{V}_l$		$\mathbf{V}_u$	
			Cluster 1	Cluster 2	Cluster 1	Cluster 2
Variable 1	3.7409	0.2295	1.7273	1.7057	0.4806	0.5519
Variable 2	5.1118	0.2279	2.0791	2.1595	0.5794	0.4919

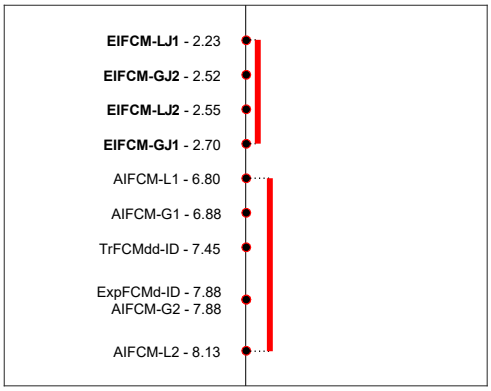
Source: Author (2022)

Figure 22 – Comparison of the algorithms with each other with the Nemenyi test on data where lower and upper bound variables have different relevance to clustering.

(a) Average performance ranking according to HUL



(b) Average performance ranking according to ARI



Source: Author (2022)

4.4.3 Experiments on synthetic interval-valued datasets with outliers

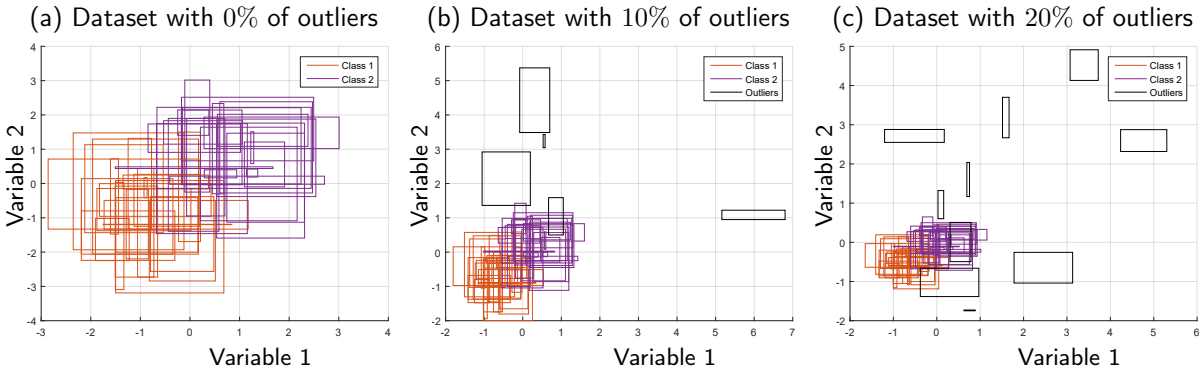
Since clustering methods can be sensitive to outliers, this experiment was considered to verify the robustness of the proposed algorithms. Addressing such, two-dimensional datasets

were generated (Figure 23 (a)) according to the following configuration:

- Class 1: Centers $U[0, 1]$ and Spreads $U[0, 1]$;
- Class 2: Centers $U[1, 2]$ and Spreads $U[0, 1]$;
- Outliers: Centers $N(4, 3)$ and Spreads $U[0, 1]$.

in which, $U[a, b]$ represents the uniform distribution with parameters a and b. Also, the $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2. Each a priori class has 25 objects, and we add 10% and 20% of outliers as shown in Figure 23.

Figure 23 – Plots of the interval-valued data with (a) 0%, (b) 10% and (c) 20% of outliers.



Source: Author (2022)

All algorithms were executed in a Monte Carlo simulation framework with 20 replications. Then, 20 synthetic datasets were generated according to the settings described above. We run the algorithms 50 times for each dataset and select the best solution of the 50 executions according to their adequacy criterion. The mean and standard deviation for the indices were computed based on the Monte Carlo replications. In addition, the statistical significance of the results obtained was explored.

4.4.3.1 Results

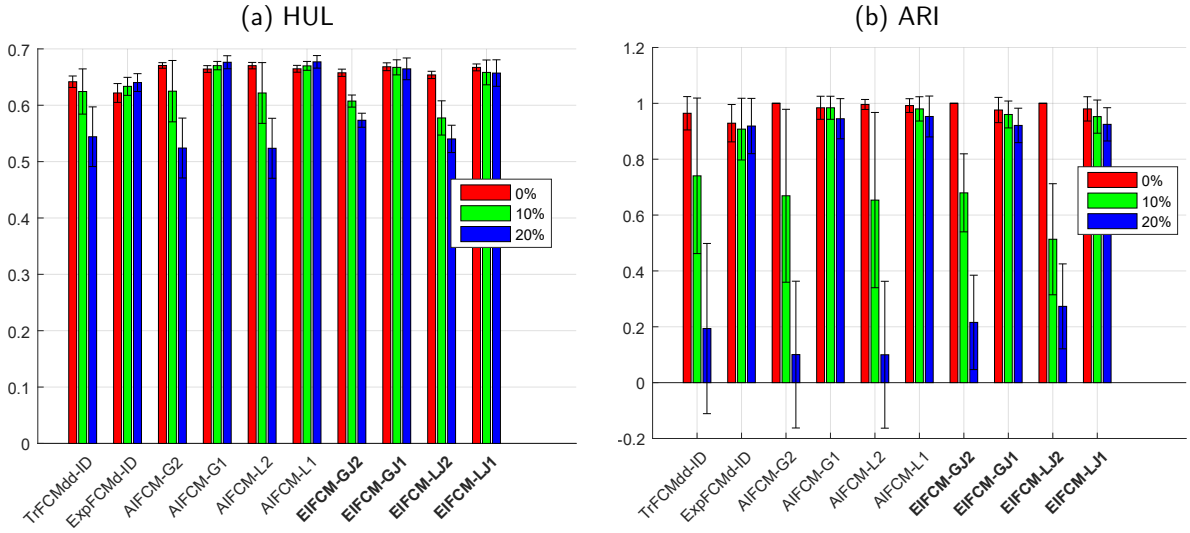
Figure 24 shows the mean and standard deviation according to the indices HUL and ARI computed from the 20 Monte Carlo replications, varying the number of outliers. Numeric values can be observed in Tables 30 and 31 (see Appendix C). Besides, Figures 25 and 26 presents the comparison of the algorithms according to the Nemenyi test. Models joined by horizontal lines do not show evidence of statistically significant differences.

When comparing the fuzzy partition $\mathbf{U}$ obtained by the algorithms and the a priori partition, we can observe that for 0% of outliers, the algorithms showed comparable results. However, when the number of outliers increased, the algorithm ExpFCMd-ID and the City-Block distance-based approaches presented the best results, showing higher robustness. After

them, the proposed methods with adaptive Euclidean distance and TrFCMdd-ID achieved the best performance. Additionally, they obtained similar fuzziness degrees for every iteration within the Monte Carlo framework, which led to low standard deviation.

On the other hand, methods based on the Euclidean distance outperform other approaches for 0% of outliers when comparing by ARI the hard partition and the a priori partition. Nevertheless, as expected, ExpFCMd-ID and City-Block distance-based methods presented the best performances when increasing the number of outliers in the datasets. They proved to be statistically more robust to outliers. Hence, the robustness of the proposed methods based on the City-Block distance was demonstrated. Besides, proposed methods based on Euclidean distance showed better clustering results than TrFCMdd-ID, AIFCM-G2, and AIFCM-L2 when the number of outliers increases. The latter had a higher standard deviation, indicating greater sensitivity to initial cluster centers and outliers.

Figure 24 – Mean and standard deviation for different percentages of outliers according to the indices HUL and ARI, respectively.

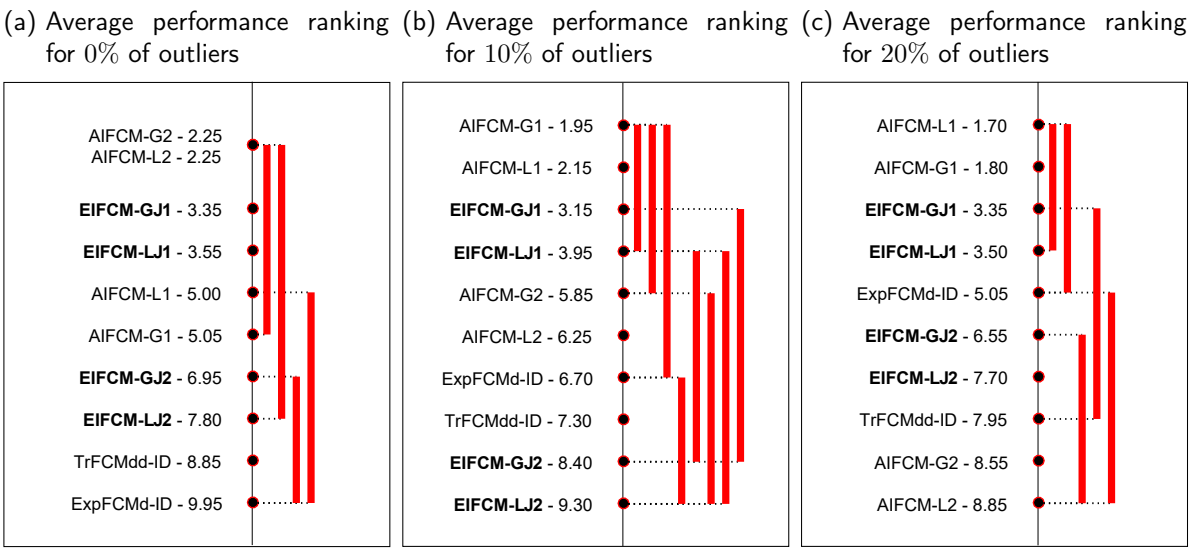


Source: Author (2022)

4.4.4 Experiments on real interval-valued datasets

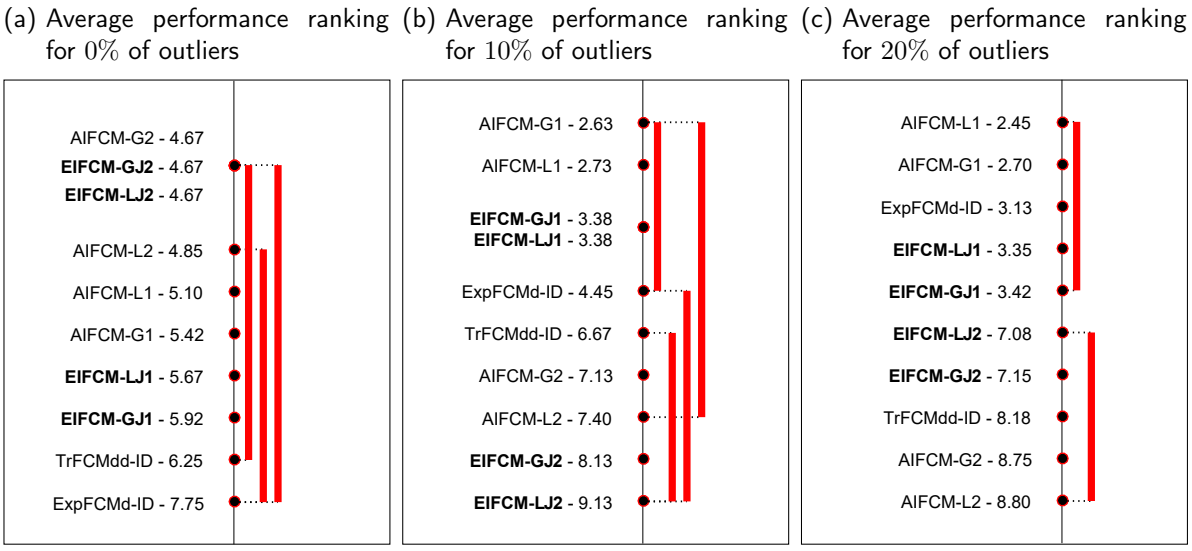
The clustering algorithms are compared on eight real interval-valued datasets: Car models (SILVA; BRITO, 2020), City temperature (GURU; KIRANAGI; NAGABHUSHAN, 2004), Freshwater fish species (BOUDOU; RIBEYRE, 1997), Fungi (WOOD; STEVENS, 2015), Horse, Ichino (ICHINO; YAGUCHI, 1994), Iris (LYNNE; EDWIN, 2006) and Wine (KALLITHRAKA et al., 2001). Table 22 briefly describes the datasets, in which the number of objects, interval variables and a priori clusters are represented by N, P and C, respectively. The algorithms were run 50 times for each dataset, and it is selected the best result according to the minimum value of their objective function.

Figure 25 – Comparison of the algorithms with each other with the Nemenyi test for HUL with different percentages of outliers.



Source: Author (2022)

Figure 26 – Comparison of the algorithms with each other with the Nemenyi test for ARI with different percentages of outliers.



Source: Author (2022)

4.4.4.1 Results

Table 32 presents the clustering results for the best execution according to their objective function. The performance rank is shown in parenthesis. Figure 27 exhibits the cumulative rank on each dataset. We can observe that when comparing the obtained fuzzy partition and the a priori partition according to HUL, the proposed approach EIFCM-LJ2 presented the highest values for the datasets Car models and City temperature. For Freshwater fish

Table 22 – Summary of the real interval-valued datasets.

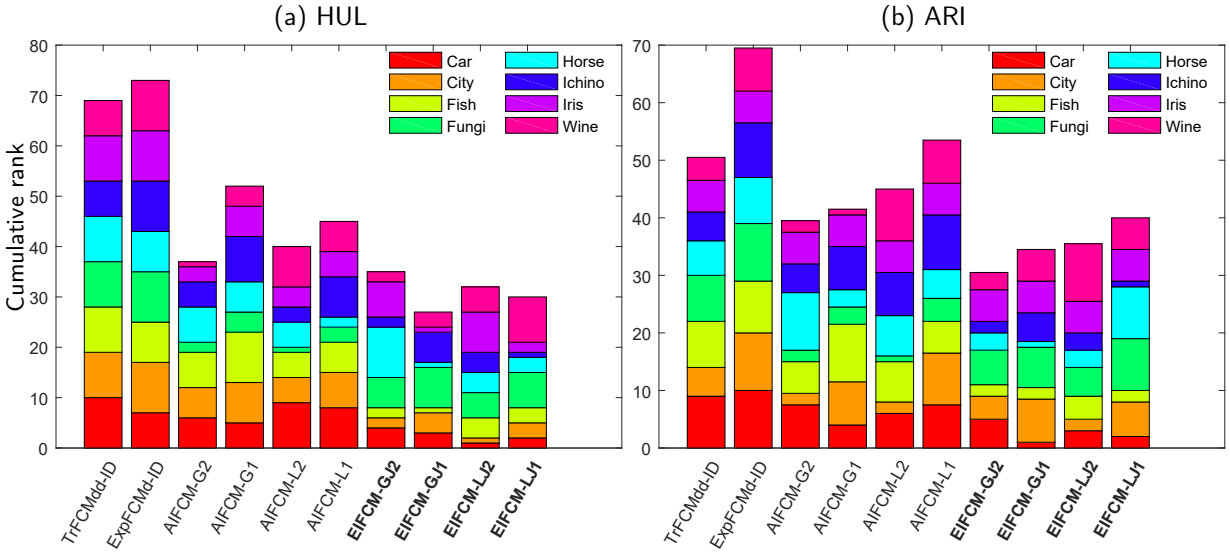
Dataset	N	P	C	Dataset	N	P	C
Car models	33	8	4	City temperature	37	12	4
Freshwater fish species	12	13	4	Fungi	55	5	3
Horse	12	7	4	Ichino	8	4	4
Iris	30	4	3	Wine	33	9	2

Source: Author (2022)

species, Horse and Iris, the proposed method EIFCM-GJ1 achieved the best clustering results. However, literature methods AIFCM-L2 and AIFCM-G2 show the best performance for Fungi and Wine, respectively. For Ichino data, the proposed algorithm EIFCM-LJ1 achieved the most significant values. Finally, Figure 27 (a) shows that the proposed methods outperformed other approaches, mainly EIFCM-GJ1, which achieved the best performance ranking. TrFCMdd-ID and ExpFCMd-ID had the worst results.

When comparing the hard partition and the a priori partition according to ARI, in the Iris dataset, all methods reached comparable clustering results. Please see Table 32. However, as shown in Figure 27 (b), EIFCM-GJ2, EIFCM-GJ1, and EIFCM-LJ2 presented the highest values for almost all datasets. AIFCM-L1 and ExpFCMd-ID obtained the worst results.

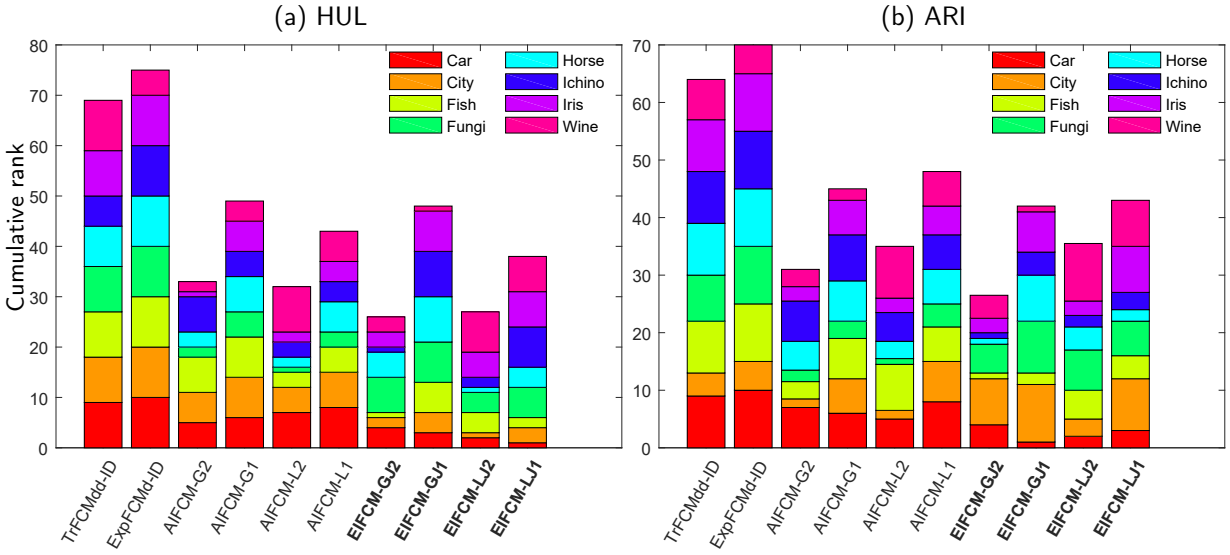
Figure 27 – Clustering results on real interval-valued data for (a) HUL and (b) ARI according to Table 32



Source: Author (2022)

The mean and standard deviation of the fifty iterations of the algorithms in the eight real interval-valued datasets were computed to analyze the sensitivity to initial cluster centers. Table 33 shows the numeric values for HUL and ARI. The performance rank of the algorithms is shown in parenthesis and summarized in Figure 28. We can observe that the

Figure 28 – Clustering results on real interval-valued data for (a) HUL and (b) ARI according to Table 33



Source: Author (2022)

proposed method IFCM-GJ2 outperformed other approaches on average for most datasets. Furthermore, it presented low standard deviations, indicating lower sensitivity to random initializations. This means that by using separate weights for lower and upper boundaries and entropy regularization, the performance of the algorithms can be improved.

4.4.4.2 Ichino dataset (Fats and Oils)

Ichino (ICHINO; YAGUCHI, 1994) is one of the real interval-valued data used in our experiments. It consists of eight objects described by four interval-valued variables, such as specific gravity, freezing point, iodine value, and saponification value. It is known that each object par (1,2), (3,4), (5,6), and (7,8) has similar properties. Then, this information was used to set the number of clusters equal to 4. In this data , almost all proposed methods presented the best performance for both indices. For a better understanding, it is analyzed the prototypes, weights of the variables, and fuzzy partition, provided by the proposed algorithm EIFCM-LJ1, which achieved the best clustering results.

Table 23 and Figure 29 present the prototypes for each cluster obtained by the proposed algorithm EIFCM-LJ1. We can see that the prototypes returned by the proposed method represent the Ichino dataset well. This allows obtaining a partition of the data that is in correspondence with the a priori classes.

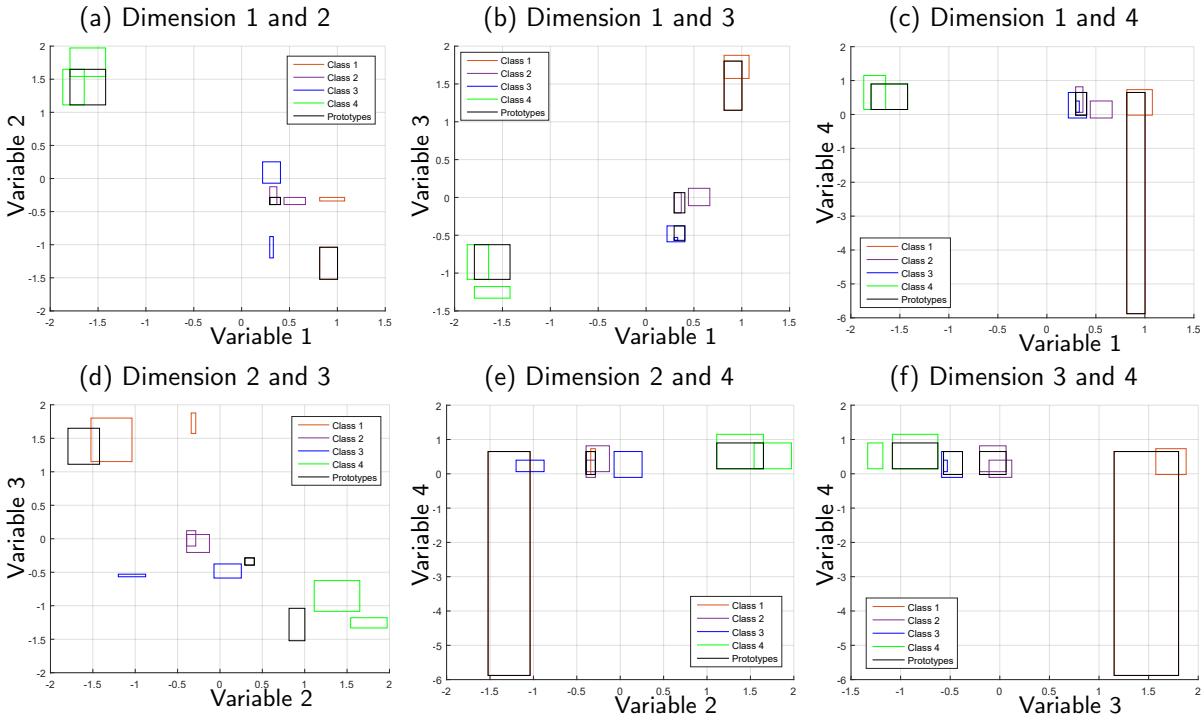
The algorithm EIFCM-LJ1 also provides the relevance weights of the interval-valued variables for lower and upper boundaries on each cluster as seen in Table 24. The closer the objects are to the prototype of a given group concerning a boundary of a given interval-valued variable,

Table 23 – Matrix of prototypes provided by proposed EIFCM-LJ1 for the Ichino dataset.

Variables	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Specific gravity	[-1.7937,-1.4210]	[0.8153,1.0017]	[0.2935,0.4053]	[0.2935,0.4053]
Freezing point	[1.1122,1.6498]	[-1.5221,-1.0383]	[-0.3931,-0.2856]	[-0.3931,-0.2856]
Iodine value	[-1.0830,-0.6245]	[1.1523,1.8019]	[-0.5672,-0.3761]	[-0.2042,0.0633]
Saponification value	[0.1464,0.8996]	[-5.8785,0.6485]	[-0.0209,0.6485]	[-0.0209,0.6485]

Source: Author (2022)

Figure 29 – Prototypes provided by the proposed algorithm EIFCM-LJ1 on the Ichino dataset.



Source: Author (2022)

the higher is the relevance weight of this boundary on this cluster. Table 24 shows that the saponification value is the most relevant variable for clusters 1, 3, and 4 in both boundaries. However, the specific gravity showed greater importance for group 2, mainly at the lower boundary. In every case, the higher-weight boundaries amplify the dissimilarity between objects and prototypes, while the lower values reduce it.

Finally, the proposed algorithm provides a fuzzy partition $\mathbf{U}$ of the objects as shown in Table 25 and the corresponding hard partition Q. When comparing them with the a priori partition, HUL=0.7615 and ARI=1. These results show how the proposed method manages to make a good partition of the input data based on the prototypes obtained and the weights of the variables, outperforming the literature approaches in both indices.

Table 24 – Relevance weights of the variables for lower and upper boundaries provided by the proposed algorithm EIFCM-LJ1 on the Ichino dataset.

Variables	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Specific gravity	[0.8080,0.6405]	[103.1783,2.5947]	[1.5052,1.2052]	[1.3983,1.0748]
Freezing point	[0.4703,0.5179]	[0.1635,0.2570]	[0.6975,0.6229]	[0.8382,0.7221]
Iodine value	[0.8820,0.5212]	[0.4602,2.5309]	[0.6109,0.5233]	[0.5716,0.5073]
Saponification value	[15.6518,1.1026]	[0.0330,2.3114]	[2.8426,1.3963]	[2.8574,1.3269]

Source: Author (2022)

Table 25 – Fuzzy and hard partitions provided by the proposed algorithm EIFCM-LJ1 on the Ichino dataset.

	Fuzzy Partition $\mathbf{U}$				
Objects	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Hard partition Q
1	0.0000	0.9995	0.0002	0.0003	2
2	0.0099	0.5622	0.1856	0.2422	2
3	0.0447	0.0000	0.4363	0.5190	4
4	0.0144	0.0000	0.4435	0.5422	4
5	0.0546	0.0000	0.5247	0.4207	3
6	0.0254	0.0000	0.5360	0.4386	3
7	0.9437	0.0000	0.0304	0.0260	1
8	0.9462	0.0000	0.0283	0.0255	1

Source: Author (2022)

4.5 CONCLUSIONS

New fuzzy clustering algorithms for interval-valued data were proposed. These methods are based on adaptive Euclidean and City-Block distances. These dissimilarity measures change with each iteration of the algorithms and consider that the boundaries of the variables have different relevance in the clustering process. The entropy regularization was considered because, usually, such approach is less sensitive to initial cluster centers and outliers.

The proposed methods are based on minimizing a clustering criterion to provide the best prototypes of the clusters, relevance weights of the variables, and fuzzy partition matrix. An expression for the minimizers of the objective function was obtained for each algorithm. Convergence properties were also studied.

In the experimental section, the properties and usefulness of proposed methods were demonstrated through experiments on synthetic and real interval-valued datasets. The results showed that weighting the variables separately, according to which boundary is more relevant, can improve the performance of the algorithms, especially when the structure of the clusters appears in one of the boundaries. The City-Block distance-based methods presented the best results for datasets with outliers. Besides, regularized approaches demonstrated higher

robustness to random initializations.

With these results, the research questions of this work were answered as follows:

1. Is it possible that considering the relevance of each boundary, the performance of the algorithms may be improved? How?

    - Proposing new objective functions with adaptive dissimilarity that consider the relevance of the variables for lower and upper boundaries allowed us to recognize clusters of different shapes and sizes in some subspaces of the variables, even in specific boundaries of the interval-valued data.
    - These approaches maintain the properties of the adaptive methods in the literature, such that if only a subset of the variables is relevant to define the groups in both boundaries, the methods assign to these variables a greater contribution to the clustering.
    - However, restricting the joint weights of the variable in both boundaries allowed the proposed methods to recognize the most relevant variable according to the most significant boundary, which is an advantage compared to the approaches in the literature. Therefore, the closer the objects are to a prototype of a given variable and boundary, the higher the significance of that variable in that boundary.
    - Experiments showed that the use of these techniques improves the clustering results when compared with literature approaches, mainly for datasets in which the upper and lower boundaries have different relevance.

2. What would be the advantage of using another type of distance to group interval-valued data?

    - Using robust dissimilarity functions allowed clustering methods to minimize the detrimental effect of outliers present in the data. These dissimilarity functions were based on the City-Block distance, since several experiments show that this type of approach is more robust to outliers.
    - Experiments on synthetic and real interval-valued data with different amounts of outliers showed that methods based on the City-Block distance degrade their performance more slowly compared to those based on the Euclidean distance.

3. What are the advantages of proposing clustering methods based on regularizations?

    - The maximum entropy clustering has received much attention in recent years due to its insensitivity to initial values and high clustering performance in large-scale, as an alternative to FCM approaches. A characteristic of taking this point of view of regularization is that the degrees of membership are taken linearly, which seems more natural than elevating them to a fuzziness coefficient. Besides, such methods

also possess definite mathematical characteristics and distinct physical meanings, which makes it easy to understand, as shown in Section 4.3.1.3.

- The experimental results show that the proposed methods with entropy regularization have a standard deviation lower than the methods in the literature, indicating a lower sensitivity to initialization and outliers.
- In proposed methods, the burden of representing fuzziness is shifted to the regularization term, in the form of a weighting factor that multiplies the regularization term added to the clustering criterion. Such approaches regularize 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 the simplicity of these approaches, other types of regularizations can be easily added.

5 FINAL REMARKS

5.1 CHALLENGES

The distance metric is an essential issue in many clustering algorithms. The commonly used Euclidean distance assumes that each feature is equally relevant and independent from others. Nevertheless, this assumption may not always be valid, especially when it comes to high-dimensional data. Besides, since the Euclidean distance-based dissimilarity measure only characterizes the mean information of a cluster, it is sensitive to noise and outliers. Therefore, selecting a distance metric with good quality should identify essential features and discriminate relevant and irrelevant ones, in addition to being robust to noise and outliers.

Partitioning clustering is usually approached by solving an optimization problem. This procedure involves finding the parameters that minimize (maximize) a given objective function constrained by some restrictions. Such a process may be simplified from the computational perspective whenever a differential function is available. Unfortunately, taking the derivative of the objective function is not always possible, which difficulties obtaining the expression for the minimizers (maximizers). Besides, in most cases, the objective function depends on hyper-parameters that control how far to move in the search space. Since any information is given about the input data, finding feasible values for the hyper-parameters in an unsupervised way is very challenging.

Continuous numerical data is a classical data type. Many algorithms for clustering this type of data can be found in the literature. However, this representation of objects may not represent complex, structured, aggregated, relational, and high-level data well. Symbolic Data Analysis was introduced as a domain associated with multivariate analysis that extends classical data analysis techniques to deal with complex data. Several methods have been proposed, for example, for interval-valued data clustering. Some of them even take into account the relevance of the variables in the clustering process. Still, the weight of significance of the lower and upper variables is considered equal. As a result, a non-relevant boundary can still have a high relevance weight. However, some interval boundaries associated with the variables could be more or less relevant or even irrelevant. Hence the need to propose methods capable of considering the information of relevance of both boundaries.

FCM methods are the most popular approaches for clustering data. Despite this, some points have been raised about the motivation for adding the parameter m to the objective function and its physical meaning (LI; MUKAIDONO, 1995), in addition to the unnatural appearance of such an exponent from a mathematical perspective. For example, why the membership degree is powered to it. Another point is that FCM methods are sensitive to initial cluster centers and noises. Hence the performance tends to deteriorate in some cases, especially with incomplete data (TAO et al., 2019).

5.2 CONTRIBUTIONS

Addressing the above points, new soft subspace clustering algorithms were proposed. These methods perform clustering in high-dimensional spaces by assigning a weight to each dimension to measure the contribution of individual features to the formation of the clusters. Mahalanobis-based methods have been considered. Experimental results show that such an approach is suitable when variables in the datasets are correlated. Besides, since outliers can be presented, methods based on the City-Block distance were introduced. Compared with other methods, they showed a robust performance in datasets with different percent of outliers and noise.

The proposals divide the input data into a fixed number of clusters by minimizing an objective function. An algebraic solution for obtaining the minimizers of the objective functions and a detailed derivation for all constraints and metrics were provided. An algorithmic solution was presented in the absence of an algebraic solution to get the prototype minimizer in City-Block distance-based approaches. The values of the hyper-parameters of the models significantly influence the result of the clustering. Hence, a sensitivity analysis was carried out to measure their influence on the quality of the clustering. Based on this study, an unsupervised process was introduced to calculate their optimal values.

Methods for interval-valued data clustering were also proposed. They consider the joint weights of the relevance of the lower and upper boundaries of the interval-valued variables. In this case, a boundary that plays a significant role in the clustering task will have a higher relevance weight. The experiments show that this new approach presented a good performance for synthetic and real datasets, especially when the structure of the clusters appears in one of the boundaries. In this last scenario, the proposed methods show a significant difference compared to other methods.

Finally, the proposals introduced an entropy term that regularizes the membership degree. They show a distinct physical meaning and well-defined mathematical characteristics because similarities with the method presented in Ref. (LI; MUKAIDONO, 1995) are shared. This approach offers a simple implementation and less computational complexity than FCM alternatives. Besides, the experimental results show that they are less sensitive to initial cluster centers and outliers.

5.3 LIMITATIONS AND FUTURE WORK

In clustering, the choice of dissimilarity functions is one of the main aspects of calculating the components of the prototypes. Several suitable dissimilarity functions have been proposed by different authors. However, sometimes an exact solution (algebraic or algorithmic) is not known to calculate the minimizers of the objective function. Hence the need to correctly select the dissimilarity measure given a problem and provide a detailed solution for each of the possible choices.

In this work, new fuzzy clustering algorithms were proposed that address the problem of data corrupted by outliers. In addition, methods based on the Mahalanobis distance were also introduced where the correlation between variables is considered. Such approaches have the property of being invariant to linear transformations. Despite their usefulness, other approaches can be proposed, for example, fuzzy kernel clustering methods. They can map the data into high-dimensional feature space by adopting an appropriate kernel function. Through such function, the non-hyperspherical-shaped groups in the original input space are likely to be transformed into hyperspherical-shaped clusters, which are easily identified by the method.

Entropy clustering algorithms have received much attention as they are insensitive to initial values and have high clustering performance in large-scale data. In addition, they possess definite mathematical characteristics and distinct physical meanings, which makes them easy to understand. Due to the simplicity of these approaches, other types of regularizations can be added. For example, Ref. (RODRÍGUEZ; CARVALHO, 2021b) proposed soft subspace clustering methods with quadratic regularization. The regularization function measures the overall fuzziness of the obtained classification pattern.

A follow-up extension for the above methods is to use different types of regularizations in addition to entropy and quadratic. An alternative may be the introduction of a fuzzification technique with Kullback-Leibler divergences (K-L information). Additionally, several regularizers can be combined into the same objective function to further constrain the solution space and find better data partitions. Besides, more straightforward methodologies to obtain free parameter values can be explored.

REFERENCES

AHMED, M. N.; YAMANY, S. M.; MOHAMED, N.; FARAG, A. A.; MORIARTY, T. A modified Fuzzy C-Means algorithm for bias field estimation and segmentation of MRI data. *IEEE Transactions on Medical Imaging*, IEEE, v. 21, n. 3, p. 193–199, 2002.

AKBARIZADEH, G.; RAHMANI, M. Efficient combination of texture and color features in a new spectral clustering method for PolSAR image segmentation. *National Academy Science Letters*, Springer, v. 40, n. 2, p. 117–120, 2017.

ANDERBERG, M. R. *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*. [S.l.]: Academic Press, 2014.

BACHE, K.; LICHMAN, M. *UCI Machine Learning Repository*. 2013. Available at: <http://archive.ics.uci.edu/ml>.

BEZDEK, J. C. Cluster validity with fuzzy sets. *J Cybern*, Taylor & Francis, v. 3, n. 3, p. 58–72, 1973.

BEZDEK, J. C. Mathematical models for systematics and taxonomy. In: FREEMAN. *Proceedings of the 8th International Conference on Numerical Taxonomy, 1975*. [S.l.], 1975. p. 143–166.

BEZDEK, J. C. *Pattern recognition with fuzzy objective function algorithms*. [S.l.]: Springer Science & Business Media, 2013.

BOCK, H.-H.; DIDAY, E. *Analysis of symbolic data: exploratory methods for extracting statistical information from complex data*. [S.l.]: Springer Science & Business Media, 2012.

BOUDOU, A.; RIBEYRE, F. Mercury in the food web: accumulation and transfer mechanisms. *Metal Ions in Biological Systems*, MARCEL DEKKER AG, v. 34, p. 289–320, 1997.

BOUGUILA, N. A model-based approach for discrete data clustering and feature weighting using MAP and stochastic complexity. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 21, n. 12, p. 1649–1664, 2009.

CAMPELLO, R. J.; MOULAVI, D.; SANDER, J. Density-based clustering based on hierarchical density estimates. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. [S.l.], 2013. p. 160–172.

CARVALHO, F. A. T. de; TENÓRIO, C. P.; JUNIOR, N. L. C. Partitional fuzzy clustering methods based on adaptive quadratic distances. *Fuzzy Sets and Systems*, Elsevier, v. 157, n. 21, p. 2833–2857, 2006.

CARVALHO, F. d. A. D.; LECHEVALLIER, Y. Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition*, Elsevier, v. 42, n. 7, p. 1223–1236, 2009.

CARVALHO, F. d. A. de. Fuzzy C-Means clustering methods for symbolic interval data. *Pattern Recognition Letters*, Elsevier, v. 28, n. 4, p. 423–437, 2007.

CARVALHO, F. d. A. de; BRITO, P.; BOCK, H.-H. Dynamic clustering for interval data based on l_2 distance. *Computational Statistics*, Springer, v. 21, n. 2, p. 231–250, 2006.

CARVALHO, F. d. A. de; LECHEVALLIER, Y. Dynamic clustering of interval-valued data based on adaptive quadratic distances. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, IEEE, v. 39, n. 6, p. 1295–1306, 2009.

CARVALHO, F. d. A. de; SIMÕES, E. C. Fuzzy clustering of interval-valued data with City-Block and Hausdorff distances. *Neurocomputing*, Elsevier, v. 266, p. 659–673, 2017.

CHARU, C. A.; CHANDAN, K. R. *Data clustering: algorithms and applications*. [S.l.]: Chapman and Hall/CRC Boston, MA, 2013.

CHEN, J.; ZENG, Y.; LI, Y.; HUANG, G.-B. Unsupervised feature selection based extreme learning machine for clustering. *Neurocomputing*, Elsevier, 2019.

CHEN, L.; WANG, S.; WANG, K.; ZHU, J. Soft subspace clustering of categorical data with probabilistic distance. *Pattern Recognition*, Elsevier, v. 51, p. 322–332, 2016.

CHEN, Y.; BILLARD, L. A study of divisive clustering with Hausdorff distances for interval data. *Pattern Recognition*, Elsevier, v. 96, p. 106969, 2019.

CHEUNG, Y.-m.; ZENG, H. A maximum weighted likelihood approach to simultaneous model selection and feature weighting in Gaussian mixture. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2007. p. 78–87.

CHIU, M.-C.; HSU, Y.-W. Using Fuzzy C-Means clustering based on integration of psychological and physiological data for therapeutic music design. *Journal of Industrial and Production Engineering*, Taylor & Francis, v. 34, n. 5, p. 382–397, 2017.

COPPI, R.; D'URSO, P. Fuzzy unsupervised classification of multivariate time trajectories with the Shannon entropy regularization. *Computational Statistics & Data Analysis*, Elsevier, v. 50, n. 6, p. 1452–1477, 2006.

DENG, Z.; CHOI, K.-S.; CHUNG, F.-L.; WANG, S. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. *Pattern Recognition*, Elsevier, v. 43, n. 3, p. 767–781, 2010.

DENG, Z.; CHOI, K.-S.; CHUNG, F.-L.; WANG, S. Eew-sc: enhanced entropy-weighting subspace clustering for high dimensional gene expression data clustering analysis. *Applied Soft Computing*, Elsevier, v. 11, n. 8, p. 4798–4806, 2011.

DENG, Z.; CHOI, K.-S.; JIANG, Y.; WANG, J.; WANG, S. A survey on soft subspace clustering. *Information Sciences*, Elsevier, v. 348, p. 84–106, 2016.

DIDAY, E. The symbolic approach in clustering and related methods of data analysis. *Proceedings of IFCS, Classification and Related Methods of Data Analysis*, p. 673–384, 1988.

DIDAY, E.; SIMON, J. Clustering analysis. In: *Digital Pattern Recognition*. [S.l.]: Springer, 1976. p. 47–94.

DU, H.; WANG, Z.; WANG, D.; WANG, X. Multi-feature fusion method applied in texture image segmentation. In: IEEE. *2018 14th International Conference on Computational Intelligence and Security (CIS)*. [S.l.], 2018. p. 135–139.

D'URSO, P.; GIOVANNI, L. D.; MASSARI, R. Trimmed fuzzy clustering for interval-valued data. *Advances in Data Analysis and Classification*, Springer, v. 9, n. 1, p. 21–40, 2015.

D'URSO, P.; MASSARI, R.; GIOVANNI, L. D.; CAPPELLI, C. Exponential distance-based fuzzy clustering for interval-valued data. *Fuzzy Optimization and Decision Making*, Springer, v. 16, n. 1, p. 51–70, 2017.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *The elements of statistical learning*. [S.l.]: Springer Series in Statistics New York, 2001.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

FRIGUI, H.; KRISHNAPURAM, R. A robust competitive clustering algorithm with applications in computer vision. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 21, n. 5, p. 450–465, 1999.

GAN, G.; MA, C.; WU, J. *Data clustering: theory, algorithms, and applications*. [S.l.]: SIAM, 2020.

GAN, G.; MA, C.; WU, J. *Data clustering: theory, algorithms, and applications*. [S.l.]: SIAM, 2020.

GORUNESCU, F. *Data Mining: concepts, models and techniques*. [S.l.]: Springer Science & Business Media, 2011.

GURU, D.; KIRANAGI, B. B.; NAGABHUSHAN, P. Multivalued type proximity measure and concept of mutual similarity value useful for clustering symbolic patterns. *Pattern Recognition Letters*, Elsevier, v. 25, n. 10, p. 1203–1213, 2004.

GUSTAFSON, D. E.; KESSEL, W. C. Fuzzy clustering with a fuzzy covariance matrix. In: IEEE. *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. [S.l.], 1979. p. 761–766.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Cluster validity methods: part i. *ACM Sigmod Record*, ACM New York, NY, USA, v. 31, n. 2, p. 40–45, 2002.

HANMANDLU, M.; VERMA, O. P.; SUSAN, S.; MADASU, V. K. Color segmentation by fuzzy co-clustering of chrominance color features. *Neurocomputing*, Elsevier, v. 120, p. 235–249, 2013.

HAVENS, T. C.; BEZDEK, J. C.; LECKIE, C.; HALL, L. O.; PALANISWAMI, M. Fuzzy C-Means algorithms for very large data. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 20, n. 6, p. 1130–1146, 2012.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. *Pattern Recognition Letters*, Elsevier, v. 24, n. 9-10, p. 1641–1650, 2003.

HUANG, J. Z.; NG, M. K.; RONG, H.; LI, Z. Automated variable weighting in K-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 27, n. 5, p. 657–668, 2005.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of Classification*, Springer, v. 2, n. 1, p. 193–218, 1985.

HULLERMEIER, E.; RIFQI, M.; HENZGEN, S.; SENGE, R. Comparing fuzzy partitions: a generalization of the rand index and related measures. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 20, n. 3, p. 546–556, 2011.

ICHINO, M.; YAGUCHI, H. Generalized Minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 24, n. 4, p. 698–708, 1994.

IRPINO, A.; VERDE, R. Dynamic clustering of interval data using a Wasserstein-based distance. *Pattern Recognition Letters*, Elsevier, v. 29, n. 11, p. 1648–1658, 2008.

IRPINO, A.; VERDE, R.; CARVALHO, F. d. A. de. Fuzzy clustering of distributional data with automatic weighting of variable components. *Information Sciences*, Elsevier, v. 406, p. 248–268, 2017.

JAIN, A. K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys (CSUR)*, ACM, v. 31, n. 3, p. 264–323, 1999.

JAJUGA, K. L_1-norm based fuzzy clustering. *Fuzzy Sets and Systems*, Elsevier, v. 39, n. 1, p. 43–50, 1991.

JIN, W.; TUNG, A. K.; HAN, J.; WANG, W. Ranking outliers using symmetric neighborhood relationship. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. [S.l.], 2006. p. 577–593.

KALLITHRAKA, S.; ARVANITOYANNIS, I.; KEFALAS, P.; EL-ZAJOULI, A.; SOUFLEROS, E.; PSARRA, E. Instrumental and sensory analysis of Greek wines; implementation of Principal Component Analysis (PCA) for classification according to geographical origin. *Food Chemistry*, Elsevier, v. 73, n. 4, p. 501–514, 2001.

KARST, O. J. Linear curve fitting using least deviations. *Journal of the American Statistical Association*, Taylor & Francis, v. 53, n. 281, p. 118–132, 1958.

KAUFMAN, L.; ROUSSEEUW, P. *Clustering by means of medoids*. [S.l.]: North-Holland, 1987.

KAUFMAN, L.; ROUSSEEUW, P. J. *Finding groups in data: an introduction to cluster analysis*. [S.l.]: John Wiley & Sons, 2009.

KOTSIANTIS, S.; PINTELAS, P. Recent advances in clustering: a brief survey. *WSEAS Transactions on Information Science and Applications*, v. 1, n. 1, p. 73–81, 2004.

KRISHNAPURAM, R.; JOSHI, A.; NASRAOUI, O.; YI, L. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 9, n. 4, p. 595–607, 2001.

KULKARNI, A.; TOKEKAR, V.; KULKARNI, P. Discovering context of labelled text documents using context similarity coefficient. *Procedia Computer Science*, v. 49, p. 118–127, 2015.

KYRKI, V.; KAMARAINEN, J.-K.; KÄLVIÄINEN, H. Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters*, Elsevier, v. 25, n. 3, p. 311–318, 2004.

LESKI, J. M. et al. Fuzzy c-ordered medoids clustering for interval-valued data. *Pattern Recognition*, Elsevier, v. 58, p. 49–67, 2016.

LI, R.-P.; MUKAIDONO, M. A maximum-entropy approach to fuzzy clustering. In: IEEE. *Proceedings of 1995 IEEE International Conference on Fuzzy Systems.* [S.l.], 1995. v. 4, p. 2227–2232.

LYNNE, B.; EDWIN, D. Symbolic data analysis: conceptual statistics and data mining. *John Wiley & Sons*, 2006.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.], 1967. v. 1, p. 281–297.

MAHMOUDI, M. R.; BALEANU, D.; MANSOR, Z.; TUAN, B. A.; PHO, K.-H. Fuzzy clustering method to compare the spread rate of Covid-19 in the high risks countries. *Chaos, Solitons & Fractals*, Elsevier, v. 140, p. 110230, 2020.

MIZUTANI, K.; MIYAMOTO, S. Possibilistic approach to kernel-based Fuzzy C-Means clustering with entropy regularization. In: SPRINGER. *International Conference on Modeling Decisions for Artificial Intelligence*. [S.l.], 2005. p. 144–155.

NAMBURU, A.; SAMAY, S. kumar; EDARA, S. R. Soft fuzzy rough set-based MR brain image segmentation. *Applied Soft Computing*, Elsevier, v. 54, p. 456–466, 2017.

NAYAK, J.; NAIK, B.; BEHERA, H. Fuzzy C-Means (FCM) clustering algorithm: a decade review from 2000 to 2014. *Computational Intelligence in Data Mining*, Springer, v. 2, p. 133–149, 2015.

NEMENYI, B. *Distribution-free multiple comparison*. Phd Thesis (PhD Thesis) — Princeton University, 1963.

PAL, N. R.; PAL, S. K. A review on image segmentation techniques. *Pattern Recognition*, Elsevier, v. 26, n. 9, p. 1277–1294, 1993.

PARSONS, L.; HAQUE, E.; LIU, H. Subspace clustering for high dimensional data: a review. *ACM Sigkdd Explorations Newsletter*, ACM New York, NY, USA, v. 6, n. 1, p. 90–105, 2004.

PIMENTEL, B. A.; SOUZA, R. M. de. A weighted multivariate Fuzzy C-Means method in interval-valued scientific production data. *Expert Systems with Applications*, Elsevier, v. 41, n. 7, p. 3223–3236, 2014.

QIAN, P.; ZHAO, K.; JIANG, Y.; SU, K.-H.; DENG, Z.; WANG, S.; JR, R. F. M. Knowledge-leveraged transfer Fuzzy C-Means for texture image segmentation with self-adaptive cluster prototype matching. *Knowledge-Based Systems*, Elsevier, v. 130, p. 33–50, 2017.

RANDEN, T. *Brodatz Texture*. <http://www.ux.uis.no/~tranden/brodatz.html>. [Online; Accessed November 10, 2021].

REDDY, C. K.; VINZAMURI, B. A survey of partitional and hierarchical clustering algorithms. *Data Clustering: Algorithms and Applications*, v. 87, 2013.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Fuzzy clustering algorithm with automatic variable selection and entropy regularization. In: IEEE. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.], 2017. p. 1–6.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Fuzzy clustering algorithm based on adaptive City-block distance and entropy regularization. In: IEEE. *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.], 2018. p. 1–8.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Fuzzy clustering algorithm based on adaptive Euclidean distance and entropy regularization for interval-valued data. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2018. p. 695–705.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. A new fuzzy clustering algorithm for interval-valued data based on City-Block distance. In: IEEE. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.], 2019. p. 1–6.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Clustering interval-valued data with automatic variables weighting. In: IEEE. *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2019. p. 1–8.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Fuzzy clustering algorithms with distance metric learning and entropy regularization. *Applied Soft Computing*, p. 107922, 2021.

RODRÍGUEZ, S. I.; CARVALHO, F. d. A. de. Soft subspace clustering of interval-valued data with regularizations. *Knowledge-Based Systems*, Elsevier, v. 227, p. 107191, 2021.

RUSPINI, E. H.; BEZDEK, J. C.; KELLER, J. M. Fuzzy clustering: a historical perspective. *IEEE Computational Intelligence Magazine*, IEEE, v. 14, n. 1, p. 45–55, 2019.

SADAAKI, M.; MASAO, M. Fuzzy C-Means as a regularization and maximum entropy approach. In: *Proceedings of the 7th International Fuzzy Systems Association World Congress (IFSA'97)*. [S.l.: s.n.], 1997. v. 2, p. 86–92.

SCHWÄMMLE, V.; JENSEN, O. N. A simple and fast method to determine the parameters for Fuzzy C-Means cluster analysis. *Bioinformatics*, Oxford University Press, v. 26, n. 22, p. 2841–2848, 2010.

SILVA, P. D.; BRITO, P. *Model and Analyse Interval Data*. 2020. <https://cran.r-project.org/web/packages/MAINT.Data/index.html>. [Online; Accessed November 10, 2021].

SING, J. K.; ADHIKARI, S. K.; BASU, D. K. A modified Fuzzy C-Means algorithm using scale control spatial information for MRI image segmentation in the presence of noise. *Journal of Chemometrics*, Wiley Online Library, v. 29, n. 9, p. 492–505, 2015.

SOUZA, L. C. de; SOUZA, R. M. C. R. D.; AMARAL, G. J. A. do. Dynamic clustering of interval data based on hybrid l_q distance. *Knowledge and Information Systems*, Springer, v. 62, n. 2, p. 687–718, 2020.

SOUZA, R. M. de; CARVALHO, F. d. A. D. Clustering of interval data based on City-Block distances. *Pattern Recognition Letters*, Elsevier, v. 25, n. 3, p. 353–365, 2004.

TANG, Y.; HU, X.; PEDRYCZ, W.; SONG, X. Possibilistic fuzzy clustering with high-density viewpoint. *Neurocomputing*, Elsevier, v. 329, p. 407–423, 2019.

TANG, Y.; REN, F.; PEDRYCZ, W. Fuzzy C-Means clustering through SSIM and patch for image segmentation. *Applied Soft Computing*, Elsevier, v. 87, p. 105928, 2020.

TAO, X.; WANG, R.; CHANG, R.; LI, C. Density-sensitive fuzzy kernel maximum entropy clustering algorithm. *Knowledge-Based Systems*, Elsevier, v. 166, p. 42–57, 2019.

TIKHONOV, A. N. Solution of incorrectly formulated problems and the regularization method. In: *Dokl. Akad. Nauk.* [S.l.: s.n.], 1963. v. 151, p. 1035–1038.

TSAI, C.-Y.; CHIU, C.-C. Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. *Computational Statistics & Data Analysis*, Elsevier, v. 52, n. 10, p. 4658–4672, 2008.

WANG, J.; DENG, Z.; CHOI, K.-S.; JIANG, Y.; LUO, X.; CHUNG, F.-L.; WANG, S. Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, Elsevier, v. 52, p. 113–134, 2016.

WANG, X.; WANG, Y.; WANG, L. Improving Fuzzy C-Means clustering based on feature-weight learning. *Pattern Recognition Letters*, Elsevier, v. 25, n. 10, p. 1123–1132, 2004.

WOOD, M.; STEVENS, F. *The Fungi of California*. 2015. <http://www.mykoweb.com/CAF/species_index.html>. [Online; Accessed November 10, 2021].

WU, J.; XIONG, H.; LIU, C.; CHEN, J. A generalization of distance functions for Fuzzy C-Means clustering with centroids of arithmetic means. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 20, n. 3, p. 557–571, 2011.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, Ieee, v. 16, n. 3, p. 645–678, 2005.

XU, R.; WUNSCH, D. *Clustering*. [S.l.]: John Wiley & Sons, 2008.

ZHAO, F.; LIU, H.; FAN, J. A multiobjective spatial fuzzy clustering algorithm for image segmentation. *Applied Soft Computing*, Elsevier, v. 30, p. 48–57, 2015.

ZHU, L.; CAO, L.; YANG, J.; LEI, J. Evolving soft subspace clustering. *Applied Soft Computing*, Elsevier, v. 14, p. 210–228, 2014.

APPENDIX A – RELATED PUBLICATIONS

This appendix contains the information where the articles (RODRÍGUEZ; CARVALHO, 2021a; RODRÍGUEZ; CARVALHO, 2021b; RODRÍGUEZ; CARVALHO, 2019b; RODRÍGUEZ; CARVALHO, 2019a; RODRÍGUEZ; CARVALHO, 2018a; RODRÍGUEZ; CARVALHO, 2018b) were published.

ScienceDirect

View PDF

Applied Soft Computing
Available online 29 September 2021, 107922
In Press, Journal Pre-proof

Fuzzy clustering algorithms with distance metric learning and entropy regularization

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

Show more

Outline | Share | Cite

https://doi.org/10.1016/j.asoc.2021.107922

Get rights and content

Highlights

- The paper proposes fuzzy clustering algorithms with entropy regularization.
- It is used local and global adaptive Euclidean, Mahalanobis and City-block metrics.
- The approaches considers the weights of relevance or the covariance of the variables.
- Experiments with different kind of data sets shows the usefulness of the approaches.

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.

Keywords

Fuzzy clustering; Prototype-based clustering; Distance metric learning; Maximum-entropy regularization

FEEDBACK

Knowledge-Based Systems
Volume 227, 5 September 2021, 107191

Soft subspace clustering of interval-valued data with regularizations

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

https://doi.org/10.1016/j.knosys.2021.107191

Abstract

Data analysis plays an indispensable role in understanding different phenomena. One of the vital means of handling these data is to group them into a set of clusters given a measure of similarity. Usually, clustering methods deal with objects described by single-valued variables. Nevertheless, this representation is too restrictive for representing complex data, such as lists, histograms, or even intervals. Furthermore, in some problems, many dimensions are irrelevant and can mask existing clusters. In this regard, new interval-valued data clustering methods with regularizations and adaptive distances are proposed. These approaches consider that the boundaries of the interval-valued variables have the same and different importance for the clustering process. The algorithms optimize an objective function alternating three steps for obtaining the representatives of each group, a fuzzy partition, and the relevance weights of the variables. Experiments on synthetic and real data sets corroborate the robustness and usefulness of the proposed methods.

Clustering interval-valued data with automatic variables weighting

Publisher: IEEE

2 Author(s) Sara Inés Rizo Rodríguez ; Francisco de Assis Tenório de Carvalho All Authors

Abstract:
Over the past few years, Symbolic Data Analysis has gained popularity providing suitable methods for managing aggregated data represented by lists, intervals, histograms or even distributions. This paper proposes a partitioning clustering algorithm for interval-valued data based on the suitable adaptive Euclidean distance that takes into account the relevance of the variables according to the boundaries. The proposed distance changes at each algorithm iteration and is different from one cluster to another. The method provides a partition and a prototype for each cluster by optimizing an adequacy criterion that measures the fitting between groups and their representatives. Experiments on synthetic and real interval-valued datasets corroborate the usefulness of the proposed algorithm.

Published in: 2019 International Joint Conference on Neural Networks (IJCNN)

Date of Conference: 14-19 July 2019
INSPEC Accession Number: 19028154
Date Added to IEEE *Xplore*: 30 September 2019
DOI: 10.1109/IJCNN.2019.8852220
ISBN Information:
Publisher: IEEE
ISSN Information:
Conference Location: Budapest, Hungary, Hungary

Fuzzy clustering Algorithm based on Adaptive City-block distance and Entropy Regularization

Publisher: IEEE

2 Author(s) Sara I.R. Rodríguez ; Francisco de A.T. de Carvalho All Authors

Abstract:
The Euclidean distance is traditionally used to compare the objects and the prototypes in the Fuzzy C-Means algorithms, but theoretical studies indicate that methods based on City-Block distances are more robust concerning the presence of outliers in the dataset than those based on Euclidean distances. Moreover, most often conventional Fuzzy C-Means clustering algorithms consider that all variables are equally important for the clustering task. However, in real situations, some variables may be more or less relevant or even irrelevant for clustering. This paper proposes a partitioning fuzzy clustering algorithm based on Adaptive City-block distances and entropy regularization. The proposed method optimizes an objective function by alternating three steps aiming to compute the fuzzy cluster representatives, the fuzzy partition, as well as relevance weights for the variables. Several experiments on synthetic and real-world datasets including its application to noisy image texture segmentation are presented to corroborate both clustering and robustness capabilities of the proposed algorithm over conventional approaches.

Published in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)

Date of Conference: 8-13 July 2018
Date Added to IEEE *Xplore*: 15 October 2018
ISBN Information:
INSPEC Accession Number: 18181549
DOI: 10.1109/FUZZ-IEEE.2018.8491441
Publisher: IEEE
Conference Location: Rio de Janeiro, Brazil

A new fuzzy clustering algorithm for interval-valued data based on City-Block distance

Sara Inés Rizo Rodríguez ; Francisco de Assis Tenório de Carvalho All Authors

Abstract:
Interval-valued data are needed, for example, when an object represents a group of individuals and the variables used to describe it need to assume a value which expresses the variability inherent to the description of a group. Interval-valued data arise in practical situations such as recording monthly interval temperatures at meteorological stations, daily interval stock prices, etc. In this paper is proposed a robust partitioning fuzzy clustering algorithm for interval-valued data based on adaptive City-Block distance that takes into account the relevance of the variables according to the boundaries. This distance changes at each iteration of the algorithm and is different from one cluster to another. The method optimizes an objective function by alternating three steps to compute the representatives of each group, the fuzzy partition, as well as relevance weights for the interval-valued variables for each boundary. Experiments on synthetic and real interval-valued datasets corroborate the usefulness and robustness of the proposed algorithm.

Published in: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)

Date of Conference: 23-26 June 2019
INSPEC Accession Number: 19046329
Date Added to IEEE *Xplore*: 10 October 2019
DOI: 10.1109/FUZZ-IEEE.2019.8859017
ISBN Information:
Publisher: IEEE
ISSN Information:
Conference Location: New Orleans, LA, USA, USA

International Conference on Artificial Neural Networks
ICANN 2018: Artificial Neural Networks and Machine Learning – ICANN 2018 pp 695-705 | Cite as

Fuzzy Clustering Algorithm Based on Adaptive Euclidean Distance and Entropy Regularization for Interval-Valued Data

Authors
Authors and affiliations

Sara Inés Rizo Rodríguez , Francisco de Assis Tenorio de Carvalho

Conference paper
First Online: 27 September 2018

Part of the Lecture Notes in Computer Science book series (LNCS, volume 11139)

Abstract

Symbolic Data Analysis provides suitable new types of variable that can take into account the variability present in the observed measurements. This paper proposes a partitioning fuzzy

APPENDIX B – RESULTS FOR THE EXPERIMENTS ON SINGLE-VALUED DATASETS.

Table 26 – Mean and standard deviation (in parentheses) for the four datasets in the first experiment.

	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.7006	0.5991	0.6735	0.5815	0.6314	0.4162	0.6368	0.4613
	(0.0231)	(0.0388)	(0.0186)	(0.0343)	(0.0206)	(0.0338)	(0.0220)	(0.0402)
EFCM-1	**0.7946**	**0.6383**	0.7926	0.5693	0.7140	0.4393	0.7549	0.4529
	(0.0788)	(0.0815)	(0.0344)	(0.0555)	(0.0659)	(0.0465)	(0.0205)	(0.0344)
EFCM-M	0.7091	0.5899	0.6190	0.5186	0.7178	**0.6631**	0.6197	0.4946
	(0.0225)	(0.0465)	(0.0365)	(0.0344)	(0.0485)	(0.1366)	(0.0372)	(0.0294)
EFCM-Mk	0.6685	0.4127	0.7427	0.5990	0.7295	0.4936	**0.7933**	**0.6499**
	(0.0887)	(0.1111)	(0.0821)	(0.0681)	(0.0901)	(0.1581)	(0.0660)	(0.1128)
EFCM-GP2	0.7063	0.5905	0.6648	0.5621	0.6357	0.4131	0.6347	0.4393
	(0.0219)	(0.0399)	(0.0252)	(0.0353)	(0.0203)	(0.0339)	(0.0197)	(0.0378)
EFCM-GP1	0.7739	0.6226	**0.7995**	0.5819	**0.7322**	0.4506	0.7545	0.4506
	(0.0759)	(0.0801)	(0.0353)	(0.0641)	(0.0600)	(0.0595)	(0.0310)	(0.0464)
EFCM-LP2	0.7055	0.4393	0.7954	**0.6466**	0.6424	0.3181	0.6942	0.5651
	(0.0169)	(0.0529)	(0.0342)	(0.0409)	(0.0130)	(0.0241)	(0.0443)	(0.0390)
EFCM-LP1	0.6904	0.5328	0.7552	0.6452	0.6673	0.3994	0.7176	0.5168
	(0.0593)	(0.0769)	(0.0530)	(0.0600)	(0.0365)	(0.0438)	(0.0537)	(0.0533)
EFCM-GS2	0.6836	0.4892	0.6890	0.5193	0.6327	0.3814	0.6515	0.4175
	(0.0368)	(0.1425)	(0.0422)	(0.0984)	(0.0229)	(0.0705)	(0.0336)	(0.0889)
EFCM-GS1	0.6663	0.5142	0.7301	0.4444	0.6267	0.3987	0.7034	0.4175
	(0.0537)	(0.1054)	(0.0509)	(0.1298)	(0.0523)	(0.0775)	(0.0344)	(0.0876)
EFCM-LS2	0.6936	0.4861	0.6988	0.5556	0.6394	0.3579	0.6483	0.4895
	(0.0344)	(0.1475)	(0.0390)	(0.0599)	(0.0235)	(0.0770)	(0.0276)	(0.0467)
EFCM-LS1	0.6528	0.4243	0.7443	0.5432	0.6245	0.3611	0.6959	0.4864
	(0.0556)	(0.1392)	(0.0470)	(0.0464)	(0.0517)	(0.0875)	(0.0589)	(0.0708)

Source: Author (2022)

Table 27 – Mean and standard deviation (in parentheses) of HUL and ARI on the synthetic single-valued dataset for different percentages of outliers.

	0% Outliers		10% Outliers		20% Outliers	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5726	**0.9761**	0.5120	0.8103	0.5077	0.6524
	(0.0226)	(0.0350)	(0.0087)	(0.3117)	(0.0097)	(0.3847)
EFCM-1	0.7708	0.9683	0.7143	0.9683	0.7383	0.9663
	(0.1351)	(0.0431)	(0.1308)	(0.0431)	(0.1495)	(0.0416)
EFCM-M	0.6319	0.8726	0.5214	0.8285	0.5108	0.7343
	(0.1209)	(0.2517)	(0.0179)	(0.2573)	(0.0067)	(0.2429)
EFCM-Mk	0.6572	0.4467	0.5308	0.1963	0.5335	0.2199
	(0.2210)	(0.4029)	(0.0964)	(0.2552)	(0.0691)	(0.2561)
EFCM-GP2	0.5392	**0.9761**	0.5117	0.8145	0.5051	0.6548
	(0.0646)	(0.0350)	(0.0094)	(0.3042)	(0.0070)	(0.3847)
EFCM-GP1	0.7716	0.9692	**0.7407**	**0.9732**	**0.7385**	0.9633
	(0.1335)	(0.0397)	(0.1449)	(0.0389)	(0.1478)	(0.0432)
EFCM-LP2	0.5657	**0.9761**	0.5033	0.3116	0.5051	0.4161
	(0.0338)	(0.0350)	(0.0300)	(0.3210)	(0.0309)	(0.3074)
EFCM-LP1	0.7703	0.9653	0.6903	0.9673	0.6759	0.9586
	(0.1351)	(0.0436)	(0.0996)	(0.0440)	(0.1029)	(0.0504)
EFCM-GS2	0.5062	**0.9761**	0.5105	0.8128	0.5063	0.6533
	(0.0175)	(0.0350)	(0.0075)	(0.3044)	(0.0101)	(0.3838)
EFCM-GS1	**0.7956**	0.9702	0.7248	**0.9732**	0.7287	**0.9672**
	(0.1454)	(0.0399)	(0.1368)	(0.0389)	(0.1589)	(0.0394)
EFCM-LS2	0.5444	0.8795	0.4997	0.1466	0.4982	0.1971
	(0.0271)	(0.2741)	(0.0233)	(0.2276)	(0.0103)	(0.2515)
EFCM-LS1	0.7495	0.9491	0.6383	0.8515	0.5898	0.7302
	(0.1804)	(0.0642)	(0.1724)	(0.2542)	(0.1506)	(0.3209)

Source: Author (2022)

Table 28 – Algorithms performance for real datasets.

	Automobile		Balance Scale		Haberman	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5692 (8.0)	0.0941 (10.0)	0.4568 (6.0)	0.1420 (2.0)	0.5892 (8.0)	-0.0026 (10.0)
EFCM-1	**0.7180** (1.0)	0.1474 (4.0)	0.4299 (11.0)	0.0000 (10.0)	0.4990 (11.0)	-0.0011 (8.0)
EFCM-M	0.4208 (12.0)	0.1073 (6.0)	0.4535 (7.0)	0.0652 (7.0)	0.5844 (9.0)	0.0035 (6.0)
EFCM-Mk	0.4806 (11.0)	0.0253 (12.0)	**0.5741** (1.0)	0.1131 (3.5)	**0.6086** (1.0)	0.1596 (3.0)
EFCM-GP2	0.7073 (3.0)	0.1415 (5.0)	0.5121 (2.0)	0.1131 (3.5)	0.6025 (5.0)	-0.0027 (11.0)
EFCM-GP1	0.7084 (2.0)	**0.2160** (1.0)	0.4299 (11.0)	0.0000 (10.0)	0.4987 (12.0)	-0.0011 (8.0)
EFCM-LP2	0.5585 (9.0)	0.1060 (7.0)	0.4640 (5.0)	0.0729 (6.0)	0.6084 (2.0)	0.1456 (4.0)
EFCM-LP1	0.6440 (6.0)	0.0869 (11.0)	0.4977 (4.0)	-0.0110 (12.0)	0.6047 (4.0)	0.1725 (2.0)
EFCM-GS2	0.5700 (7.0)	0.0971 (9.0)	0.4474 (8.0)	**0.2934** (1.0)	0.5097 (10.0)	-0.0040 (12.0)
EFCM-GS1	0.6790 (4.0)	0.1604 (3.0)	0.4299 (11.0)	0.0000 (10.0)	0.6018 (6.0)	-0.0011 (8.0)
EFCM-LS2	0.5336 (10.0)	0.0975 (8.0)	0.4992 (3.0)	0.1024 (5.0)	0.6072 (3.0)	0.1001 (5.0)
EFCM-LS1	0.6683 (5.0)	0.2020 (2.0)	0.4299 (9.0)	0.0011 (8.0)	0.6005 (7.0)	**0.1789** (1.0)
	Heart		Image Segmentation		Ionosphere	
EFCM-2	0.5451 (5.0)	0.3487 (8.0)	0.8027 (6.0)	0.4911 (6.0)	0.5358 (12.0)	0.1588 (4.0)
EFCM-1	**0.6910** (1.0)	0.4227 (4.0)	0.8627 (2.0)	**0.5257** (1.0)	0.5377 (8.0)	0.0936 (9.0)
EFCM-M	0.5156 (7.0)	0.1338 (10.0)	0.1425 (12.0)	0.0041 (12.0)	0.5407 (5.0)	0.0085 (12.0)
EFCM-Mk	0.5027 (12.0)	-0.0036 (11.5)	0.2101 (11.0)	0.0151 (11.0)	**0.5713** (1.0)	0.0321 (11.0)
EFCM-GP2	0.5050 (9.0)	0.3576 (7.0)	0.8365 (3.0)	0.4309 (8.0)	0.5384 (7.0)	0.1588 (4.0)
EFCM-GP1	0.5908 (4.0)	0.1807 (9.0)	**0.8659** (1.0)	0.5221 (2.0)	0.5418 (4.0)	0.1045 (8.0)
EFCM-LP2	0.5312 (6.0)	0.4131 (5.0)	0.7281 (9.0)	0.3242 (9.0)	0.5360 (11.0)	0.1406 (6.0)
EFCM-LP1	0.5046 (10.0)	-0.0036 (11.5)	0.6862 (10.0)	0.2979 (10.0)	0.5362 (10.0)	0.0870 (10.0)
EFCM-GS2	0.5045 (11.0)	0.4325 (3.0)	0.8008 (7.0)	0.5021 (4.0)	0.5384 (6.0)	0.1588 (4.0)
EFCM-GS1	0.6792 (3.0)	**0.4423** (1.5)	0.8300 (4.0)	0.5137 (3.0)	0.5558 (2.0)	0.1243 (7.0)
EFCM-LS2	0.5111 (8.0)	0.3757 (6.0)	0.7937 (8.0)	0.4864 (7.0)	0.5365 (9.0)	0.1634 (2.0)
EFCM-LS1	0.6794 (2.0)	**0.4423** (1.5)	0.8291 (5.0)	0.4928 (5.0)	0.5508 (3.0)	**0.2092** (1.0)
	Iris plants		Mnist		Thyroid	
EFCM-2	0.7524 (11.0)	0.6199 (11.0)	0.5072 (6.0)	0.9564 (3.0)	0.5997 (12.0)	0.3623 (8.0)
EFCM-1	0.8538 (6.0)	0.6656 (7.5)	**0.5660** (1.0)	0.9503 (8.0)	**0.8702** (1.0)	**0.7324** (1.0)
EFCM-M	0.9020 (2.0)	**0.9037** (1.0)	0.5538 (3.0)	**0.9569** (1.5)	0.6106 (11.0)	0.1136 (11.0)
EFCM-Mk	0.5870 (12.0)	0.2824 (12.0)	0.5538 (2.0)	**0.9569** (1.5)	0.7184 (4.0)	0.0931 (12.0)
EFCM-GP2	0.8878 (4.0)	0.8510 (3.0)	0.5022 (11.0)	0.9561 (4.5)	0.6547 (8.0)	0.5038 (4.0)
EFCM-GP1	**0.9481** (1.0)	0.8857 (2.0)	0.5082 (5.0)	0.9532 (6.0)	0.8586 (2.0)	0.7167 (2.0)
EFCM-LP2	0.7811 (8.0)	0.6882 (5.0)	0.5168 (4.0)	0.0513 (9.0)	0.6892 (5.0)	0.6931 (3.0)
EFCM-LP1	0.8931 (3.0)	0.8019 (4.0)	0.5033 (7.0)	0.0169 (10.0)	0.7287 (3.0)	0.4148 (7.0)
EFCM-GS2	0.7607 (9.0)	0.6303 (9.5)	0.5022 (10.0)	0.9561 (4.5)	0.6611 (7.0)	0.4731 (5.0)
EFCM-GS1	0.8535 (7.0)	0.6656 (7.5)	0.5028 (8.0)	0.9511 (7.0)	0.6830 (6.0)	0.3337 (9.0)
EFCM-LS2	0.7549 (10.0)	0.6303 (9.5)	0.5021 (12.0)	0.0041 (12.0)	0.6215 (9.0)	0.4392 (6.0)
EFCM-LS1	0.8584 (5.0)	0.6757 (6.0)	0.5023 (9.0)	0.0096 (11.0)	0.6208 (10.0)	0.1349 (10.0)

Source: Author (2022)

Table 28 – Algorithms performance for real datasets.

	UKM		Vehicle		Vertebral column	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.4900 (10.0)	0.1647 (11.0)	0.5660 (8.0)	0.0736 (11.0)	0.5749 (6.0)	0.2993 (9.0)
EFCM-1	0.6095 (6.0)	0.1935 (9.0)	0.6503 (3.0)	0.1124 (7.0)	0.6016 (5.0)	0.3330 (8.0)
EFCM-M	0.7151 (2.0)	**0.3530** (1.0)	0.6653 (2.0)	0.1399 (2.0)	0.3811 (12.0)	0.2479 (10.0)
EFCM-Mk	0.4269 (12.0)	0.0176 (12.0)	0.4362 (12.0)	0.0951 (8.0)	0.4658 (10.0)	0.0045 (12.0)
EFCM-GP2	0.6937 (3.0)	0.3024 (4.0)	0.5535 (9.0)	0.0755 (10.0)	0.5640 (7.0)	0.3344 (7.0)
EFCM-GP1	**0.7373** (1.0)	0.3499 (2.0)	0.6435 (4.0)	0.1269 (5.0)	0.6131 (4.0)	0.3416 (4.0)
EFCM-LP2	0.5459 (7.0)	0.2750 (6.0)	0.5730 (7.0)	0.1299 (4.0)	0.5595 (9.0)	0.3526 (3.0)
EFCM-LP1	0.6633 (4.0)	0.3264 (3.0)	**0.6695** (1.0)	**0.1425** (1.0)	0.6204 (3.0)	0.3398 (5.0)
EFCM-GS2	0.4728 (11.0)	0.1679 (10.0)	0.5416 (10.0)	0.0698 (12.0)	0.5628 (8.0)	0.3368 (6.0)
EFCM-GS1	0.5236 (8.0)	0.2327 (7.0)	0.5043 (11.0)	0.1135 (6.0)	**0.6984** (1.0)	**0.4365** (1.0)
EFCM-LS2	0.6180 (5.0)	0.2858 (5.0)	0.5865 (6.0)	0.0792 (9.0)	0.4628 (11.0)	0.0269 (11.0)
EFCM-LS1	0.5107 (9.0)	0.2319 (8.0)	0.6047 (5.0)	0.1353 (3.0)	0.6417 (2.0)	0.3558 (2.0)
	WDBC		WFRN		Wine	
EFCM-2	0.5677 (5.0)	0.6895 (6.0)	0.4972 (10.0)	0.1581 (10.0)	0.5550 (9.0)	0.5953 (8.0)
EFCM-1	0.8032 (2.0)	0.7551 (4.0)	0.5527 (5.0)	0.1591 (9.0)	0.7601 (4.0)	0.8804 (3.0)
EFCM-M	0.5314 (12.0)	0.0100 (12.0)	**0.5893** (1.0)	0.1451 (11.0)	**0.9765** (1.0)	**0.9651** (1.0)
EFCM-Mk	0.5451 (7.0)	0.1578 (10.0)	0.5381 (6.0)	0.2165 (4.0)	0.5014 (12.0)	0.2619 (12.0)
EFCM-GP2	0.5318 (11.0)	0.6954 (5.0)	0.5297 (8.0)	0.1608 (8.0)	0.6149 (7.0)	0.6316 (7.0)
EFCM-GP1	**0.8207** (1.0)	0.7736 (3.0)	0.5252 (9.0)	0.2777 (3.0)	0.7730 (3.0)	0.8804 (3.0)
EFCM-LP2	0.5342 (9.0)	0.4513 (8.0)	0.5548 (3.0)	0.3419 (2.0)	0.6505 (6.0)	0.5114 (10.0)
EFCM-LP1	0.7963 (3.0)	**0.7925** (1.0)	0.5530 (4.0)	**0.3543** (1.0)	0.7905 (2.0)	0.8804 (3.0)
EFCM-GS2	0.7121 (4.0)	0.7802 (2.0)	0.4847 (12.0)	0.1887 (5.0)	0.5147 (11.0)	0.5083 (11.0)
EFCM-GS1	0.5620 (6.0)	0.5057 (7.0)	0.5613 (2.0)	0.1814 (7.0)	0.5869 (8.0)	0.7185 (6.0)
EFCM-LS2	0.5332 (10.0)	0.1390 (11.0)	0.4850 (11.0)	0.1864 (6.0)	0.5213 (10.0)	0.5188 (9.0)
EFCM-LS1	0.5388 (8.0)	0.4292 (9.0)	0.5369 (7.0)	0.0717 (12.0)	0.6601 (5.0)	0.8185 (5.0)

Source: Author (2022)

Table 29 – Results on real data for the best solution according to the mean of HUL and ARI.

	Automobile		Balance Scale		Haberman	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5653 (9.0)	0.1162 (7.0)	0.4495 (5.0)	0.1321 (3.0)	0.5941 (9.0)	-0.0026 (11.0)
(std)	(0.0033)	(0.0379)	(0.0053)	(0.0969)	(0.0052)	(0.0001)
EFCM-1	**0.7260** (1.0)	**0.1727** (1.0)	0.4299 (10.5)	0.0000 (10.0)	0.5531 (10.0)	-0.0006 (9.0)
(std)	(0.0070)	(0.0348)	(0.0000)	(0.0000)	(0.0555)	(0.0006)
EFCM-M	0.6769 (4.0)	0.0796 (11.0)	0.4475 (7.0)	0.1278 (5.0)	0.6009 (7.0)	0.0173 (6.0)
(std)	(0.0496)	(0.0288)	(0.0058)	(0.1006)	(0.0044)	(0.0321)
EFCM-Mk	0.4930 (12.0)	0.0203 (12.0)	**0.5820** (1.0)	0.1245 (6.0)	**0.6086** (1.0)	0.1596 (3.0)
(std)	(0.0229)	(0.0168)	(0.0207)	(0.0438)	(0.0000)	(0.0000)
EFCM-GP2	0.6931 (3.0)	0.1221 (6.0)	0.5037 (2.0)	0.1138 (7.0)	0.5998 (8.0)	-0.0027 (12.0)
(std)	(0.0229)	(0.0311)	(0.0150)	(0.0115)	(0.0020)	(0.0001)
EFCM-GP1	0.7192 (2.0)	0.1652 (2.0)	0.4299 (10.5)	0.0000 (10.0)	0.5530 (11.0)	-0.0012 (10.0)
(std)	(0.0074)	(0.0268)	(0.0000)	(0.0000)	(0.0556)	(0.0015)
EFCM-LP2	0.5821 (8.0)	0.0860 (10.0)	0.4474 (8.0)	0.1283 (4.0)	0.6084 (2.0)	0.1456 (4.0)
(std)	(0.0127)	(0.0238)	(0.0061)	(0.0844)	(0.0000)	(0.0000)
EFCM-LP1	0.6108 (7.0)	0.1051 (9.0)	0.4989 (3.0)	-0.0093 (12.0)	0.6045 (5.0)	0.1620 (2.0)
(std)	(0.0150)	(0.0260)	(0.0021)	(0.0044)	(0.0006)	(0.0193)
EFCM-GS2	0.5610 (10.0)	0.1111 (8.0)	0.4476 (6.0)	0.1420 (2.0)	0.5290 (12.0)	0.0068 (7.0)
(std)	(0.0079)	(0.0250)	(0.0021)	(0.0509)	(0.0375)	(0.0378)
EFCM-GS1	0.6579 (5.0)	0.1509 (3.0)	0.4299 (10.5)	0.0000 (10.0)	0.6058 (4.0)	-0.0005 (8.0)
(std)	(0.0226)	(0.0427)	(0.0000)	(0.0000)	(0.0038)	(0.0006)
EFCM-LS2	0.5385 (11.0)	0.1229 (5.0)	0.4598 (4.0)	**0.1469** (1.0)	0.6072 (3.0)	0.1001 (5.0)
(std)	(0.0077)	(0.0183)	(0.0113)	(0.0658)	(0.0000)	(0.0000)
EFCM-LS1	0.6371 (6.0)	0.1346 (4.0)	0.4299 (10.5)	0.0028 (8.0)	0.6016 (6.0)	**0.1775** (1.0)
(std)	(0.0260)	(0.0414)	(0.0000)	(0.0073)	(0.0006)	(0.0166)
	Heart		Image Segmentation		Ionosphere	
EFCM-2	0.5431 (6.0)	0.3487 (8.0)	0.8013 (6.0)	0.4892 (5.0)	0.5368 (10.0)	0.1573 (5.0)
(std)	(0.0046)	(0.0000)	(0.0022)	(0.0054)	(0.0008)	(0.0162)
EFCM-1	**0.6891** (1.0)	**0.4261** (1.0)	0.8614 (2.0)	**0.5219** (1.0)	0.5377 (7.0)	0.0936 (8.0)
(std)	(0.0191)	(0.0435)	(0.0053)	(0.0158)	(0.0000)	(0.0000)
EFCM-M	0.5055 (11.0)	0.0412 (11.0)	0.1425 (12.0)	0.0024 (12.0)	0.5401 (4.0)	0.0290 (12.0)
(std)	(0.0023)	(0.0580)	(0.0000)	(0.0013)	(0.0008)	(0.0412)
EFCM-Mk	0.5043 (12.0)	-0.0024 (12.0)	0.2115 (11.0)	0.0152 (11.0)	**0.5675** (1.0)	0.0514 (10.0)
(std)	(0.0008)	(0.0024)	(0.0016)	(0.0036)	(0.0020)	(0.0442)
EFCM-GP2	0.5123 (8.0)	0.3488 (7.0)	0.8523 (3.0)	0.4642 (8.0)	0.5378 (6.0)	0.1588 (3.5)
(std)	(0.0024)	(0.0009)	(0.0226)	(0.0504)	(0.0002)	(0.0000)
EFCM-GP1	0.5587 (5.0)	0.1247 (10.0)	**0.8639** (1.0)	0.5159 (4.0)	0.5418 (3.0)	0.1045 (7.0)
(std)	(0.0387)	(0.0711)	(0.0053)	(0.0199)	(0.0000)	(0.0000)
EFCM-LP2	0.5278 (7.0)	0.4065 (3.0)	0.7611 (10.0)	0.3729 (10.0)	0.5363 (11.0)	0.1353 (6.0)
(std)	(0.0056)	(0.0432)	(0.0164)	(0.0299)	(0.0005)	(0.0039)
EFCM-LP1	0.6102 (4.0)	0.3247 (9.0)	0.7647 (9.0)	0.4014 (9.0)	0.5362 (12.0)	0.0894 (9.0)
(std)	(0.0553)	(0.1654)	(0.0291)	(0.0540)	(0.0003)	(0.0172)
EFCM-GS2	0.5122 (9.0)	0.3502 (6.0)	0.7927 (8.0)	0.4796 (7.0)	0.5380 (5.0)	0.1588 (3.5)
(std)	(0.0020)	(0.0101)	(0.0037)	(0.0162)	(0.0001)	(0.0000)
EFCM-GS1	0.6631 (2.0)	0.4094 (2.0)	0.8283 (4.0)	0.5170 (3.0)	0.5376 (8.0)	0.0307 (11.0)
(std)	(0.0498)	(0.0978)	(0.0108)	(0.0162)	(0.0019)	(0.0152)
EFCM-LS2	0.5083 (10.0)	0.3757 (5.0)	0.7938 (7.0)	0.4869 (6.0)	0.5375 (9.0)	0.1634 (2.0)
(std)	(0.0024)	(0.0000)	(0.0004)	(0.0008)	(0.0005)	(0.0000)
EFCM-LS1	0.6615 (3.0)	0.4061 (4.0)	0.8235 (5.0)	0.5196 (2.0)	0.5508 (2.0)	**0.2092** (1.0)
(std)	(0.0524)	(0.1022)	(0.0055)	(0.0201)	(0.0000)	(0.0000)

Source: Author (2022)

Table 29 – Results on real data for the best solution according to the mean of HUL and ARI.

	Iris plants		Mnist		Thyroid	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.7538 (11.0)	0.6199 (10.5)	0.5037 (7.5)	0.9554 (4.0)	0.5995 (11.0)	0.3389 (7.0)
(std)	(0.0005)	(0.0000)	(0.0013)	(0.0069)	(0.0010)	(0.0180)
EFCM-1	0.8215 (6.0)	0.6199 (10.5)	0.5255 (2.0)	0.9522 (5.0)	0.7497 (2.0)	0.5024 (3.0)
(std)	(0.0454)	(0.0682)	(0.0176)	(0.0043)	(0.1288)	(0.2553)
EFCM-M	0.8044 (7.0)	**0.8933** (1.0)	0.5222 (3.5)	0.9557 (2.5)	0.6476 (8.0)	0.2338 (9.0)
(std)	(0.0379)	(0.0304)	(0.0139)	(0.0069)	(0.0189)	(0.1326)
EFCM-Mk	0.5870 (12.0)	0.2824 (12.0)	0.5222 (3.5)	0.9557 (2.5)	0.7113 (4.0)	0.1494 (10.0)
(std)	(0.0000)	(0.0000)	(0.0139)	(0.0069)	(0.0157)	(0.1302)
EFCM-GP2	0.8620 (3.0)	0.7618 (3.0)	0.5037 (7.5)	0.9465 (6.0)	0.6879 (6.0)	**0.6246** (1.0)
(std)	(0.0504)	(0.1737)	(0.0010)	(0.0956)	(0.0274)	(0.1881)
EFCM-GP1	**0.8796** (1.0)	0.7557 (4.0)	0.5042 (6.0)	0.9159 (7.0)	**0.7740** (1.0)	0.5467 (2.0)
(std)	(0.0923)	(0.1756)	(0.0016)	(0.1562)	(0.1032)	(0.2067)
EFCM-LP2	0.7813 (8.0)	0.6882 (5.0)	**0.5276** (1.0)	0.1414 (9.0)	0.7074 (5.0)	0.3346 (8.0)
(std)	(0.0011)	(0.0000)	(0.0142)	(0.1878)	(0.0129)	(0.2908)
EFCM-LP1	0.8780 (2.0)	0.7649 (2.0)	0.5045 (5.0)	0.0295 (10.0)	0.7315 (3.0)	0.4173 (5.0)
(std)	(0.0375)	(0.1006)	(0.0012)	(0.0822)	(0.0069)	(0.0414)
EFCM-GS2	0.7613 (9.0)	0.6303 (8.5)	0.5022 (11.5)	**0.9561** (1.0)	0.6840 (7.0)	0.3469 (6.0)
(std)	(0.0009)	(0.0000)	(0.0000)	(0.0000)	(0.0183)	(0.0867)
EFCM-GS1	0.8303 (5.0)	0.6337 (7.0)	0.5023 (9.5)	0.8315 (8.0)	0.5818 (12.0)	0.1114 (12.0)
(std)	(0.0411)	(0.0600)	(0.0002)	(0.2423)	(0.0525)	(0.0918)
EFCM-LS2	0.7567 (10.0)	0.6303 (8.5)	0.5022 (11.5)	0.0106 (12.0)	0.6214 (9.0)	0.4365 (4.0)
(std)	(0.0006)	(0.0000)	(0.0000)	(0.0423)	(0.0002)	(0.0134)
EFCM-LS1	0.8322 (4.0)	0.6364 (6.0)	0.5023 (9.5)	0.0270 (11.0)	0.6197 (10.0)	0.1386 (11.0)
(std)	(0.0403)	(0.0591)	(0.0000)	(0.0222)	(0.0013)	(0.0048)
	UKM		Vehicle		Vertebral column	
EFCM-2	0.4874 (9.0)	0.1603 (8.0)	0.5669 (7.0)	0.0729 (11.0)	0.5748 (6.0)	0.3000 (9.0)
(std)	(0.0072)	(0.0115)	(0.0004)	(0.0003)	(0.0001)	(0.0024)
EFCM-1	0.6003 (5.0)	0.1806 (7.0)	0.6420 (2.0)	0.0870 (7.0)	0.6005 (5.0)	0.3436 (4.0)
(std)	(0.0154)	(0.0322)	(0.0045)	(0.0160)	(0.0019)	(0.0177)
EFCM-M	0.5854 (6.0)	0.1201 (11.0)	0.3967 (12.0)	0.0738 (10.0)	0.3775 (12.0)	0.2170 (10.0)
(std)	(0.0463)	(0.0787)	(0.1493)	(0.0419)	(0.0032)	(0.0971)
EFCM-Mk	0.4223 (12.0)	0.0161 (12.0)	0.4403 (11.0)	0.0987 (6.0)	0.4630 (10.0)	0.0079 (12.0)
(std)	(0.0051)	(0.0027)	(0.0267)	(0.0391)	(0.0147)	(0.0031)
EFCM-GP2	0.6538 (2.0)	0.2640 (3.0)	0.5530 (8.0)	0.0755 (9.0)	0.5660 (7.0)	0.3320 (8.0)
(std)	(0.0656)	(0.1032)	(0.0033)	(0.0004)	(0.0055)	(0.0019)
EFCM-GP1	**0.7036** (1.0)	0.2747 (2.0)	0.6352 (3.0)	0.1186 (4.0)	0.6118 (4.0)	0.3477 (3.0)
(std)	(0.0409)	(0.0984)	(0.0141)	(0.0201)	(0.0022)	(0.0123)
EFCM-LP2	0.4730 (11.0)	0.1420 (10.0)	0.5725 (6.0)	0.1291 (3.0)	0.5650 (8.0)	0.3433 (5.0)
(std)	(0.0107)	(0.0222)	(0.0037)	(0.0055)	(0.0096)	(0.0176)
EFCM-LP1	0.6337 (3.0)	0.2449 (4.0)	**0.6685** (1.0)	**0.1423** (1.0)	0.6218 (3.0)	0.3396 (6.0)
(std)	(0.0235)	(0.0524)	(0.0046)	(0.0109)	(0.0018)	(0.0319)
EFCM-GS2	0.4780 (10.0)	0.1516 (9.0)	0.5417 (9.0)	0.0694 (12.0)	0.5648 (9.0)	0.3342 (7.0)
(std)	(0.0114)	(0.0193)	(0.0009)	(0.0012)	(0.0052)	(0.0023)
EFCM-GS1	0.5102 (7.0)	0.2096 (5.0)	0.5186 (10.0)	0.1023 (5.0)	**0.6634** (1.0)	0.3915 (2.0)
(std)	(0.0113)	(0.0302)	(0.0099)	(0.0197)	(0.0138)	(0.0265)
EFCM-LS2	0.6014 (4.0)	**0.3001** (1.0)	0.5824 (5.0)	0.0803 (8.0)	0.4564 (11.0)	0.1055 (11.0)
(std)	(0.0131)	(0.0447)	(0.0053)	(0.0051)	(0.0131)	(0.1145)
EFCM-LS1	0.5005 (8.0)	0.2065 (6.0)	0.5920 (4.0)	0.1382 (2.0)	0.6281 (2.0)	**0.4096** (1.0)
(std)	(0.0113)	(0.0359)	(0.0211)	(0.0087)	(0.0089)	(0.0530)

Source: Author (2020).

Table 29 – Results on real data for the best solution according to the mean of HUL and ARI..

	WDBC		WFRN		Wine	
Algorithms	HUL	ARI	HUL	ARI	HUL	ARI
EFCM-2	0.5607 (5.0)	0.6895 (6.0)	0.4958 (10.0)	0.1635 (9.0)	0.5546 (7.0)	0.5978 (7.0)
(std)	(0.0082)	(0.0000)	(0.0016)	(0.0097)	(0.0018)	(0.0083)
EFCM-1	0.8032 (2.0)	0.7551 (3.0)	0.5543 (3.5)	0.1755 (8.0)	0.7589 (3.0)	0.8772 (3.0)
(std)	(0.0000)	(0.0000)	(0.0067)	(0.0372)	(0.0126)	(0.0323)
EFCM-M	0.5312 (12.0)	0.0239 (12.0)	**0.5678** (1.0)	0.0987 (11.0)	0.4891 (11.0)	0.2878 (11.0)
(std)	(0.0009)	(0.0300)	(0.0164)	(0.0299)	(0.2346)	(0.3090)
EFCM-Mk	0.5396 (7.0)	0.0944 (11.0)	0.5381 (7.5)	0.2165 (4.0)	0.4298 (12.0)	0.1603 (12.0)
(std)	(0.0035)	(0.0936)	(0.0001)	(0.0002)	(0.0316)	(0.0740)
EFCM-GP2	0.5422 (6.0)	0.6896 (5.0)	0.5381 (7.5)	0.1222 (10.0)	0.6160 (6.0)	0.6291 (5.0)
(std)	(0.0028)	(0.0006)	(0.0137)	(0.0431)	(0.0017)	(0.0081)
EFCM-GP1	**0.8207** (1.0)	0.7736 (2.0)	0.5088 (9.0)	0.2267 (3.0)	0.7718 (2.0)	0.8773 (2.0)
(std)	(0.0000)	(0.0000)	(0.0068)	(0.0296)	(0.0122)	(0.0313)
EFCM-LP2	0.5342 (10.0)	0.4513 (7.0)	0.5543 (3.5)	**0.3415** (1.0)	0.6510 (4.0)	0.5162 (10.0)
(std)	(0.0000)	(0.0000)	(0.0056)	(0.0073)	(0.0005)	(0.0075)
EFCM-LP1	0.7963 (3.0)	**0.7925** (1.0)	0.5513 (5.0)	0.3414 (2.0)	**0.7875** (1.0)	**0.8804** (1.0)
(std)	(0.0000)	(0.0000)	(0.0051)	(0.0271)	(0.0005)	(0.0000)
EFCM-GS2	0.6381 (4.0)	0.7341 (4.0)	0.4833 (12.0)	0.1888 (6.0)	0.5145 (10.0)	0.5293 (9.0)
(std)	(0.0841)	(0.0867)	(0.0071)	(0.0068)	(0.0002)	(0.0082)
EFCM-GS1	0.5379 (9.0)	0.3923 (9.0)	0.5603 (2.0)	0.1769 (7.0)	0.5499 (8.0)	0.6107 (6.0)
(std)	(0.0087)	(0.1505)	(0.0030)	(0.0172)	(0.0249)	(0.0608)
EFCM-LS2	0.5332 (11.0)	0.1390 (10.0)	0.4846 (11.0)	0.2028 (5.0)	0.5210 (9.0)	0.5328 (8.0)
(std)	(0.0000)	(0.0000)	(0.0004)	(0.0106)	(0.0007)	(0.0158)
EFCM-LS1	0.5388 (8.0)	0.4292 (8.0)	0.5391 (6.0)	0.0817 (12.0)	0.6417 (5.0)	0.7573 (4.0)
(std)	(0.0000)	(0.0000)	(0.0089)	(0.0419)	(0.0201)	(0.0838)

Source: Author (2022)

APPENDIX C – RESULTS FOR THE EXPERIMENTS ON SYNTHETIC INTERVAL-VALUED DATASETS.

Table 30 – Mean and standard deviation for HUL varying the number of outliers.

Algorithms	0%		10%		20%	
	mean	(std)	mean	(std)	mean	(std)
TrFCMdd-ID	0.6418	(0.0102)	0.6244	(0.0401)	0.5442	(0.0529)
ExpFCMd-ID	0.6218	(0.0167)	0.6335	(0.0160)	0.6403	(0.0158)
AIFCM-G2	**0.6706**	(0.0050)	0.6250	(0.0544)	0.5241	(0.0531)
AIFCM-G1	0.6643	(0.0060)	**0.6703**	(0.0074)	0.6763	(0.0116)
AIFCM-L2	0.6703	(0.0057)	0.6218	(0.0539)	0.5237	(0.0532)
AIFCM-L1	0.6647	(0.0061)	0.6699	(0.0078)	**0.6770**	(0.0111)
EIFCM-GJ2	0.6576	(0.0064)	0.6074	(0.0107)	0.5734	(0.0125)
EIFCM-GJ1	0.6683	(0.0069)	0.6673	(0.0134)	0.6646	(0.0193)
EIFCM-LJ2	0.6538	(0.0065)	0.5775	(0.0304)	0.5403	(0.0243)
EIFCM-LJ1	0.6672	(0.0061)	0.6583	(0.0220)	0.6571	(0.0235)

Source: Author (2022)

Table 31 – Mean and standard deviation for ARI varying the number of outliers.

Algorithms	0%		10%		20%	
	mean	(std)	mean	(std)	mean	(std)
TrFCMdd-ID	0.9645	(0.0597)	0.7405	(0.2784)	0.1936	(0.3047)
ExpFCMd-ID	0.9290	(0.0669)	0.9077	(0.1105)	0.9188	(0.0990)
AIFCM-G2	**1.0000**	(0.0000)	0.6691	(0.3096)	0.1004	(0.2628)
AIFCM-G1	0.9842	(0.0413)	**0.9842**	(0.0413)	0.9450	(0.0719)
AIFCM-L2	0.9960	(0.0179)	0.6536	(0.3138)	0.0999	(0.2630)
AIFCM-L1	0.9920	(0.0246)	0.9802	(0.0435)	**0.9530**	(0.0732)
EIFCM-GJ2	**1.0000**	(0.0000)	0.6796	(0.1397)	0.2157	(0.1690)
EIFCM-GJ1	0.9762	(0.0452)	0.9601	(0.0482)	0.9210	(0.0617)
EIFCM-LJ2	**1.0000**	(0.0000)	0.5134	(0.1988)	0.2731	(0.1521)
EIFCM-LJ1	0.9802	(0.0435)	0.9525	(0.0594)	0.9248	(0.0596)

Source: Author (2022)

Table 32 – Clustering results on real interval-valued data.

	HUL	ARI	HUL	ARI	HUL	ARI
Algorithms	Car models		City Temperature		Freshwater fish species	
TrFCMdd-ID	0.6937 (10.0)	0.4773 (9.0)	0.6766 (9.0)	0.4786 (5.0)	0.6171 (9.0)	0.2226 (8.0)
ExpFCMd-ID	0.7094 (7.0)	0.2836 (10.0)	0.6433 (10.0)	0.2817 (10.0)	0.6284 (8.0)	0.1852 (9.0)
AIFCM-G2	0.7108 (6.0)	0.4998 (7.5)	0.7025 (6.0)	**0.5458** (2.0)	0.6413 (7.0)	0.3473 (5.5)
AIFCM-G1	0.7183 (5.0)	0.5623 (4.0)	0.6820 (8.0)	0.4162 (7.5)	0.6008 (10.0)	0.1417 (10.0)
AIFCM-L2	0.7016 (9.0)	0.5257 (6.0)	0.7126 (5.0)	**0.5458** (2.0)	0.6704 (5.0)	0.2276 (7.0)
AIFCM-L1	0.7072 (8.0)	0.4998 (7.5)	0.6854 (7.0)	0.4001 (9.0)	0.6699 (6.0)	0.3473 (5.5)
EIFCM-GJ2	0.7362 (4.0)	0.5589 (5.0)	0.8153 (2.0)	0.4928 (4.0)	0.7218 (2.0)	**0.5210** (2.0)
EIFCM-GJ1	0.7457 (3.0)	**0.6802** (1.0)	0.7459 (4.0)	0.4162 (7.5)	**0.7454** (1.0)	**0.5210** (2.0)
EIFCM-LJ2	**0.7537** (1.0)	0.5720 (3.0)	**0.8275** (1.0)	**0.5458** (2.0)	0.7044 (4.0)	0.3663 (4.0)
EIFCM-LJ1	0.7535 (2.0)	0.6142 (2.0)	0.7466 (3.0)	0.4439 (6.0)	0.7136 (3.0)	**0.5210** (2.0)
	Fungi		Horse		Ichino	
TrFCMdd-ID	0.5886 (9.0)	0.2962 (8.0)	0.6028 (9.0)	0.1842 (6.0)	0.6682 (7.0)	0.4444 (5.0)
ExpFCMd-ID	0.5304 (10.0)	0.1934 (10.0)	0.6078 (8.0)	0.1379 (8.0)	0.6462 (10.0)	0.2222 (9.5)
AIFCM-G2	0.7138 (2.0)	0.8291 (2.0)	0.6269 (7.0)	0.0559 (10.0)	0.6931 (5.0)	0.4444 (5.0)
AIFCM-G1	0.6949 (4.0)	0.7773 (3.0)	0.6297 (6.0)	0.3134 (3.0)	0.6546 (9.0)	0.3396 (7.5)
AIFCM-L2	**0.7184** (1.0)	**0.8837** (1.0)	0.6309 (5.0)	0.1417 (7.0)	0.7362 (3.0)	0.3396 (7.5)
AIFCM-L1	0.7031 (3.0)	0.7361 (4.0)	0.6719 (2.0)	0.2276 (5.0)	0.6581 (8.0)	0.2222 (9.5)
EIFCM-GJ2	0.6286 (6.0)	0.3711 (6.0)	0.5759 (10.0)	0.3134 (3.0)	0.7447 (2.0)	0.6038 (2.0)
EIFCM-GJ1	0.6146 (8.0)	0.3064 (7.0)	**0.7198** (1.0)	**0.4558** (1.0)	0.6789 (6.0)	0.4444 (5.0)
EIFCM-LJ2	0.6326 (5.0)	0.3754 (5.0)	0.6580 (4.0)	0.3134 (3.0)	0.7316 (4.0)	0.5882 (3.0)
EIFCM-LJ1	0.6275 (7.0)	0.2735 (9.0)	0.6624 (3.0)	0.1157 (9.0)	**0.7615** (1.0)	**1.0000** (1.0)

Source: Author (2022)

Table 32 – Clustering results on real interval-valued data.

	HUL	ARI	HUL	ARI
Algorithms	Iris		Wine	
TrFCMdd-ID	0.7327 (9.0)	1.0000 (5.5)	0.5045 (7.0)	0.0448 (4.0)
ExpFCMd-ID	0.7078 (10.0)	**1.0000** (5.5)	0.4985 (10.0)	-0.0065 (7.5)
AIFCM-G2	0.7782 (3.0)	**1.0000** (5.5)	**0.5169** (1.0)	0.1282 (2.0)
AIFCM-G1	0.7595 (6.0)	**1.0000** (5.5)	0.5117 (4.0)	**0.1297** (1.0)
AIFCM-L2	0.7716 (4.0)	**1.0000** (5.5)	0.5031 (8.0)	-0.0292 (9.0)
AIFCM-L1	0.7657 (5.0)	**1.0000** (5.5)	0.5093 (6.0)	-0.0065 (7.5)
EIFCM-GJ2	0.7389 (7.0)	**1.0000** (5.5)	0.5154 (2.0)	0.0836 (3.0)
EIFCM-GJ1	**0.7845** (1.0)	**1.0000** (5.5)	0.5119 (3.0)	0.0165 (5.5)
EIFCM-LJ2	0.7341 (8.0)	**1.0000** (5.5)	0.5096 (5.0)	-0.0299 (10.0)
EIFCM-LJ1	0.7824 (2.0)	**1.0000** (5.5)	0.5016 (9.0)	0.0165 (5.5)

Source: Author (2022)

Table 33 – Mean and standard deviation according to the 50 executions of the algorithms.

	HUL	ARI	HUL	ARI	HUL	ARI
Algorithms	Car models		City Temperature		Freshwater fish species	
TrFCMdd-ID	0.6483 (9.0)	0.4254 (9.0)	0.6745 (9.0)	0.4588 (4.0)	0.5778 (9.0)	0.1827 (9.0)
(std)	(0.0336)	(0.1386)	(0.0275)	(0.1014)	(0.0434)	(0.1863)
ExpFCMd-ID	0.3780 (10.0)	0.1244 (10.0)	0.6361 (10.0)	0.4268 (5.0)	0.5332 (10.0)	0.1025 (10.0)
(std)	(0.1334)	(0.1001)	(0.0750)	(0.1397)	(0.0537)	(0.1204)
AIFCM-G2	0.7108 (5.0)	0.4998 (7.0)	0.7025 (6.0)	**0.5458** (1.5)	0.6473 (7.0)	0.3325 (3.0)
(std)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0346)	(0.0893)
AIFCM-G1	0.7044 (6.0)	0.5050 (6.0)	0.6750 (8.0)	0.4264 (6.0)	0.5986 (8.0)	0.2542 (7.0)
(std)	(0.0305)	(0.0950)	(0.0139)	(0.0697)	(0.0359)	(0.1678)
AIFCM-L2	0.7000 (7.0)	0.5263 (5.0)	0.7126 (5.0)	**0.5458** (1.5)	0.6834 (3.0)	0.1921 (8.0)
(std)	(0.0042)	(0.0028)	(0.0000)	(0.0000)	(0.0082)	(0.0305)
AIFCM-L1	0.6882 (8.0)	0.4990 (8.0)	0.6794 (7.0)	0.4112 (7.0)	0.6711 (5.0)	0.2766 (6.0)
(std)	(0.0320)	(0.0877)	(0.0147)	(0.0547)	(0.0296)	(0.1459)
EIFCM-GJ2	0.7362 (4.0)	0.5589 (4.0)	0.8074 (2.0)	0.4080 (8.0)	**0.7210** (1.0)	**0.5172** (1.0)
(std)	(0.0000)	(0.0000)	(0.0086)	(0.0929)	(0.0083)	(0.0271)
EIFCM-GJ1	0.7379 (3.0)	**0.6425** (1.0)	0.7273 (4.0)	0.3967 (10.0)	0.6592 (6.0)	0.4463 (2.0)
(std)	(0.0244)	(0.0660)	(0.0168)	(0.0477)	(0.1024)	(0.1373)
EIFCM-LJ2	0.7443 (2.0)	0.6131 (2.0)	**0.8206** (1.0)	0.4932 (3.0)	0.6829 (4.0)	0.2772 (5.0)
(std)	(0.0079)	(0.0269)	(0.0101)	(0.0775)	(0.0284)	(0.1077)
EIFCM-LJ1	**0.7526** (1.0)	0.6073 (3.0)	0.7293 (3.0)	0.4039 (9.0)	0.7032 (2.0)	0.3300 (4.0)
(std)	(0.0043)	(0.0276)	(0.0180)	(0.0451)	(0.0064)	(0.1341)
	Fungi		Horse		Ichino	
TrFCMdd-ID	0.5490 (9.0)	0.3343 (8.0)	0.5596 (8.0)	0.1332 (9.0)	0.6435 (6.0)	0.3191 (9.0)
(std)	(0.0536)	(0.1158)	(0.0452)	(0.1199)	(0.0136)	(0.2043)
ExpFCMd-ID	0.4226 (10.0)	0.0885 (10.0)	0.4805 (10.0)	-0.0357 (10.0)	0.5836 (10.0)	0.2556 (10.0)
(std)	(0.0263)	(0.0267)	(0.0807)	(0.1153)	(0.0805)	(0.1699)
AIFCM-G2	0.7138 (2.0)	0.8291 (2.0)	0.6231 (3.0)	0.2517 (5.0)	0.6422 (7.0)	0.3764 (7.0)
(std)	(0.0000)	(0.0000)	(0.0039)	(0.1589)	(0.0383)	(0.2213)
AIFCM-G1	0.5955 (5.0)	0.4670 (3.0)	0.5769 (7.0)	0.1972 (7.0)	0.6487 (5.0)	0.3514 (8.0)
(std)	(0.0577)	(0.1876)	(0.0406)	(0.1526)	(0.0163)	(0.2441)
AIFCM-L2	**0.7184** (1.0)	**0.8837** (1.0)	0.6471 (2.0)	0.3034 (3.0)	0.6985 (3.0)	0.4409 (5.0)
(std)	(0.0000)	(0.0000)	(0.0102)	(0.1018)	(0.0352)	(0.1188)
AIFCM-L1	0.6190 (3.0)	0.4340 (4.0)	0.5877 (6.0)	0.2261 (6.0)	0.6574 (4.0)	0.3906 (6.0)
(std)	(0.0516)	(0.1810)	(0.0439)	(0.1480)	(0.0173)	(0.2893)
EIFCM-GJ2	0.5767 (7.0)	0.4082 (5.0)	0.5984 (5.0)	**0.3989** (1.0)	**0.7345** (1.0)	**0.6025** (1.0)
(std)	(0.0106)	(0.0077)	(0.0134)	(0.0512)	(0.0284)	(0.0043)
EIFCM-GJ1	0.5762 (8.0)	0.3050 (9.0)	0.5460 (9.0)	0.1704 (8.0)	0.6069 (9.0)	0.4780 (4.0)
(std)	(0.0536)	(0.0982)	(0.1441)	(0.2336)	(0.1049)	(0.1196)
EIFCM-LJ2	0.6172 (4.0)	0.3448 (7.0)	**0.6615** (1.0)	0.2748 (4.0)	0.7316 (2.0)	0.5882 (2.0)
(std)	(0.0167)	(0.0487)	(0.0220)	(0.1075)	(0.0005)	(0.0000)
EIFCM-LJ1	0.5946 (6.0)	0.3521 (6.0)	0.6024 (4.0)	0.3753 (2.0)	0.6347 (8.0)	0.5783 (3.0)
(std)	(0.0373)	(0.0605)	(0.0784)	(0.1190)	(0.0867)	(0.3014)

Source: Author (2022)

Table 33 – Mean and standard deviation according to the 50 executions of the algorithms.

	HUL	ARI	HUL	ARI
Algorithms	Iris		Wine	
TrFCMdd-ID	0.6212 (9.0)	0.6719 (9.0)	0.5043 (10.0)	0.0013 (7.0)
(std)	(0.1098)	(0.2611)	(0.0041)	(0.0370)
ExpFCMd-ID	0.5148 (10.0)	0.4918 (10.0)	0.5141 (5.0)	0.0672 (5.0)
(std)	(0.1042)	(0.2915)	(0.0082)	(0.0661)
AIFCM-G2	**0.7781** (1.0)	**1.0000** (2.5)	0.5169 (2.0)	0.1282 (3.0)
(std)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
AIFCM-G1	0.7213 (6.0)	0.8748 (6.0)	0.5142 (4.0)	0.1284 (2.0)
(std)	(0.0727)	(0.2381)	(0.0057)	(0.0846)
AIFCM-L2	0.7716 (2.0)	**1.0000** (2.5)	0.5050 (9.0)	-0.0223 (9.0)
(std)	(0.0000)	(0.0000)	(0.0056)	(0.0150)
AIFCM-L1	0.7361 (4.0)	0.9116 (5.0)	0.5128 (6.0)	0.0251 (6.0)
(std)	(0.0638)	(0.2053)	(0.0070)	(0.0822)
EIFCM-GJ2	0.7389 (3.0)	**1.0000** (2.5)	0.5154 (3.0)	0.0836 (4.0)
(std)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
EIFCM-GJ1	0.6979 (8.0)	0.8404 (7.0)	**0.5264** (1.0)	**0.2100** (1.0)
(std)	(0.1477)	(0.2494)	(0.0039)	(0.1377)
EIFCM-LJ2	0.7341 (5.0)	**1.0000** (2.5)	0.5082 (8.0)	-0.0298 (10.0)
(std)	(0.0000)	(0.0000)	(0.0026)	(0.0003)
EIFCM-LJ1	0.6983 (7.0)	0.8169 (8.0)	0.5107 (7.0)	-0.0036 (8.0)
(std)	(0.1451)	(0.2496)	(0.0029)	(0.0164)

Source: Author (2022)