Pós-Graduação em Ciência da Computação

Sara Inés Rizo Rodríguez

A Fuzzy Partitional Clustering algorithm with Adaptive Euclidean distance and Entropy Regularization

Federal University of Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

RECIFE
2018

Sara Inés Rizo Rodríguez

A FUZZY PARTITIONAL CLUSTERING ALGORITHM WITH ADAPTIVE EUCLIDEAN DISTANCE AND ENTROPY REGULARIZATION

A M.Sc. Dissertation presented to the Center for Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Advisor: Francisco de Assis Tenorio de Carvalho

RECIFE
2018

Catalogação na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB 4-1758

R627f Rizo Rodríguez, Sara Inés.
A fuzzy partitional clustering algorithm with adaptative euclidean distance and entropy regularization / Sara Inés Rizo Rodríguez . – 2018.
108 f.: fig., tab.

Orientador: Francisco de Assis Tenório de Carvalho.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação. Recife, 2018.
Inclui referências e apêndice.

1. Mineração de dados. 2. Agrupamento difuso. I. Carvalho, Francisco de Assis Tenório de. (orientador). II. Título

006.312 CDD (22. ed.) UFPE-MEI 2018-74

Sara Inés Rizo Rodríguez

A Fuzzy Partitional Clustering algorithm with Adaptive Euclidean distance and Entropy Regularization

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 21/02/2018

BANCA EXAMINADORA

____

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

____

Prof. Dr. Regivan Hugo Nunes Santiago
Departamento de Informática e Matemática Aplicada/UFRN

____

Prof. Dr. Francisco de Assis Tenório de Carvalho
Centro de Informática / UFPE
(Orientador)

To my family.

ACKNOWLEDGMENT

I would like to thank my husband and my parents for supporting me spiritually throughout writing this thesis and my life in general. I want to express my sincere gratitude to my advisor Prof. Francisco Carvalho for the continuous support of my research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. Thanks to all the professors at informatics center for their excellent classes and help.

"The only man who never makes a mistake is the man who never does anything."

(Theodore Roosevelt)

RESUMO

O agrupamento de dados é uma das questões mais importantes na mineração de dados e na aprendizagem de máquinas. O objetivo principal é descobrir grupos homogêneos nos objetos estudados. A maior dificuldade é que não se tem conhecimento prévio sobre o conjunto de dados. Usualmente as abordagens de agrupamento tradicionais são projetadas para pesquisar grupos em todo o espaço. No entanto, em conjuntos de dados reais de alta dimensão, geralmente existem muitas características irrelevantes para o agrupamento, onde os métodos tradicionais não apresentam bom performance. O agrupamento em subespaços é uma extensão do agrupamento tradicional que permite encontrar grupos em subespaços gerados apenas pelas variáveis relevantes do conjunto de dados. No entanto a maioria desses métodos precisam configurações de parâmetros não triviais e portanto o uso desses métodos em aplicações práticas é dificultado devido a encontrar a parametrização apropriada. Este trabalho propõe um algoritmo de agrupamento particional difuso com regularização de entropia e seleção automática de variáveis. A seleção de variáveis é feita através de distância adaptativa sendo a medida de dissimilaridade a soma das distâncias Euclidiana entre padrões e protótipos para cada variável. A principal vantagem da abordagem proposta sobre os métodos de agrupamento convencionais é a possibilidade do uso de distâncias adaptativas, as quais mudam a cada iteração do algoritmo. Este tipo de medida de dissimilaridade é adequado ao aprendizado dos pesos das variáveis dinamicamente durante o processo de agrupamento, levando a uma melhora do desempenho dos algoritmos. Outra vantagem da abordagem proposta é o uso do termo de regularização da entropia que serve como um fator regulador durante o processo de minimização. O método proposto é um algoritmo iterativo de três passos que fornece uma partição difusa, um representante para cada grupo difuso e aprende um peso de relevância para cada variável em cada grupo. Para isto é minimizada uma função objetivo que inclui uma função de distância multidimensional como medida de dissimilaridade e entropia como o termo de regularização. Os experimentos realizados em conjuntos de dados simulados, do mundo real e em imagens corroboram a utilidade do algoritmo proposto.

Palavras-chaves: Agrupamento difuso. Distância adaptativa. Regularização de máxima entropia. Função objetivo.

ABSTRACT

Data Clustering is one of the most important issues in data mining and machine learning. Clustering is a task of discovering homogeneous groups of the studied objects. Recently, many researchers have a significant interest in developing clustering algorithms. The most problem in clustering is that we do not have prior information knowledge about the given dataset. The traditional clustering approaches are designed for searching clusters in the entire space. However, in high-dimensional real world datasets, there are usually many irrelevant dimensions for clustering, where the traditional clustering methods work often improperly. Subspace clustering is an extension of traditional clustering that enables finding subspace clusters only in relevant dimensions within a data set. However, most subspace clustering methods usually suffer from the issue that their complicated parameter settings are almost troublesome to be determined, and therefore it can be difficult to implement these methods in practical applications. This work proposes a partitioning fuzzy clustering algorithm with entropy regularization and automatic variable selection through adaptive distance where the dissimilarity measure is obtained as the sum of the Euclidean distance between objects and prototypes calculated individually for each variable. The main advantage of the proposed approach to conventional clustering methods is the possibility of using adaptive distances, which change with each iteration of the algorithm. This type of dissimilarity measure is adequate to learn the weights of the variables dynamically during the clustering process, leading to an improvement of the performance of the algorithms. Another advantage of the proposed approach is the use of the entropy regularization term that serves as a regulating factor during the minimization process. The proposed method is an iterative three-step algorithm that provides a fuzzy partition, a representative for each fuzzy cluster. For this, an objective function that includes a multidimensional distance function as a measure of dissimilarity and entropy as the regularization term is minimized. Experiments on simulated, real world and image data corroborate the usefulness of the proposed algorithm.

Key-words: Fuzzy clustering. Adaptive distances. Maximum-entropy regularization. Objective function.

LIST OF ILLUSTRATIONS

Figure 1 – Obtained hard partition from the dataset. 23

Figure 2 – Obtained fuzzy partition from the dataset. 24

Figure 3 – Prototype vector provided from the clustering algorithms. 24

Figure 4 – Plot of (a) synthetic 3 classes dataset, (b) projection into Feature 1 and Feature 2 subspace, (c) projection into Feature 2 and Feature 3 subspace and (d) projection into Feature 1 and Feature 3 subspace. 46

Figure 5 – The projection of the synthetic data in a single dimension (a) Feature 1, (b) Feature 2, (c) Feature 3, two dimensions (d) Feature 1 and Feature 2, (e) Feature 2 and Feature 3, (f) Feature 1 and Feature 3 and three dimensions (g) Feature 1, Feature 2 and Feature 3. In both (d) and (e) we can see that two clusters are properly separated, but the remaining two are mixed together. In (f) the four clusters are more visible, but still overlap each other are are impossible to completely separate 62

Figure 6 – Plots of synthetic datasets with different shape sets (a) Aggregation, (b) Compound, (c) Flame and (d) Spiral. 63

Figure 7 – Fuzzy and hard partition matrices obtained by (a) and (e) Fuzzy C-Means (FCM), (b) and (f) Fuzzy C-Means with Entropy Regularization (FCM-ER), (c) and (g) Fuzzy Co-Clustering of Images (FCCI) and (d) and (h) Adaptive Fuzzy C-Mean clustering algorithm with entropy regularization (AFCM-ER) algorithms. The first row shows the fuzzy partition matrices and the second row the hard. Lighter colors correspond to higher object membership values in the cluster, and darker colors are associated with low membership values. In every case, abscissa axis corresponds to the objects and ordinate axis with retrieved clusters. 65

Figure 8 – Box plots for the obtained values of (a) Simplified Fuzzy Silhouette (SFS) in Gaussian dataset, (b) Hullermeier and Rifqi (FRHR) in Flame dataset and (c) FRHR in Aggregation dataset. 69

Figure 9 – Distribution of different attributes of Iris plant dataset. 74

Figure 10 – Fuzzy and hard partition matrices in real datasets obtained by (a) and (e) FCM, (b) and (f) FCM-ER, (c) and (g) FCCI and (d) and (h) AFCM-ER algorithms. The first row corresponds to the fuzzy partition matrices and the second row the hard. Lighter colors correspond to higher object membership values in the cluster, and darker colors are associated with low membership values. In every case, abscissa axis corresponds to the objects and ordinate axis with retrieved clusters. 75

Figure 11 – Clustering results on Breast tissue dataset using different metrics. 78

Figure 12 – Clustering results on UKM dataset using different metrics. 78
Figure 13 – Box plot for Scale dataset. 80
Figure 14 – Segmentation results (a) for two clusters, (b) for three clusters, (c) for four clusters and (d) the corresponding Clustering Validity Measure (Xie-Beni index). 84
Figure 15 – Color segmentation results of 10 test images from Berkeley segmentation dataset by the proposed method. 86
Figure 16 – Color segmentation results of different images (from top to bottom) with (a) Original images from Berkeley segmentation database (b) FCM method (c) FCM-ER (d) FCCI (e) Proposed method segmentations. 88
Figure 17 – Color segmentation results of different images (from top to bottom) with (a) Original images from Berkeley segmentation dataset (b) FCM method (c) FCM-ER (d) FCCI (e) Proposed method segmentations. 89

LIST OF TABLES

Table 1 – Dataset representation 22

Table 2 – Mean vector (μ_1, μ_2, μ_3) and standard deviation vector $(\sigma_1, \sigma_2, \sigma_3)$ for every cluster in synthetic Gaussian dataset. 61

Table 3 – Summary of the synthetic datasets. 63

Table 4 – Parameters values for each algorithm in synthetic datasets. 64

Table 5 – The best clustering results obtained on Synthetic dataset for fuzzy partitions. 64

Table 6 – Average performance ranking for fuzzy partitions. 65

Table 7 – The best clustering results obtained on Synthetic dataset for hard partitions. . 66

Table 8 – Average performance ranking for hard partitions. 67

Table 9 – Attribute weight assignment on Gaussian dataset. 67

Table 10 – Clustering performance on the synthetic datasets for fuzzy partitions. 68

Table 11 – Clustering performance on the synthetic datasets for hard partitions. 70

Table 12 – Summary of the real datasets available at the *UCI* machine learning repository. 71

Table 13 – Parameters values in real datasets. 73

Table 14 – Attribute weight assignment on Iris plant dataset. 74

Table 15 – Cluster validity measures on real datasets for fuzzy partitions. 76

Table 16 – Average performance ranking for fuzzy partitions. 76

Table 17 – Clustering performance on real datasets for fuzzy partitions. The first value corresponds to the mean and the second value is the standard deviation of 100 results. 77

Table 18 – Cluster validity measures on real datasets for hard partitions. 79

Table 19 – Average performance ranking for hard partition on real datasets. 79

Table 20 – Clustering performance on real datasets for hard partitions. The first value corresponds to the mean and the second value is the standard deviation of 100 results. 81

Table 21 – Probabilistic Rand index (PR) and Liu's F-measure values for the obtained images segmentation from Berkeley dataset by the proposed algorithm (AFCM-ER) and the other approaches. 87

LIST OF ABBREVIATIONS AND ACRONYMS

AFCM-ER Adaptive Fuzzy C-Mean clustering algorithm with entropy regularization.

ARI Adjusted Rand Index.

CS Crisp Silhouette.

CSSC Conventional Soft Subspace Clustering.

EM Expectation-Maximization.

ESSC Enhanced Soft Subspace Clustering.

ExSSC Extended Soft Subspace Clustering.

FCCI Fuzzy Co-Clustering of Images.

FCM Fuzzy C-Means.

FCM-ER Fuzzy C-Means with Entropy Regularization.

FM F-Measure.

FRHR Hullermeier and Rifqi.

FS Fuzzy Silhouette.

GMMs Gaussian Mixture Models.

HSC Hard Subspace Clustering.

ISSC Independent Soft Subspace Clustering.

K-L Kullback–Leibler.

KLFCM Fuzzy C-Means clustering with regularization by Kullback–Leibler information.

MPC Modified Partition Coefficient.

OAR Overall Average performance Ranking.

OERC Overall Error Rate of Classification.

PC Partition Coefficient.

PE Partition Entropy coefficient.

PR Probabilistic Rand index.

SC Subspace Clustering.

SCS Simplified Crisp Silhouette.

SFS Simplified Fuzzy Silhouette.

SSC Soft Subspace Clustering.

LIST OF SYMBOLS

N	Number of objects
P	Number of variables
C	Number of clusters
E	Observations set
e_i	i-th element of E
$\mathbf{x}_i$	Vector that describes e_i
x_{ij}	Describes i-th object in the j-th variable
D	Data set
D_i	i-th cluster
$\mathbf{G}$	Cluster prototypes set
$\mathbf{g}_k$	Prototype of the k-th cluster
$\mathbf{U}$	Fuzzy partition
u_{ik}	Membership degree of j-th object in the cluster k
$\mathbf{V}$	Matrix of relevance weights
v_{kj}	Relevance weight of the j-th variable for the k-th fuzzy cluster
$\emptyset$	Empty set
∂	Partial derivative
α	Fuzzy C-Means parameter
λ	Lagrange multiplier for the membership degree matrix constraint.
β	Lagrange multiplier for the variable relevance weights constraint.
J	Objective function
O	Time complexity
T	Maximum number of iterations

ε	Minimum improvement in objective function between two consecutive iterations
T_u	Objective function parameter
T_v	Objective function parameter to measure entropy regularization term importance.
$\mathcal{L}$	Lagrangian expression
$\mathbf{Q}$	Hard partition
$\mathbf{Q}_k$	*k*-th partition hard

SUMMARY

1 INTRODUCTION . . . 18

1.1 Motivation . . . 18

1.2 Objective . . . 19

1.3 Publications . . . 20

1.4 Organization of Dissertation . . . 20

2 OVERVIEW OF THE PROBLEM . . . 21

2.1 Partitional clustering overview . . . 22

2.1.1 FCM algorithm . . . 26

2.2 Entropy and Clustering . . . 29

2.2.1 Maximum entropy clustering algorithms . . . 29

2.2.1.1 FCM-ER algorithm . . . 30

2.3 Subspace Clustering Theory . . . 33

2.3.1 Dimensionality Reduction . . . 34

2.3.2 Subspace Clustering . . . 35

2.3.2.1 FCCI algorithm . . . 38

2.4 Chapter summary . . . 42

3 FUZZY C-MEANS CLUSTERING ALGORITHM WITH ADAPTIVE EUCLIDEAN DISTANCES AND ENTROPY REGULARIZATION . . . 44

3.1 The AFCM-ER algorithm objective function . . . 46

3.1.1 The optimization steps of the AFCM-ER algorithm . . . 47

3.1.2 The algorithm . . . 51

3.2 Chapter summary . . . 52

4 EXPERIMENTAL EVALUATION . . . 54

4.1 Experimental setting . . . 54

4.1.1 Partition Coefficient . . . 55

4.1.2 Partition Entropy coeficient . . . 55

4.1.3 Modified Partition Coefficient . . . 55

4.1.4 Crisp silhouette . . . 56

4.1.5 Simplified Fuzzy Silhouette . . . 56

4.1.6 Adjusted Rand Index . . . 57

4.1.7 Campello . . . 58

4.1.8 Hullermeier and Rifqi . . . 58

4.1.9 F-Measure . . . 59

4.1.10 Overall Error Rate of Classification 59
4.1.11 Parameter setting 60
4.2 Experiments on synthetic dataset 61
4.2.1 Results and analysis 63
4.3 Experiments on real datasets 71
4.3.1 Results and analysis 73
4.4 Color segmentation 82
4.4.1 Segmentation evaluation indices 82
4.4.2 Experimental setting 83
4.4.3 Results and analysis 85
4.5 Chapter summary 90

5 CONCLUSIONS 91
5.1 Contributions 91
5.2 Summary of Results 92
5.3 Limitations 93
5.4 Future Work 93

REFERENCE 95

APPENDIX 105

APPENDIX A – PAPER FUZZ-IEEE 2017 106

1 INTRODUCTION

Clustering method aims at partitioning objects or data samples into groups such that the data samples in the same cluster are relatively similar, while the objects in different clusters are relatively dissimilar (BEZDEK, 2013; HAVENS et al., 2012; HUANG; CHUANG; CHEN, 2012; WU et al., 2012; RANA; JASOLA; KUMAR, 2013; LIPOVETSKY, 2013). Clustering as a kind of unsupervised technique has a long history in machine learning, pattern recognition, data mining, and so on, and many clustering algorithms have been exploited for various application scenarios. However, clustering research is still far from a termination. The main reason is that the ever-growing complexity of data constitutes a central challenge for the clustering community. This complexity exhibits in various aspects, including the size of the dataset, the type of features, the temporarily, or scalability and more generally the multiplicity of data (JIANG et al., 2015).

The most popular clustering algorithms are hierarchical and partitioning methods. 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, whereas partitioning methods aim to obtain a single partition of the input data in a fixed number of clusters, typically by optimizing (usually locally) an objective function. The result is a creation of separation hypersurfaces among groups. Partitioning clustering methods can be performed mainly in two different ways: hard and fuzzy. In hard clustering, the clusters are disjoint and non-overlapped. In this case, any pattern may belong to one and only one group. On the other hand, in fuzzy clustering, an object may belong to all clusters with a specific fuzzy membership degree. The degree of membership is essential for discovering intricate relations which may arise between a given data object and all clusters (KAUFMAN; ROUSSEEUW, 2009). The fuzzy clustering method is peculiarly effective when the boundaries between clusters of data are ambiguous. A good review of main fuzzy clustering algorithms can be found in (HÖPPNER, 1999). Moreover, surveys of various clustering methods can be found, for example, in (JAIN, 2010; JAIN; MURTY; FLYNN, 1999; XU; WUNSCH, 2005).

1.1 Motivation

Over the past few decades, various partitioning clustering algorithms have been proposed, where K-means (DUDA; HART, 1973) and Fuzzy C-Means (FCM) (DUNN, 1973) are two well-known clustering algorithms. One drawback of these clustering algorithms is that they treat all features equally in deciding objects cluster membership. This disadvantage is not desirable in some applications, such as high-dimensions sparse data clustering, where the cluster structure in the dataset is often limited to a subset of features rather than the entire feature set. A better solution is to introduce the proper attribute weight into the clustering process (TSAI; CHIU, 2008a). Subspace clustering methods focus on seeking clusters in particular projections of dimensions (subspaces). A subspace cluster can be found in arbitrary subspaces rather than only in the entire space. In other words, only the significant subspaces are found with clusters by subspace clustering algorithms.

Several attribute-weighted fuzzy clustering methods have been proposed in the past few decades. Keller and Klawonn (KELLER; KLAWONN, 2000) introduced a basic attribute-weighted FCM algorithm by assigning one influence parameter to each single data dimension for each cluster, while Frigui and Nasraoui (FRIGUI; NASRAOUI, 2004b) put forward an approach searching for the optimal prototype parameters and the optimal feature weights simultaneously. The algorithm proposed in (DENG et al., 2010) presents a Enhanced Soft Subspace Clustering (ESSC) algorithm by employing both within-cluster and between-cluster information.

Sadaaki and Masao (SADAAKI; MASAO, 1997) proposed a version of Fuzzy C-Means with Entropy Regularization (FCM-ER). Despite the usefulness of FCM-ER, in comparison with the conventional FCM, it still assumes that the variables have the same importance for the clustering task. Later, Ref. (HANMANDLU et al., 2013) proposed a FCM with entropy regularization and automatic selection of variables.

Choosing parameters is important for many clustering algorithms because adjusting the parameters can change the clustering results. In many subspace clustering algorithms, parameters are usually required for both clusters and subspaces detection. However, a major problem is that most parameters can be difficult to determine, especially if a data set has unknown information or complicated types of data. The clustering algorithms can be even inapplicable for some situations when their parameters are hard to choose.

1.2 Objective

The main aim of this research is to develop a partitional Soft Subspace Clustering (SSC) method for clustering data that resolves the previous works limitations. In order to achieve this purpose, we try to reduce parameters and to make parameters easily determinable.

Specific objectives of this work are:

- Investigate state of the art in fuzzy clustering algorithms, especially in methods based on automatic variable selection and entropy regularization.
- Proposal and implementation a new fuzzy partitional clustering algorithm.
- Apply the algorithm to pattern recognition problems.
- Evaluate the results obtained with the proposed method.
- Compare results with literature algorithms.

1.3 Publications

This work resulted, as a bibliographical production, in the article ***"Fuzzy clustering Algorithm with Automatic Variable Selection and Entropy Regularization"*** accepted and published in the ***2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*** where the fuzzy clustering with automatic selection of variables and entropy regularization was evaluated. In Appendix A is show the first page of the article (RODRÍGUEZ; CARVALHO, 2017).

1.4 Organization of Dissertation

In addition to this introductory chapter, this dissertation is organized into more five chapters, as follows.

In chapter 2 it is seen an overview of the problem where hierarchical and partitional clustering concepts are described. Fuzzy partitional clustering methods are reviewed, and the FCM algorithm is briefly commented. The use of regularization term in the objective function is shown in section 2.2 using the maximum entropy approach. The FCM-ER is described and analyzed in subsection 2.2.1.1. In section 2.3 are discussed feature transformation and feature selection techniques. The so-called Fuzzy Co-Clustering of Images (FCCI) (HANMANDLU et al., 2013) is also reviewed because of the importance it has for the understanding of the proposed work.

In chapter 3 the contribution of this work is shown: the proposal of a new FCM-type algorithm for SSC based on adaptive Euclidean distances and entropy regularization term in the objective function.

In chapter 4 are presented a set of experiments performed with both simulated and real datasets, to demonstrate the effectiveness of the proposed method. The obtained results validated the superiority of the proposed clustering method respect to conventional approaches. The algorithm is also tested for color segmentation with satisfactory results.

Finally, in chapter 5, conclusions about method performance, limitations, contributions and future work are provided.

2 OVERVIEW OF THE PROBLEM

We are living in a world full of data. Every day, people encounter a significant amount of information and store or represent it as data, for further analysis and management. One of the vital means of dealing with these data is to classify or group them into a set of categories or clusters. Actually, as one of the most fundamental activities of human beings, classification plays an essential and indispensable role in the long history of human development. To learn a new object or understand a new phenomenon, people always try to seek the features that can describe it, and further compare it with other known objects or phenomena, based on the similarity or dissimilarity, generalized as proximity, according to some certain standards or rules. Classification systems are either supervised or unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively (BISHOP, 1995). In supervised classification, the mapping from a set of input data vectors $\mathbf{x} \in \mathbb{R}^P$, where P is the input space dimensionality, to a finite set of discrete class labels $y \in \{1, ..., C\}$, where C is the total number of classes, is modeled regarding some mathematical function $y = f(\mathbf{x}, \mathbf{w})$, where $\mathbf{w}$ is a vector of adjustable parameters. The values of these parameters are determined (optimized) by an inductive learning algorithm (also termed inducer). This algorithm aims to minimize an empirical risk functional (related to an inductive principle) on a finite dataset of input-output examples, $\{(\mathbf{x}_i, y_i), i = 1..N\}$, where N is the finite cardinality of the available dataset.

In unsupervised classification, also called clustering or exploratory data analysis, no labeled data are available (EVERITT et al., 2001). The goal of clustering is to separate a finite unlabeled dataset into a limited and discrete set of "natural" hidden data structures, rather than provide an accurate characterization of unobserved samples generated from the same probability distribution. Clustering algorithms partition data into a certain number of clusters (groups, subsets, or categories). Most researchers describe a cluster by considering the internal homogeneity and the external separation (HANSEN; JAUMARD, 1997), i.e., objects in the same cluster should be similar to each other, while patterns in different groups should not. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way. Here is given some simple mathematical definitions of several types of clustering, based on the descriptions of (HANSEN; JAUMARD, 1997).

"Data are defined as series of observation, measurements, or facts in the form of numbers, words, sounds and/or images"(ROBERTS, 2000). Data can be treated as the lowest level of abstraction from which information and knowledge are derived. A dataset is a data-collection, which is usually structured in a tabular form that consists of rows and columns, a tree form with hierarchical structure or a graph form with interconnected nodes. Different structures are required by various applications.

Let $E = \{e_1, ...e_N\}$ be a set of N input patterns/objects. Each object e_i, with $1 \leq i \leq N$, is described by a feature vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{iP})$, with $x_{ij} \in \mathbb{R}$, with $1 \leq j \leq P$. Then, a dataset is a set of all feature vectors, $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and can be represented in tabular form as in Table 1.

e_1	x_{11}	x_{12}	...	x_{1P}
e_i	x_{i1}	x_{i2}	...	x_{iP}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
e_N	x_{N1}	x_{N2}	...	x_{NP}

Table 1 – Dataset representation

A classification for clustering methods is given according to how groups are created. The most popular are hierarchical and partitioning methods. A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change (FISHER, 1987). These type of algorithms are divided into two categories: divisible and agglomerative algorithms (FRITZ; GARCÍA-ESCUDERO; MAYO-ISCAR, 2013). Also, hierarchical clustering suffers from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is used to achieve smaller computation costs by not having to worry about a combinatorial number of different choices (AGRAWAL et al., 1998). Some interesting studies in this direction are Chameleon (GORUNESCU, 2011) and Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) (MINING, 2006).

Hierarchical algorithms are based on distance, density, and continuity. These algorithms have advantages of decreasing calculations and costs and as a disadvantage that they are nonscalable. For large datasets, hierarchical methods become impractical unless other techniques are incorporated, because usually hierarchical methods are $O(N^2)$ for memory space and $O(N^3)$ for CPU time (ZAÏT; MESSATFA, 1997; HARTIGAN; HARTIGAN, 1975; MURTAGH, 1983), where N is the number of data points in the dataset. One of significant issue of hierarchical methods is the inability to reform a wrong decision, making this research focus on partitional algorithms that do not suffers this kind of problems (section 2.1).

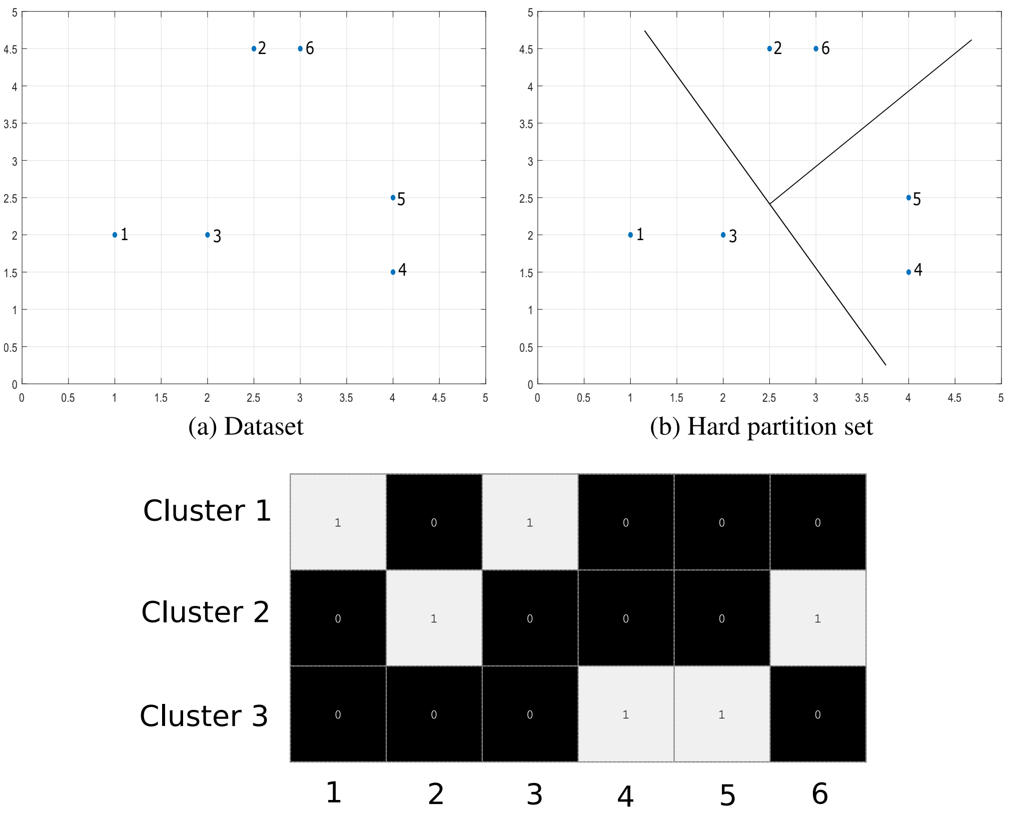
2.1 Partitional clustering overview

Given the number of partitions to construct, a partitional method creates an initial partition. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another (AGRAWAL et al., 1998). The general objective is to obtain the partition that, for a fixed number of clusters, minimizes the total square error. Every partitional clustering algorithm obtains a single partition of the data instead of the clustering structure. Thus, partitional methods have the advantage in applications involving large datasets for which the construction of a dendrogram is computationally very complex (FRITZ; GARCÍA-ESCUDERO; MAYO-ISCAR, 2013).

Clustering algorithms can be classified into two categories, hard and soft (fuzzy) clustering (DUNN, 1973). Hard partitional clustering attempts to seek a C-partition of D, $D = D_1 \cup D_2 \cup ... \cup D_C, C \leq N$ such that:

1. $D_i \neq \emptyset, i = 1, ..., C$;
2. $\cup_{i=1}^{C} D_i = D$;
3. $D_i \cap D_j = \emptyset \; i, j = 1, ..., C$ and $i \neq j$.

For hard partitional clustering, each pattern only belongs to one cluster as established in Axioma 3. (Figure 1). However, an object may also be allowed to belong to all groups with a degree of membership, $u_{ik} \in [0, 1]$, which represents the membership coefficient of the i-th object in the k-th cluster and satisfies that: $\sum_{k=1}^{C} u_{ik} = 1, \forall i$ and $\sum_{i=1}^{N} u_{ik} < N, \forall k$, as introduced in fuzzy set theory (BEZDEK, 1981).



(a) Dataset

(b) Hard partition set

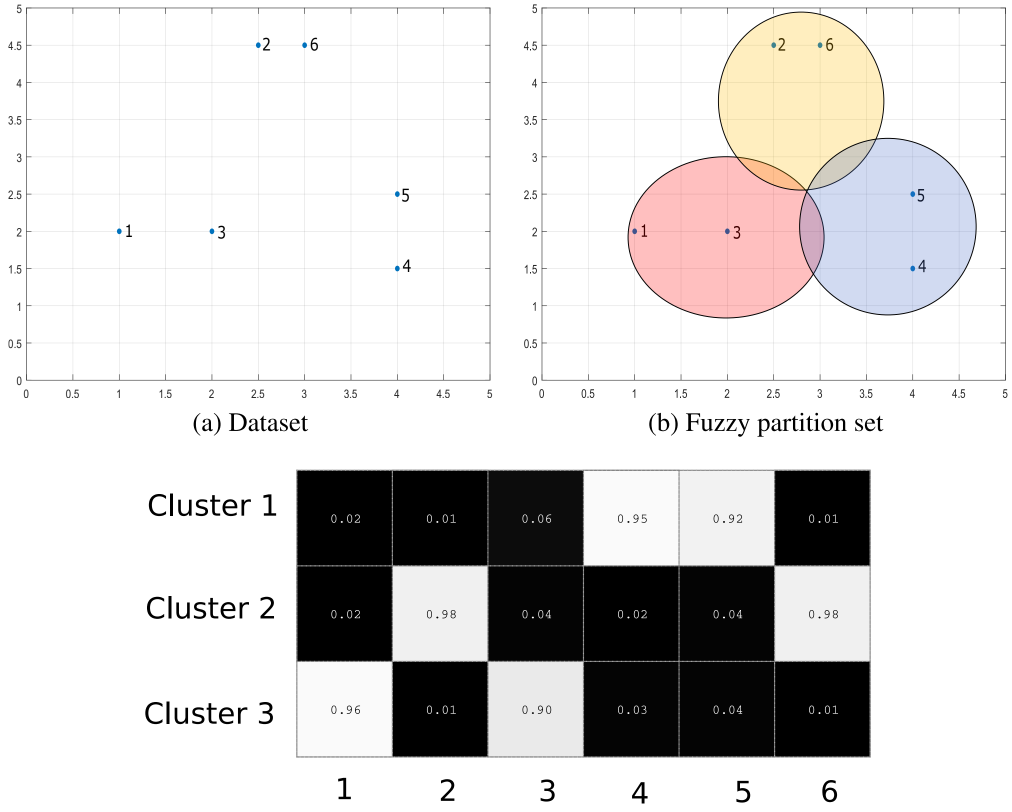
	1	2	3	4	5	6
Cluster 1	1	0	1	0	0	0
Cluster 2	0	1	0	0	0	1
Cluster 3	0	0	0	1	1	0

(c) Hard partition matrix

Figure 1 – Obtained hard partition from the dataset.

Fuzzy clustering is a widely applied method for obtaining fuzzy models from data (Figure 2). It has been used successfully in various fields including geographical surveying, finance or marketing. Fuzzy cluster analysis relaxes that each object must be assigned to exactly one cluster by allowing memberships, thus offering the opportunity to deal with data that belong to more than one group at the same time (ESTER et al., 1996).

Generally, fuzzy clustering algorithms provide:



(a) Dataset

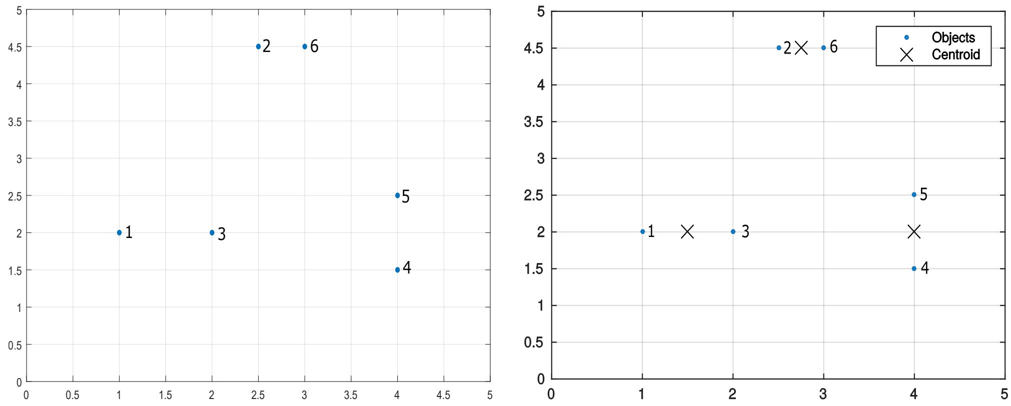
(b) Fuzzy partition set

(c) Fuzzy partition matrix

Figure 2 – Obtained fuzzy partition from the dataset.

- A fuzzy partition represented by the matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N) {=} (u_{ik})_{\substack{1 \le i \le N \\ 1 \le k \le C}}$, where u_{ik} is the membership degree of object e_i into the fuzzy cluster k and $\mathbf{u}_i = (u_{i1}, \ldots, u_{iC})$ (where C is the number of clusters and N the number of objects.);

- A vector $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_C)$ where the component $\mathbf{g}_k = (g_{k1}, \ldots, g_{kP})$ is the representative or prototype (Figure 3) of the fuzzy cluster k, where $g_{kj} \in \mathbb{R}$.



(a) Dataset

(b) Prototype

Figure 3 – Prototype vector provided from the clustering algorithms.

The most common algorithms used for hard clustering are K-Means (DUDA; HART, 1973), K-Medoids (KAUFMAN; ROUSSEEUW, 1987), Partitioning Around Medoids (PAM) (KAUFMAN; ROUSSEEUW, 1987), Clustering LARge Applications (CLARA) (KAUFMAN; ROUSSEEUW, 2009), Clustering Large Applications based on RANdomized Search (CLARANS) (LU et al., 2013) and for soft clustering Fuzzy C-Means (FCM) (MACQUEEN et al., 1967).

K-Means: It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the object and the cluster centers until a convergence criterion is met. The method is relatively scalable and efficient for processing large datasets. The time and space complexity are relatively small, and it is an order-independent algorithm (FRITZ; GARCÍA-ESCUDERO; MAYO-ISCAR, 2013), but the method often terminates at a local optimum and is not suitable for discovering clusters with nonconvex shapes or groups of very different size (AGRAWAL et al., 1998). Moreover, some divergences exist in scientific community about the best approach for initial partition selection, partition updating, number of clusters adjustment, and stopping criterion (FRITZ; GARCÍA-ESCUDERO; MAYO-ISCAR, 2013). A major problem with this algorithm is that it is sensitive to noise and outliers (GENNARI; LANGLEY; FISHER, 1989).

K-Medoid/PAM: PAM was one of the first K-Medoids algorithms introduced. The algorithm uses the most centrally located object in a cluster, the medoid, instead of the mean. Then, PAM starts from an initial set of medoids, and it iteratively replaces one of the medoids by one of the nonmedoids if improves the total cost of the resulting clustering. This algorithm works efficiently for small datasets but does not scale well for large datasets.

CLARA: Instead of taking the whole set of data into consideration, a small portion of the actual data is chosen as a representative of the data. Medoids are then selected from this sample using PAM. CLARA draws multiple examples of the dataset, applies PAM on each sample, and returns its best clustering as the output. As expected, CLARA can deal with larger datasets than PAM.

CLARANS: It draws a sample with some randomness in each step of the search. Conceptually, the clustering process can be viewed as a search through a graph. At each step, PAM examines all of the neighbors of the current node in its search for a minimum cost solution. The current node is then replaced by the neighbor with the largest descent in costs. The algorithm also enables the detection of outliers.

FCM: The Fuzzy C-Means algorithm is most widely used (DUNN, 1973), and an extension of classical and crisp K-Means clustering method in the fuzzy set domain so that it is extensively studied and applied in pattern recognition, image segmentation and image clustering, data mining, wireless sensor network among others (MITRA; ACHARYA, 2005). The objects in FCM can belong to more than one cluster, and a membership grade is associated with each of the objects indicating the degree to which objects belong to different groups. This research is based of FCM algorithm and more details about it are given in subsection 2.1.1.

Partitional algorithms are based on distance. These algorithms have as advantage low computational complexity and as a disadvantage the need to set the number of clusters and the stop criterion.

One of the standard measures in the partitional clustering algorithms is based on a squared error metric or the sum of squares metrics, which measures the squared (or sum of the squared) distance (i.e., Euclidean spaces) between each instance in the associated cluster and the cluster centroid, $\mathbf{g}_k$. As mentioned before, this type of grouping is called here the sum-of-squares clustering, which aims to minimize the total within-cluster sum of squares (WEBB, 2003).

The sum-of-squares clustering is a categorization method type which divides a large group of instances into a pre-determined number, i.e., C smaller clusters so that the instances within each group (within-cluster information) are relatively similar while those between groups (between-cluster information) are relatively dissimilar (KONONENKO; KUKAR, 2007). Let us consider a set of clusters $D = \{D_1, ..., D_C\}$ with the respective centroids $\mathbf{G} = \{\mathbf{g}_1, ..., \mathbf{g}_c\}$, which are mapped from a set of instances $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_i, ..., \mathbf{x}_N\}$. The sum of the sums of squares (square error) over the C clusters is defined as

$$d_{ik}^2 = ||\mathbf{x}_i - \mathbf{g}_k||^2 = \sum_{j=1}^{P}(x_{ij} - g_{kj})^2 \tag{2.1}$$

Where $\mathbf{x}_i$ is the i-th object, $\mathbf{g}_k$ is the prototype of the k-th fuzzy cluster and P represents the number of variables. Many different types of sum-of-squares clustering algorithms depend on the choice of clustering criterion to be optimized and the optimization procedure to be adopted (WEBB, 2003), but mostly, they follow the basic algorithm structure by optimizing the metric measures (i.e., in Equation 2.1) to obtain the desired clusters.

2.1.1 FCM algorithm

The fuzzy set theory was proposed by (ZADEH, 1965), and since then it has had significant influence in various areas and applications, which follow the concept of fuzziness and uncertainty of belonging through a membership function. In the datasets, it is uncertain for the instances to form exactly one cluster in their membership relations when cluster analysis is carried out. Therefore, the concept of fuzzy membership in fuzzy clustering has been widely studied by researchers. The basic idea of fuzzy clustering method is to allow each instance to belong to all clusters with different degrees of membership (membership function) by extending the hard membership interval from $\{0, 1\}$ to $[0, 1]$. Thus, the fuzzy objective function that the FCM optimizes (minimizes) is defined by (WANG, 1983) as follows:

$$J_{FCM}(\mathbf{U}, \mathbf{G}) = \sum_{i=1}^{N}\sum_{k=1}^{C} u_{ik}^{\alpha} d_{ik}^2 \tag{2.2}$$

where $d_{ik}^2 = \sum_{j=1}^{P}(x_{ij} - g_{kj})^2$ (Equation 2.1) is the squared distance between the feature vectors $\mathbf{x}_i$ and the prototype $\mathbf{g}_k$ and u_{ik} is the membership degree of x_i object in the cluster k. The prototype vector and membership degree matrix are estimated by Equation 2.3 and Equation 2.4 respectively as shown below:

$$\mathbf{g}_k = \frac{\sum_{i=1}^{N} u_{ik}^{\alpha} \mathbf{x}_i}{\sum_{i=1}^{N} u_{ik}^{\alpha}} \tag{2.3}$$

where $i = 1, ..., N$, $k = 1, ..., C$ and $\alpha \in (1, \infty)$

$$u_{ik} = \frac{1}{\sum_{h=1}^{C} \left(\frac{d_{ik}}{d_{ih}}\right)^{\frac{2}{\alpha-1}}} \tag{2.4}$$

where $0 \leq u_{ik} \leq 1$, $\sum_{k=1}^{C} u_{ik} = 1$ and $0 < \sum_{i=1}^{N} u_{ik} < N$. By minimizing Equation 2.2 using the Lagrangian optimization:

$$\mathcal{L}_{FCM} = J_{FCM} - \sum_{i=1}^{N} \lambda_i \left[\sum_{k=1}^{C} u_{ik} - 1\right]$$

Taking the partial derivative of $\mathcal{L}_{FCM}$ concerning u_{ik} and set the gradient to zero:

$$\frac{\partial \mathcal{L}_{FCM}}{\partial u_{ik}} = \alpha u_{ik}^{\alpha-1} d_{ik}^2 - \lambda_i = 0 \tag{2.5}$$

then

$$u_{ik} = \left(\frac{\lambda_i}{\alpha} \frac{1}{d_{ik}^2}\right)^{\frac{1}{\alpha-1}} \tag{2.6}$$

It is known that $\sum_{k=1}^{C} u_{ik} = 1$, then:

$$\sum_{h=1}^{C} u_{ih} = \sum_{h=1}^{C} \left(\frac{\lambda_i}{\alpha} \frac{1}{d_{ih}^2}\right)^{\frac{1}{\alpha-1}} = 1 \tag{2.7}$$

Solving Equation 2.7 for $\left(\frac{\lambda_i}{\alpha}\right)^{\frac{1}{\alpha-1}}$ and substituting in Equation 2.6

$$u_{ik} = \frac{1}{\sum_{h=1}^{C} \left(\frac{d_{ik}}{d_{ih}}\right)^{\frac{2}{\alpha-1}}}$$

The updating equation for estimating cluster centroid (center) $\mathbf{g}_k$ can be derived as:

$$\frac{\partial J_{FCM}}{\partial \mathbf{g}_k} = \sum_{i=1}^{N} u_{ik}^{\alpha} \mathbf{x}_i - \mathbf{g}_k \sum_{i=1}^{N} u_{ik}^{\alpha} = 0 \tag{2.8}$$

Solving Equation 2.8 yields the formula for $\mathbf{g}_k$ as:

$$\mathbf{g}_k = \frac{\sum_{i=1}^{N} u_{ik}^{\alpha} \mathbf{x}_i}{\sum_{i=1}^{N} u_{ik}^{\alpha}}$$

In FCM method the loss (objective) function is defined as shown in Equation 2.2. Where $\mathbf{U}$ denotes the grade of membership of the i-th pattern in the k-th fuzzy cluster, $\mathbf{g}_k$ is interpreted as cluster center or prototype of the k-th cluster and α is a weighting exponent that controls the extent of membership sharing between fuzzy clusters.

This FCM algorithm is a generalization of the basic K-Means algorithm. When the value of $\alpha = 1$, the FCM tends to be the basic K-Means. In general, the FCM performs comparatively better than K-Means because the uncertainty less influences it and overlapping in the dataset. The Algorithm 1 summarizes the FCM method steps.

Algorithm 1 FCM Algorithm

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

Output:
       The vector of prototypes $\mathbf{G}$;
       The matrix of membership degrees $\mathbf{U}$;
Initialization
       Set $t = 0$;
       Randomly select $C$ distinct prototypes $\mathbf{g}_k^{(t)} \in D$, $k = \{1, ..., C\}$ to obtain the vector
       of prototypes $\mathbf{G} = (\mathbf{g}_1, ..., \mathbf{g}_C)$;
       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 $\sum_{k=1}^{C} u_{ik}^{(t)} = 1$;
       Compute $J_{FCM}$ according to Equation 2.2.
repeat
       Set $t = t + 1$
       Set $J_{OLD} = J_{FCM}$
       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 2.3.
       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 2.4.
       Compute $J_{FCM}$ according to Equation 2.2.
       Set $J_{NEW} = J_{FCM}$.
until $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$
```

A strict implementation of the FCM algorithm has an expected runtime complexity of $O(NPTC^2)$ (HORE; HALL; GOLDGOF, 2007), where N is the number of data examples, C is the number of clusters, P is the dimension of the data, and T is the number of iterations. It is possible to reduce the runtime to $O(NPTC)$ with an optimization proposed by Kolen and Hutcheson (KOLEN; HUTCHESON, 2002). The memory complexity of FCM is $O(NP + NC)$ (SREENIVASA-RAO; VIDYAVATHI, 2010), where NP is the size of the dataset and NC the size of the $\mathbf{U}$ matrix.

Though FCM is a more applicable method; there are some objections against it:

1. Due to using Euclidean distance based on L_2-norm space, the presence of outliers in the dataset degrades the quality of the computed clustering centers.
2. The physical meaning of the fuzziness parameter α and a way of choosing its optimal value is not well understood.
3. FCM deals with local minimum solutions only: it does not possess a mechanism with which one can get global minimum solutions.
4. There is no easy way to determine the number of clusters.
5. Sensitivity to the initial guess (speed, local minima)
6. The FCM algorithm gives high membership values for the outlier points and due to this the algorithm has difficulty in handling outlier points.

These problems sometimes lead to ineffective partitioning, incorrect localization and eventually, poor clustering performance (THOMAS; NASHIPUDIMATH, 2012; COX, 2005).

2.2 Entropy and Clustering

Entropy is the measure of information and uncertainty of a random variable (SHANNON, 2001). Formally, if X is a random variable, $S(X)$ the set of values that X can take, and $p(x)$ the probability function of X, the entropy $E(X)$ is defined as:

$$E(X) = -\sum_{X \in S(X)} p(X)\log(p(X)) \tag{2.9}$$

Entropy is sometimes referred to as a measure of the amount of disorder in a system. A room with socks strewn all over the floor has more entropy than a room in which socks are paired up, neatly folded, and placed in one side of the sock and underwear drawer.

According to the principle of maximum entropy, if nothing is known about a distribution except that it belongs to a particular category (usually defined regarding specified properties or measures), then the distribution with the largest entropy should be chosen as the least-informative default. The motivation is twofold: first, maximizing entropy minimizes the amount of prior information built into the distribution; second, many physical systems tend to move towards maximal entropy configurations over time.

2.2.1 Maximum entropy clustering algorithms

As seen in previous sections, the objective of a clustering algorithm is the assignment of a set of N feature vectors $\mathbf{x}_i \in D$ into C clusters, which are represented by the prototypes $\mathbf{g}_k \in \mathbf{G}$. The certainty of the assignment of the feature vector $\mathbf{x}_i$ into various clusters is measured by the membership functions $u_{ik} \in [0,1], k = 1, ..., C$ which satisfy the property $\sum_{i=1}^{C} u_{ik} = 1$. In other words $u_{ik} \in \mathbf{U}$, where the set $\mathbf{U}$ is defined as $\mathbf{U} = \{u_{ik} \in [0,1] : \sum_{k=1}^{C} u_{ik} = 1 \; \forall i = 1, ..., N\}$. For a given set of membership functions, the squared distance between the feature vectors $\mathbf{x}_i$ and the prototype $\mathbf{g}_k$ is measured by:

$$d_{ik}^2 = \sum_{j=1}^{P} (x_{ij} - g_{kj})^2 \tag{2.10}$$

Most of the disadvantages of crisp clustering algorithms can be overcome by clustering algorithms based on the gradual transition from soft to crisp decisions during the clustering process (KARAYIANNIS; PAI, 1995). Such strategies can be implemented by formulating the clustering problem as the minimization of an objective function which guarantees the transition from the maximum uncertainty or minimum selectivity phase in the initial stages of the clustering process to the minimum uncertainty or maximum selectivity phase toward the end of the clustering process. The minimization of the squared distance between the feature vectors and the prototypes measured by Equation 2.10 can be the basis for the clustering process in the minimum uncertainty phase. If the entropy of u_{ik} is maximized, the assignment of feature vectors into clusters is characterized by maximum uncertainty. In the minimum selectivity phase, the clustering process is based on the maximization of the entropy

$$E(u_{ik}) = -\sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik} \log(u_{ik}) \tag{2.11}$$

or, equivalently, the minimization of the negative entropy, given by

$$E(u_{ik}) = \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik} \log(u_{ik}) \tag{2.12}$$

Several maximum entropy clustering algorithms and their variants are available in the literature (ROSE; GUREWITZ; FOX, 1990; JAYNES, 1957). Miyagishi *et al.* (MIYAGISHI et al., 2000) generalized the regularized objective function replacing the entropy term with Kullback–Leibler (K-L) term and proposed an FCM-type counterpart of the Gaussian Mixture Models (GMMs) with full unknown parameters and the clustering technique is called the Fuzzy C-Means clustering with regularization by Kullback–Leibler information (KLFCM).

2.2.1.1 FCM-ER algorithm

One approach of interest is a variant of FCM which takes into account entropy regularization named Fuzzy C-Means with Entropy Regularization (FCM-ER) (LI; SADAAKI, 1995; SADAAKI; MASAO, 1997). It involves the minimization of the following objective function:

$$\begin{aligned} J_{FCM-ER} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(x_{ij}-g_{kj})^2 \\ &+ T_u\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) \end{aligned} \tag{2.13}$$

Subject to: $$\sum_{k=1}^{C}(u_{ik}) = 1 \tag{2.14}$$

The first term in the Equation 2.13 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 which serves as a regularization factor during minimization process. The parameter T_u is the weight factor in the entropy term that specifies the degree of fuzziness. Increasing T_u increases the fuzziness of the clusters.

Deriving the update equations

The constrained optimization problem of FCM-ER can now be defined from Equation 2.13 by applying the Lagrange multipliers λ_i to constraint in Equation 2.14 as shown below.

$$\begin{aligned} \mathcal{L}_{FCM-ER} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\sum_{j=1}^{P}(x_{ij}-g_{kj})^2 \\ &+ T_u\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\ln(u_{ik}) + \sum_{i=1}^{N}\lambda_i\left[\sum_{k=1}^{C}u_{ik}-1\right] \end{aligned} \tag{2.15}$$

Taking the partial derivative of $\mathcal{L}_{FCM-ER}$ in Equation 2.15 with respect to u_{ik} and setting the gradient to zero:

$$\frac{\partial\mathcal{L}_{FCM-ER}}{\partial u_{ik}} = \sum_{j=1}^{P}(x_{ij}-g_{kj})^2 + T_u(\ln(u_{ik})+1) + \lambda_i = 0 \tag{2.16}$$

Solving Equation 2.16

$$u_{ik} = e^{-\frac{\sum_{j=1}^{P}(x_{ij}-g_{kj})^2}{T_u}}e^{-1-\frac{\lambda_i}{T_u}} \tag{2.17}$$

It is known that $\sum_{k=1}^{C}(u_{ik}) = 1$ then substituting u_{ik} for the expression in Equation 2.17:

$$\sum_{k=1}^{C}u_{ik} = \sum_{k=1}^{C}e^{-\frac{\sum_{j=1}^{P}(x_{ij}-g_{kj})^2}{T_u}}e^{-1-\frac{\lambda_i}{T_u}} = 1 \tag{2.18}$$

Solving Equation 2.18

$$e^{-1-\frac{\lambda_i}{T_u}} = \frac{1}{\sum_{k=1}^{C}e^{-\frac{\sum_{j=1}^{P}(x_{ij}-g_{kj})^2}{T_u}}} \tag{2.19}$$

Substituting $e^{-1-\frac{\lambda_i}{T_u}}$ (Equation 2.19) in Equation 2.17, the formula for computing the object membership function u_{ki} reduces to:

$$u_{ik} = \frac{e^{-\sum_{j=1}^{P} \frac{(x_{ij}-g_{kj})^2}{T_u}}}{\sum_{h=1}^{C} e^{-\sum_{j=1}^{P} \frac{(x_{ij}-g_{hj})^2}{T_u}}} \tag{2.20}$$

Taking the partial derivative of $J_{FCM-ER}(\mathbf{U}, \mathbf{G})$ concerning $\mathbf{G}$ and setting the gradient to zero:

$$\frac{\partial J_{FCM-ER}}{\partial g_{kj}} = \sum_{i=1}^{N} u_{ik} x_{ij} - g_{kj} \sum_{i=1}^{N} u_{ik} = 0 \tag{2.21}$$

Solving Equation 2.21 yields the formula for g_{kj} as:

$$g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{2.22}$$

Optimal partitions $\mathbf{U}^*$ of D can be obtained by solving for $(\mathbf{U}^*, \mathbf{G}^*)$ at the local minima of J_{FCM-ER}.

Proposition 1. *The updated values of u_{ik} given by Equation 2.20 never increase the objective function in every iteration.*

Proof. Consider the objective function as a function of u_{ik} alone.

$$J_{FCM-ER}(\mathbf{U}, \mathbf{G}) = \sum_{k=1}^{C} \sum_{i=1}^{N} u_{ik} \sum_{j=1}^{P} (x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C} \sum_{i=1}^{N} u_{ik} \ln(u_{ik}) \tag{2.23}$$

where $(x_{ij} - g_{kj})^2$ may be considered as constant. To prove Proposition 1, it has to be proved that $\mathbf{U}^*$, i.e the updated values of u_{ik} given by Equation 2.20 are the local minima of the objective function $J_{FCM-ER}(\mathbf{U}^*)$ provided that the constraint in Equation 2.14 is satisfied. For this it is needed prove that the Hessian matrix $\partial^2 J_{FCM-ER}(\mathbf{U}^*)$ is positive definite.

$$\partial^2 J_{FCM-ER}(\mathbf{U}^*) = \begin{bmatrix} \frac{\partial^2 J_{FCM-ER}(\mathbf{U})}{\partial u_{11} \partial u_{11}} & \cdots & \frac{\partial^2 J_{FCM-ER}(\mathbf{U})}{\partial u_{11} \partial u_{NC}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J_{FCM-ER}(\mathbf{U})}{\partial u_{NC} \partial u_{11}} & \cdots & \frac{\partial^2 J_{FCM-ER}(\mathbf{U})}{\partial u_{NC} \partial u_{NC}} \end{bmatrix} = \begin{bmatrix} \frac{T_u}{u_{11}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{T_u}{u_{NC}} \end{bmatrix}$$

At $\mathbf{U}^*, u_{ik} \geq 0$ and T_u is always assigned a positive value. Therefore the Hessian matrix $\partial^2 J_{FCM-ER}(\mathbf{U}^*)$ is positive definite. Then, it is proved the first necessary condition

$$\frac{\partial J_{FCM-ER}(u_{ik})}{\partial u_{ik}} = 0 \tag{2.24}$$

and the second sufficient condition that $\delta^2 J_{FCM-ER}(\mathbf{U}^*)$ is positive definite. Therefore u_{ik} update is indeed a local minimum of $J_{FCM-ER}(\mathbf{U})$ and it never increases the objective function value. □

Algorithm 2 FCM-ER Algorithm

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

Output:
       The vector of prototypes $\mathbf{G}$;
       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 $\mathbf{G} = (\mathbf{g}_1, ..., \mathbf{g}_C)$;
       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 $\sum_{k=1}^{C} u_{ik}^{(t)} = 1$;
       Calculate $J_{FCM-ER}$ according to Equation 2.13;
2: repeat
       Set $t = t + 1$;
       Set $J_{OLD} = J_{FCM-ER}$;
       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 2.22;
       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 2.20;
       Calculate $J_{FCM-ER}$ according to Equation 2.13;
       Set $J_{NEW} = J_{FCM-ER}$;
3: until $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$
```

The FCM-ER algorithm steps are summarized in Algorithm 2.

The computational complexity for FCM-ER algorithm is $O(NCPT)$, where N is the number of data examples, C is the number of clusters, P is the dimension of the data and T is the number of iterations. The memory complexity of the algorithm is:

- CN for the fuzzy partition matrix $\mathbf{U}$;

- CP for the representative (prototype) vector $\mathbf{G}$.

2.3 Subspace Clustering Theory

Traditional clustering algorithms consider all features of an input dataset in an attempt to learn as much as possible about each object described. In high dimensional data, however, many of the dimensions are often irrelevant. These irrelevant attributes can confuse clustering algorithms by hiding clusters in noisy data. In very high dimensions it is common for all of the objects in a dataset to be nearly equidistant from each other, completely masking the clusters.

2.3.1 Dimensionality Reduction

Techniques for clustering high dimensional data have included both feature transformation and feature selection techniques. Feature transformations are commonly used on high dimensional datasets. These methods include techniques such as principal component analysis and singular value decomposition. The transformations preserve the original, relative distances between objects. In this way, they summarize the dataset by creating linear combinations of the attributes, and hopefully, uncover latent structure. Feature transformation is often a preprocessing step, allowing the clustering algorithm to use just a few of the newly created features. A few clustering methods have incorporated the use of such transformations to identify important features and iteratively improve their clustering (DING et al., 2002; HINNEBURG; KEIM, 1999). While often very useful, these techniques do not remove any of the original attributes from consideration. Thus, information from irrelevant dimensions is preserved, making these techniques ineffective at revealing clusters when there are large numbers of irrelevant characteristics that mask the groups. Another disadvantage of using combinations of attributes is that they are difficult to interpret, often making the clustering results less useful. Because of this, feature transformations are best suited to datasets where most of the dimensions are relevant to the clustering task, but many are highly correlated or redundant.

Feature selection attempts to discover the attributes of a dataset that are most relevant to the data mining task at hand. It is a commonly used and powerful technique for reducing the dimensionality of a problem to more manageable levels. Feature selection involves searching through various feature subsets and evaluating each of these subsets using some criterion (BLUM; LANGLEY, 1997; LIU; MOTODA, 2012; PENA et al., 2001). The most popular search strategies are greedy sequential searches through the feature space, either forward or backward. The evaluation criteria follow one of two basic models, the wrapper model, and the filter model (KOHAVI; JOHN, 1997). The wrapper model techniques evaluate the dataset using the data mining algorithm that will ultimately be employed. Thus, they *wrap* the selection process around the data mining algorithm. Algorithms based on the filter model examine intrinsic properties of the data to evaluate the feature subset before data mining.

Much of the work in feature selection has been directed at supervised learning. The main difference between feature selection in supervised and unsupervised learning is the evaluation criterion. Supervised wrapper models use classification accuracy as a measure of goodness. The filter-based approaches almost always rely on the class labels, most commonly assessing correlations between features and the class labels. In the unsupervised clustering problem, there are no universally accepted measures of accuracy and no class labels. However, some methods adapt feature selection to clustering.

Entropy measurements are the basis of the filter model approach presented in (DASH et al., 2002; DASH; LIU; YAO, 1997). The authors argue that entropy tends to be low for data that contains tight clusters, and this is a good measure to determine feature subset relevance. The wrapper method proposed in (DY; BRODLEY, 2004) forms a new feature subset and evaluates the resulting set by applying a standard K-Means algorithm. The Expectation-Maximization (EM) clustering algorithm is used in the wrapper framework in (DY; BRODLEY, 2000). Hybrid methods have also been developed that use a filter approach as a heuristic and refine the results with a clustering algorithm. One such method by Devaney and Ram uses category utility to evaluate the feature subsets (DEVANEY; RAM, 1997). Another hybrid approach uses a greedy algorithm to rank features based on entropy values and then uses K-Means to select the best subsets of features (DASH; LIU, 2000).

In addition to using different evaluation criteria, unsupervised feature selection methods have employed various search methods in attempts to scale to large, high dimensional datasets. With such datasets, random searching becomes a viable heuristic method and has been used with many of the criteria mentioned above (ACHLIOPTAS, 2001; BINGHAM; MANNILA, 2001; FERN; BRODLEY, 2003). Sampling is another method used to improve the scalability of algorithms. Mitra *et al.* partition the original feature set into clusters based on a similarity function and select representatives from each group to form the sample (MITRA; MURTHY; PAL, 2002).

While quite successful on many datasets, feature selection algorithms have difficulty when clusters are found in different subspaces. It is this type of data that motivated the evolution to subspace clustering algorithms. Unlike feature selection methods which examine the dataset as a whole, subspace clustering algorithms localize their search and can uncover clusters that exist in multiple, possibly overlapping subspaces.

2.3.2 Subspace Clustering

Despite extensive studies of clustering techniques over the past decades, conventional methods fall short when clustering is performed in high-dimensional spaces (PARSONS; HAQUE; LIU, 2004; SIM et al., 2013; KRIEGEL; KRÖGER; ZIMEK, 2009). A key challenge of most clustering algorithms is that, in many real-world problems, data points in different clusters are often correlated with some subsets of features, i.e., groups may exist in various subspaces or a particular subspace of all characteristics. Therefore, for any given pair of neighboring data points within the same cluster, it is possible that the points are indeed far apart from each other in a few dimensions of high-dimensional space.

Subspace Clustering (SC) is an extension of feature selection which tries to identify clusters in different subspaces of the same dataset. Like feature selection, subspace clustering needs a search method and evaluation criteria. Also, subspace clustering must somehow restrict the scope of the evaluation criteria to consider different subspaces for each separate cluster.

In recent years a plethora of SC techniques have been developed to overcome this challenge. SC seeks to group objects into clusters on subsets of dimensions or attributes of a dataset. It pursues two tasks, identification of the subsets where clusters can be found and discovery of the groups from different subsets of attributes. According to how the subsets are identified, it can be divided subspace clustering methods into two categories. The algorithms in the first class determine the specific subsets of dimensions where clusters are discovered. These methods are named Hard Subspace Clustering (HSC). The methods in the second category identify the subsets of features according to the contributions of the dimensions in finding the corresponding clusters. The contribution of an attribute is measured by a weight that is assigned to the dimension in the clustering process. These methods are named Soft Subspace Clustering SSC because every dimension contributes to the discovery of clusters, but the attributes with larger weights form the subsets of dimensions of the groups. The method in this work falls in the second category.

Research into SC begins with an in-depth study of HSC methods for clustering high-dimensional data. With HSC algorithms, an attempt is made to identify the exact subspaces for different clusters, a process that can be further divided into bottom-up and top-down subspace search methods (PARSONS; HAQUE; LIU, 2004).

The bottom-up search method takes advantage of the downward closure property of density to reduce the search space, using an a priori style approach. Algorithms first create a histogram for each dimension and select those bins with frequencies above a given threshold. The downward closure property of density means that if there are dense units in k dimensions, there are dense units in all $(k-1)$ dimensional projections. Candidate subspaces in two dimensions can then be formed using only those dimensions which contained dense units, dramatically reducing the search space. The algorithm proceeds until there are no more dense units found. Adjacent dense units are then combined to form clusters. This is not always easy, and one cluster may be mistakenly reported as two smaller clusters. The nature of the bottom-up approach leads to overlapping groups, where one instance can be in zero or more clusters. Obtaining meaningful results is dependent on the proper tuning of the grid size and the density threshold parameters. These can be particularly difficult to set, mainly since they are used across all of the dimensions in the dataset. A popular adaptation of this strategy provides data-driven, adaptive grid generation to stabilize the results across a range of density thresholds.

The top-down subspace clustering approach starts by finding an initial approximation of the clusters in the full feature space with equally weighted dimensions. Next, each dimension is assigned a weight for each group. The updated weights are then used in the next iteration to regenerate the clusters. This approach requires multiple iterations of expensive clustering algorithms in the full set of dimensions. Many of the implementations of this strategy use a sampling technique to improve performance. Top-down algorithms create clusters that are partitions of the dataset, meaning each instance is assigned to only one cluster. Many algorithms also allow for an additional group of outliers. Parameter tuning is necessary to get meaningful results. Often the most critical parameters for top-down algorithms is the number of clusters and the size of the subspaces, which are usually very difficult to determine ahead of time. Also, since subspace size is a parameter, top-down algorithms tend to find clusters in the same or similarly sized subspaces. For techniques that use sampling, the size of the sample is another critical parameter and can play a significant role in the quality of the final results.

While the exact subspaces are identified in HSC, weight is assigned to each dimension in the clustering process of SSC to measure the contribution of each dimension to the formation of a particular cluster. In the clustering procedure, each dimension (i.e., feature) contributes differently to every cluster. The value of weights can identify the subspaces of different clusters after clustering. SSC can be considered as an extension of the standard feature weighting clustering (BOUGUILA, 2009; CHEUNG; ZENG, 2007; DESARBO et al., 1984; HUANG et al., 2005; DIDAY; GOVAERT, 1977) which employs a common weight vector for the whole dataset in the clustering procedure. However, it is also distinct in that different weight vectors are assigned to different clusters. From this perspective, soft subspace clustering may thus be referred to as multiple features weighting clustering. SSC has recently emerged as a hot research topic, and many algorithms have been reported (CHAN et al., 2004; FRIGUI; NASRAOUI, 2004b; FRIGUI; NASRAOUI, 2004a; DOMENICONI et al., 2004; FRIEDMAN; MEULMAN, 2004; JING et al., 2005; GAN; WU; YANG, 2006; GAN; WU, 2008; JING; NG; HUANG, 2007; DOMENICONI et al., 2007).

SSC algorithms can be broadly classified into three main categories:

1. Conventional Soft Subspace Clustering (CSSC): Classic feature weighting clustering algorithms, with all of the clusters sharing the same subspace and a common weight.

2. Independent Soft Subspace Clustering (ISSC): Multiple feature weighting clustering algorithms, with all of the clusters having their weight vectors, i.e., each group has an independent subspace, and the weight vectors are controllable by different mechanisms.

3. Extended Soft Subspace Clustering (ExSSC): Algorithms extending the CSSC or ISSC algorithms with new clustering mechanisms for performance enhancement and special purposes.

In CSSC, clustering is performed identifying first the subspace using some strategies, and then carrying out clustering in the subspace that was obtained to partition the data. This approach is referred to as separated feature weighting (SOETE, 1986; MAKARENKOV; LEGENDRE, 2001; MODHA; SPANGLER, 2003), where data partitioning involves two separate processes: subspace identification and clustering in subspace. Clustering can also be conducted by performing the two methods simultaneously, the approach is known as coupled feature weighting (BOUGUILA, 2009; CHEUNG; ZENG, 2007; DESARBO et al., 1984; HUANG et al., 2005; LI; GAO; JIAO, 2005; TSAI; CHIU, 2008b).

In ISSC, algorithms are developed based on the K-Means model (CHAN et al., 2004; DOMENICONI et al., 2007; GAN; WU; YANG, 2006; JING; NG; HUANG, 2007; JING et al., 2005), FCM model (FRIGUI; NASRAOUI, 2004b; KELLER; KLAWONN, 2000), and probability mixture model, in a process where fuzzy weighting, entropy weighting, or other weighting mechanisms are adopted to implement feature weighting (CHEN; JIANG; WANG, 2008; CHEN; JIANG; WANG, 2012; PENG; ZHANG, 2011).

Finally, ExSSC algorithms can be subdivided into eight subcategories, depending on the strategies used to enhance the CSSC and ISSC algorithms. These subcategories are between-class separation (DENG et al., 2010; LI; CHEN, 2008), evolutionary learning (LU et al., 2011; ZHU; CAO; YANG, 2012), the adoption of new metrics (AMORIM; MIRKIN, 2012; SHEN et al., 2006; WANG et al., 2014b), ensemble learning (DOMENICONI; AL-RAZGAN, 2009; GULLO; DOMENICONI; TAGARELLI, 2013), multi-view learning (CHEN et al., 2013; GAN; NG, 2015), imbalanced clusters (PARVIN; MINAEI-BIDGOLI, 2013), subspace extraction in the transformed feature space (SOETE; CARROLL, 1994; TIMMERMAN et al., 2010), and other approaches such as the reliability mechanism and those used for clustering categorical datasets (AHMAD; DEY, 2011; BAI et al., 2011; CHEN et al., 2016; WANG et al., 2014a).

A category of ISSC algorithms have been developed based on the FCM/K-Means model and entropy weighting. The weighting in this category of algorithms is controllable by entropy. The attribute weighted fuzzy clustering problem is considered as a maximum entropy inference problem (ZHOU et al., 2016), which aims to search for global regularity and obtain the smoothest reconstructions from the available data.

2.3.2.1 FCCI algorithm

The FCCI algorithm (HANMANDLU et al., 2013) is a work of interest for the proposed approach. The algorithm presents a novel color segmentation technique using the fuzzy co-clustering approach in which both objects and features have assigned a membership function. An objective function which includes a multi-dimensional distance function as the dissimilarity measure and entropy as the regularization term is formulated. Despite the matrix $\mathbf{U}$ of membership degrees and the vector $\mathbf{G}$ of prototypes, the algorithm provides:

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

FCCI involves the minimization of the following objective function:

$$\begin{aligned} J_{FCCI}(\mathbf{U}, \mathbf{V}, \mathbf{G}) &= \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 \\ &+ T_u \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \ln(u_{ik}) + T_v \sum_{k=1}^{C}\sum_{j=1}^{P} v_{kj} \ln(v_{kj}) \end{aligned} \tag{2.25}$$

Subject to:

$$\sum_{k=1}^{C}(u_{ik}) = 1 \tag{2.26}$$

$$\sum_{j=1}^{P}(v_{kj}) = 1 \tag{2.27}$$

where C and N represent the number of clusters and data points respectively, u_{ki} is the fuzzy membership, v_{kj} the feature membership and T_u and T_v the weight factors in the entropy terms.

The minimization of the first term in (Equation 2.25) assigns to the object a higher membership value taking into account the cluster center it is closest to and which feature is more relevant for that particular cluster. The inner product $v_{kj}(x_{ij} - g_{kj})^2$ assigns a higher weight to the distance function about the prominent features and a lower weight to irrelevant features. The first term, therefore, denotes the useful square of the Euclidean distance. The second and third entropy regularization terms combine all u_{ki} and v_{kj} separately. These contribute to the fuzziness in the resulting clusters. T_u and T_v are the weighting parameters that specify the degree of fuzziness. Increasing T_u and T_v increases the fuzziness of the groups.

Deriving the update equations

The constrained optimization problem of FCCI can now be defined from Equation 2.25 by applying the Lagrange multipliers λ_i and β_k to constraints Equation 2.26 and Equation 2.27 respectively as shown below.

$$\begin{aligned}\mathcal{L}_{FCCI} &= \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 \\ +T_u \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik}\ln(u_{ik}) &+ T_v \sum_{k=1}^{C}\sum_{j=1}^{P} v_{kj}\ln(v_{kj}) \\ +\sum_{i=1}^{N}\lambda_i(\sum_{k=1}^{C} u_{ik} - 1) &- \sum_{k=1}^{C}\beta_k(\sum_{j=1}^{P} v_{kj} - 1)\end{aligned} \tag{2.28}$$

Taking the partial derivative of $\mathcal{L}_{FCCI}$ in Equation 2.28 with respect to u_{ik} and setting the gradient to zero:

$$\frac{\partial \mathcal{L}_{FCCI}}{\partial u_{ik}} = \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 + T_u(1 + \ln(u_{ik})) + \lambda_i = 0 \tag{2.29}$$

Solving Equation 2.29:

$$u_{ik} = e^{-\frac{\lambda_i}{T_u} - 1 - \frac{\sum_{j=1}^{P} v_{kj}(x_{ij}-g_{kj})^2}{T_u}} \tag{2.30}$$

Because $\sum_{h=1}^{C}(u_{ih}) = 1$ then Equation 2.30 can be expressed like:

$$\sum_{h=1}^{C} u_{ih} = \sum_{h=1}^{C} e^{-\frac{\lambda_i}{T_u} - 1 - \frac{\sum_{j=1}^{P} v_{hj}(x_{ij}-g_{hj})^2}{T_u}} = 1 \tag{2.31}$$

Solving Equation 2.31

$$e^{-\frac{\lambda_i}{T_u} - 1} = \frac{1}{\sum_{h=1}^{C} e^{-\frac{\sum_{j=1}^{P} v_{hj}(x_{ij}-g_{hj})^2}{T_u}}} \tag{2.32}$$

Substituting Equation 2.32 in Equation 2.30 the formula for computing the object membership function update u_{ki} reduces to:

$$u_{ik} = \frac{e^{-\sum_{j=1}^{P}\frac{v_{kj}(x_{ij}-g_{kj})^2}{T_u}}}{\sum_{h=1}^{C} e^{-\sum_{j=1}^{P}\frac{v_{hj}(x_{ij}-g_{hj})^2}{T_u}}} \tag{2.33}$$

Similarly, taking the partial derivative of $\mathcal{L}_{FCCI}$ concerning v_{kj} and set the gradient to zero:

$$\frac{\partial \mathcal{L}_{FCCI}}{\partial v_{kj}} = \sum_{i=1}^{N} u_{ik}(x_{ij} - g_{kj})^2 + T_v(1 + \ln(v_{kj})) + \beta_k = 0 \tag{2.34}$$

Solving Equation 2.34 respect to v_{kj}

$$v_{kj} = e^{-\frac{\beta_k}{T_v}-1-\frac{\sum_{i=1}^{N} u_{ik}(x_{ij}-g_{kj})^2}{T_v}} \tag{2.35}$$

Because $\sum_{h=1}^{P}(v_{kh}) = 1$ the Equation 2.35 can be expressed like:

$$\sum_{h=1}^{P} v_{kh} = \sum_{h=1}^{P} e^{-\frac{\beta_k}{T_v}-1-\frac{\sum_{i=1}^{N} u_{ik}(x_{ih}-g_{kh})^2}{T_v}} = 1 \tag{2.36}$$

Solving Equation 2.36 respect to the expression $e^{-\frac{\beta_k}{T_v}-1}$

$$e^{-\frac{\beta_k}{T_v}-1} = \frac{1}{\sum_{h=1}^{P} e^{-\frac{\sum_{i=1}^{N} u_{ik}(x_{ih}-g_{kh})^2}{T_v}}} \tag{2.37}$$

Substituting Equation 2.37 in Equation 2.35, is obtained the formula for the feature membership function v_{kj} as:

$$v_{kj} = \frac{e^{-\sum_{i=1}^{N}\frac{u_{ik}(x_{ij}-g_{kj})^2}{T_v}}}{\sum_{h=1}^{P} e^{-\sum_{i=1}^{N}\frac{u_{ik}(x_{ih}-g_{kh})^2}{T_v}}} \tag{2.38}$$

Taking the partial derivative of $J_{FCCI}(\mathbf{U}, \mathbf{V}, \mathbf{G})$ with respect to $\mathbf{G}$ and setting the gradient to zero:

$$\frac{\partial J_{FCCI}}{\partial g_{kj}} = v_{kj} \sum_{i=1}^{N} u_{ik} x_{ij} - v_{kj} g_{kj} \sum_{i=1}^{N} u_{ik} = 0 \tag{2.39}$$

Solving Equation 2.39 yields the formula for g_{kj} as:

$$g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{2.40}$$

Optimal partitions $\mathbf{U}^*$ of D can be obtained by solving for $(\mathbf{U}^*, \mathbf{V}^*, \mathbf{G}^*)$ at the local minima of J_{FCCI}.

Proposition 1. *The updated values of u_{ki} given by Equation 2.33 never increase the objective function in every iteration.*

Proof. Consider the objective function as a function of u_{ik} alone.

$$J_{FCCI}(U,V,G) = \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 + T_u \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ik} \ln(u_{ik}) + constant \tag{2.41}$$

where, constant=$T_v \sum_{k=1}^{C} \sum_{j=1}^{P} v_{kj} \ln(v_{kj})$. Also, the product $v_{kj}(x_{ij} - g_{kj})^2$ may be considered as constant.

To prove Proposition 1, it has to be proved that $\mathbf{U}^*$, i.e the updated values of u_{ik} given by Equation 2.33 are the local minima of the objective function $J_{FCCI}(\mathbf{U}^*)$ provided that the constraints in Equation 2.26 and Equation 2.27 are satisfied. For this it is needed prove that the Hessian matrix $\partial^2 J_{FCCI}(\mathbf{U}^*)$ is positive definite.

$$\partial^2 J_{FCCI}(\mathbf{U}^*) = \begin{bmatrix} \frac{\partial^2 J_{FCCI}(\mathbf{U})}{\partial u_{11}\partial u_{11}} & \cdots & \frac{\partial^2 J_{FCCI}(\mathbf{U})}{\partial u_{11}\partial u_{NC}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J_{FCCI}(\mathbf{U})}{\partial u_{NC}\partial u_{11}} & \cdots & \frac{\partial^2 J_{FCCI}(\mathbf{U})}{\partial u_{NC}\partial u_{NC}} \end{bmatrix} = \begin{bmatrix} \frac{T_u}{u_{11}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{T_u}{u_{NC}} \end{bmatrix}$$

At $\mathbf{U}^*$, $u_{ik} \geq 0$ and T_u is always assigned a positive value. Therefore the Hessian matrix $\partial^2 J_{FCCI}(\mathbf{U}^*)$ is positive definite. It is proved the first necessary condition

$$\frac{\partial J_{FCCI}(u_{ik})}{\partial u_{ik}} = 0$$

and the second sufficient condition that $\partial^2 J_{FCCI}(\mathbf{U}^*)$ is positive definite. Therefore u_{ik} updated is indeed a local minimum of $J_{FCCI}(\mathbf{U})$, and it never increases the objective function value. □

Proposition 2. *For every iteration the updated values of v_{kj} given by Equation 2.38 never increases the objective function.*

Proof. Proof is similar to proof of Proposition 1. □

The steps of FCCI algorithm are summarized in Algorithm 3.

The computational complexity for the FCCI algorithm is $O(CNPT)$ according to (LIU et al., 2017), where N is the number of data examples, C is the number of clusters, P is the dimension of the data and T is the number of iterations.

The memory complexity of the algorithm is:

- CN for the fuzzy partition matrix $\mathbf{U}$;
- CP for the matrix of relevance weights $\mathbf{V}$;
- CP for the representative (prototype) vector $\mathbf{G}$.

Algorithm 3 FCCI Algorithm

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

Output:
       The vector of prototypes $\mathbf{G}$;
       The matrix of membership degrees $\mathbf{U}$;
       The relevance weight vectors $\mathbf{V}$.
Initialization
    Set $t = 0$;
    Randomly select $C$ distinct prototypes $\mathbf{g}_k^{(t)} \in D, , k = \{1, .., C\}$ to obtain the vector of prototypes $\mathbf{G} = (\mathbf{g}_1, ..., \mathbf{g}_C)$;
    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 $\sum_{k=1}^{C} u_{ik}^{(t)} = 1$;
    Initialize the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1/P \leq k \leq C \\ 1 \leq j \leq P}}$ with $v_{kj} = 1, \forall k, j$;
    Compute $J_{FCCI}$ according to Equation 2.25.
repeat
    Set $t = t + 1$
    Set $J_{OLD} = J_{FCCI}$
    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 2.40.
    Step 2: weighting.
        Compute the elements $v_{kj}$ of the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ according to Equation 2.38.
    Step 3: allocation.
        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 2.33.
        Compute $J_{FCCI}$ according to Equation 2.25.
        Set $J_{NEW} = J_{FCCI}$.
until $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$
```

2.4 Chapter summary

This chapter presented concepts like partitional and fuzzy clustering, objective function optimization and was described clustering problem as an essential data processing technique. Several methods of the literature were reviewed and commented briefly. The advantages and limitations of the approaches presented were also discussed. The regularization term benefit was analyzed using the maximum entropy approach because of the advantages over conventional algorithms. The FCM-ER algorithm was described and analyzed.

Classical fuzzy clustering algorithms treat all features equally in deciding objects cluster membership. This disadvantage is not desirable in some applications, such as high-dimensions data clustering, where the cluster structure in the dataset is often limited to a subset of the features instead of the entire feature set. In this chapter was discussed a solution to this problem including the appropriate weight of the variables learning in the clustering process. For this purpose, the transformation techniques, feature selection and soft subspace clustering, considered as an extension of the standard feature weighting clustering, were shown and discussed. The so-called Fuzzy Co-Clustering of Images was revised due to the importance it has for the understanding of proposed work.

3 FUZZY C-MEANS CLUSTERING ALGORITHM WITH ADAPTIVE EUCLIDEAN DISTANCES AND ENTROPY REGULARIZATION

This chapter provides a description of proposed partitional Fuzzy C-Means clustering algorithm with Adaptive euclidean distances and Entropy Regularization AFCM-ER in the objective function. This method performs fuzzy clustering using more intuitive relevance weight constraint for SSC than current state-art.

The key in the information theory is a measure of uncertainty associated with the value of a random variable. The information entropy (SHANNON, 1948) was designed to have the following properties: (i) a random variable which follows a broad probability distribution reveals more uncertainty than a one which follows a strongly peaked distribution; (ii) the uncertainty measure should be additive concerning independent sources introducing uncertainty. Assuming X is a discrete random variable taking values $\{X_1, ..., X_i, ..., X_n\}$, according to a probability distribution $p(X_i)$ and $S(X)$ is the set of values that X can take. A function fulfilling the above requirements is given by Equation 3.1 as mentioned in the previous chapter.

$$E(X) = -\sum_{X \in S(X)} p(X) \log(p(X)) \tag{3.1}$$

Assume that we are given a random variable X with an unknown probability distribution $p(X_i)$. Additionally, we are also given a prior information about the random variable: the expected value a (also defined as $E[f(X_i)]$) of some property, here described by the function $f(X_i)$.

$$a = E[f(X_i)] = \sum_{i=1}^{n} p(X_i) f(X_i) \tag{3.2}$$

The question is: what is the unbiased inference about the distribution $p(X_i)$? In other words, what is the distribution which does not reduce the amount of uncertainty about the random variable?

The maximum entropy principle states that the probability distribution should maximize the information entropy subject to the prior knowledge about the random variable. If there is no previous information, the solution is, quite intuitively, a uniform distribution assigning the same probability to every value of the random variable which leaves you the largest remaining uncertainty consistent with your constraints. That way you have not introduced any additional assumptions or biases into your calculations.

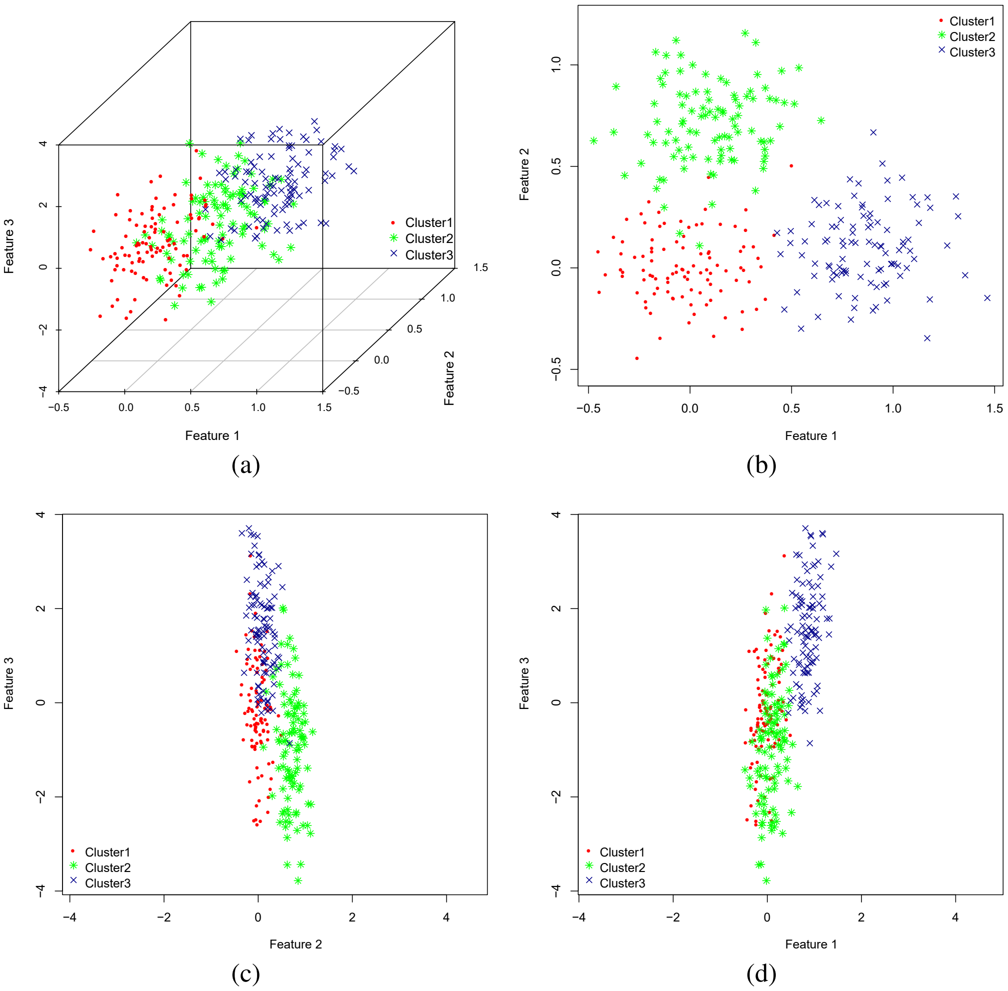
Several maximum entropy clustering algorithms have been proposed in the past few decades. One of them is the algorithm FCM-ER seen in subsection 2.2.1.1 proposed by Sadaaki (SADAAKI; MASAO, 1997). Despite the usefulness of this algorithm in comparison with the conventional FCM, it still assumes that the variables have the same importance for the clustering task. However, in most areas, we typically have to deal with high-dimensional datasets. In this way, some variables may have a small weight for the clustering process or even be irrelevant, and among the relevant variables, some may have greater weights than others. Also, the weight of each variable in the construction of each cluster may be different, i.e., each cluster may have a different set of important variables. An example is described in Figure 4, where Feature 1 and Feature 2 variables are important for all clusters and the variable Feature 3 represents noise. The synthetic dataset containing 3 groups is show in Figure 4 (a) and each possible subspace are show Figure 4 (b)-(d). As seen in Figure 4 (b), relevant variable weighting can achieve better description of a dataset. Several maximum entropy clustering algorithms and their variants are available in the literature to solve this problem (ZHOU et al., 2016; DENG et al., 2010; CHEN et al., 2012). An example is the FCCI algorithm presented by (HANMANDLU et al., 2013) seen in subsection 2.3.2.1, that includes a multi-dimensional distance function as the dissimilarity measure and entropy of objects and variables simultaneously as the regularization term in the objective function. This method is a FCM-type algorithm with entropy regularization and automatic variable selection and uses two positive regularizing parameters T_u and T_v that specify the clusters fuzziness degree.

FCM-ER and FCCI algorithms are highly sensitive to parameter values selection. For this reason, finding the best parameter set for the algorithms is fundamental for the clustering process. Selecting good parameters values will improve the algorithms performance. For FCM-ER algorithm, there is only one regularization parameter (T_u). If this parameter is searched in a given set including n values, then the computational cost for grid search is $O(n)$. Meanwhile, for FCCI algorithm, there are two regularization parameters (T_u and T_v). If these two parameters are searched in the given sets with n_1 values and n_2 values, respectively, then the computational cost for grid search is $O(n_1 n_2)$. In conclusion, find the best parameters for FCCI algorithm is more difficult than for the FCM-ER algorithm.

A new FCM-type algorithm for SSC based on adaptive Euclidean distances (DIDAY, 1977), and the entropy of the objects as the regularization term in the objective function is defined. The proposed is an iterative three-steps relocation algorithm which determines:

- A matrix **U** of membership degrees;
- A vector **G** of representatives (prototypes) for the fuzzy clusters;
- A matrix **V** of relevance weights of the variables into the fuzzy clusters.

By optimizing an adequacy criterion that measures the fitting between the fuzzy clusters and their representatives. The relevance weights of the variables change at each iteration of the algorithm and differ from one fuzzy group to another. This work has the advantage that uses adaptive Euclidean distances as dissimilarity measure improving the clustering process and just requires the tuning of one parameter in comparison to the FCCI algorithm.



(a)

(b)

(c)

(d)

Figure 4 – Plot of (a) synthetic 3 classes dataset, (b) projection into Feature 1 and Feature 2 subspace, (c) projection into Feature 2 and Feature 3 subspace and (d) projection into Feature 1 and Feature 3 subspace.

3.1 The AFCM-ER algorithm objective function

The vector **G** of prototypes, the matrix **V** of relevance weights of the variables into the fuzzy clusters, and the fuzzy partition **U** are obtained interactively in three steps (representation, weighting, and allocation) by the minimization of a suitable objective function hereafter denoted as $J_{AFCM-ER}$. In this work, it is defined an adaptive distance between the objects and the prototype as:

$$d_{v_k}(\mathbf{x}_i, \mathbf{g}_k) = \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 \tag{3.3}$$

where $d_{v_k}(\mathbf{x}_i, \mathbf{g}_k)$ is a squared Euclidean distance between $\mathbf{x}_i$ and $\mathbf{g}_k$ parameterized by the vector of relevance weights of the variables $\mathbf{v}_k$ on the fuzzy cluster k. The relevance weights change at each iteration of the algorithm, and differ from one fuzzy cluster to another. The advantage of this adaptive distances is that the algorithm is able to recognize clusters of different shapes and sizes because introduce the proper attribute weight into the clustering process.

The proposed clustering criterion measures the heterogeneity of the fuzzy partition **U** as the sum of the heterogeneity in each fuzzy cluster defined as:

$$\begin{aligned} J_{AFCM-ER} &= \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})\, d_{v_k}(\mathbf{x}_i, \mathbf{g}_k) \\ &+ T_u \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik}) \ln(u_{ik}) \end{aligned} \tag{3.4}$$

Where T_u is a positive regularizing parameter, with $T_u > 0$. This optimization problem (Equation 3.4) is subject to:

$$\sum_{k=1}^{C}(u_{ik}) = 1 \qquad (3.5) \qquad\qquad \prod_{j=1}^{P}(v_{kj}) = 1 \qquad (3.6)$$

The minimization of the first term in Equation 3.4 assigns to the object a higher membership value taking into account the cluster center it is closest to and which feature is more relevant for that particular cluster. The inner product $v_{kj}(x_{ij} - g_{kj})^2$ assigns a higher weight to the distance function about the prominent features and a lower weight to irrelevant features. In this objective function, the first distance-based term defines the shape and size of the clusters and encourages agglomeration, while the second term is the negative entropy of the elements and is used to control the membership degree u_{ik}. These contribute to the fuzziness in the resulting clusters. T_u is the weighting parameters that specify the degree of fuzziness. Increasing T_u increases the fuzziness of the groups. Proposed objective function $J_{AFCM-ER}$, is similar to J_{FCM-ER} (Equation 2.13) with an adaptive euclidean distance, to perform SSC, rather than simple Euclidean distance used in FCM-ER algorithm. The used of Euclidean adaptive distance introduce another constraint for relevance weights of variables, like in FCCI algorithm. However, unlike Equation 2.27, a product constraint (Equation 3.6) is used to obtain better (not bounded) and more intuitive relevance weights.

3.1.1 The optimization steps of the AFCM-ER algorithm

In the AFCM-ER algorithm, from an initial solution, the vector **G** of prototypes, the matrix **V** of relevance weights of the variables on the fuzzy clusters, and the matrix of membership degrees **U** are obtained interactively in three steps (representation, weighting and allocation) by the minimization of the objective function $J_{AFCM-ER}$ (Equation 3.4).

Due to the convexity of the optimization problem given by Equation 3.4, the computation of the matrices **U** and **V** can be obtained applying the Lagrange multipliers λ_i and β_k to constraints in Equation 3.5 and Equation 3.6 as follows:

$$\begin{aligned}\mathcal{L}_{AFCM-ER} &= J_{AFCM-ER} - \sum_{k=1}^{C} \beta_k \left[\prod_{j=1}^{P} v_{kj} - 1 \right] \\ &+ \sum_{i=1}^{N} \lambda_i \left[\sum_{k=1}^{C} u_{ik} - 1 \right] \end{aligned} \tag{3.7}$$

Representation

This step provides an optimal solution to the computation of the fuzzy clusters prototypes. During the representation step of AFCM-ER, the matrix **V** of relevance weights of the variables on fuzzy clusters, and the matrix of membership degrees **U** are kept fixed. Taking the partial derivative which minimizes the criterion $J_{AFCM-ER}$ given in Equation 3.4 concerning the cluster centroids g_{kj} and set the gradient to zero we have:

$$\frac{\partial J_{AFCM-ER}}{\partial g_{kj}} = v_{kj} \sum_{i=1}^{N} u_{ik} x_{ij} - v_{kj} g_{kj} \sum_{i=1}^{N} u_{ik} = 0 \tag{3.8}$$

Solving Equation 3.8 yields the formula for g_{kj} as:

$$g_{kj} = \frac{\sum_{i=1}^{N} u_{ik} x_{ij}}{\sum_{i=1}^{N} u_{ik}} \tag{3.9}$$

Weighting

This step provides an optimal solution to the computation of the relevance weights of the variables in each fuzzy cluster.

During the weighting step of AFCM-ER, the vector **G** of prototypes and the matrix of membership degrees **U** are kept fixed. The objective function (Equation 3.4) is optimized with respect to the relevance weights.

Proposition 1. *The vector of weights* $\mathbf{v}_k = (v_{k1}, \ldots, v_{kP})$ *which minimizes the criterion* $J_{AFCM-ER}$ *given in Equation 3.4 under* $v_{kj} > 0 \; \forall \; k, j$ *and* $\prod_{j=1}^{P} v_{kj} = 1 \; \forall \; k$*, has its components* $v_{kj} (k = 1, ...C, j = 1, ..., P)$ *computed according to the following expression:*

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

Proof. The constrained optimization problem of AFCM-ER can now be defined from Equation 3.4 by applying the Lagrange multipliers λ_i and β_k to constraints Equation 3.5 and Equation 3.6 respectively as shown in Equation 3.7.

Taking the partial derivative of $\mathcal{L}_{AFCM-ER}$ in Equation 3.7 concerning v_{kj} and set the gradient to zero it has:

$$\frac{\partial \mathcal{L}_{AFCM-ER}}{\partial v_{kj}} = \sum_{i=1}^{N} u_{ik} (x_{ij} - g_{kj})^2 - \frac{\beta_k}{v_{kj}} = 0 \tag{3.11}$$

From Equation 3.11 is obtained that

$$v_{kj} = \frac{\beta_k}{\sum_{i=1}^{N} u_{ik}(x_{ij} - g_{kj})^2} \tag{3.12}$$

Substituting Equation 3.12 in Equation 3.6 we have

$$\prod_{w=1}^{P} v_{kw} = \prod_{w=1}^{P} \frac{\beta_k}{\sum_{i=1}^{N} u_{ik}(x_{iw} - g_{kw})^2} = 1 \tag{3.13}$$

Solving Equation 3.13 we have:

$$\beta_k = \left\{ \prod_{w=1}^{P} \sum_{i=1}^{N} u_{ik}(x_{iw} - g_{kw})^2 \right\}^{\frac{1}{P}} \tag{3.14}$$

Substituting Equation 3.14 in Equation 3.12 the relevance weights of the variables in each fuzzy cluster are calculated as:

$$v_{kj} = \frac{\{\prod_{w=1}^{P} \sum_{i=1}^{N} u_{ik}(x_{iw} - g_{kw})^2\}^{\frac{1}{P}}}{\sum_{i=1}^{N} u_{ik}(x_{ij} - g_{kj})^2} \tag{3.15}$$

If we rewrite the criterion $J_{AFCM-ER}$ as

$$J_{AFCM-ER}(\mathbf{v}_1, ..., \mathbf{v}_C) = \sum_{k=1}^{C} J_k(\mathbf{v}_k) \tag{3.16}$$

with

$$J_k(\mathbf{v}_k) = J_k(v_{k1}, ..., v_{kP}) = \sum_{j=1}^{P} v_{kj} J_{kj} \tag{3.17}$$

where $J_{kj} = \sum_{i=1}^{N} u_{ik}(x_{ij} - g_{kj})^2$ and $T_u \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik} \ln(u_{ik})$ are seem like a constant. Thus, an extreme minimum value of J_k is reached when

$$J_k(v_{k1}, ..., v_{kP}) = P \left\{ \prod_{j=1}^{P} J_{kj} \right\}^{\frac{1}{P}} \tag{3.18}$$

As $J_k(1, ..., 1) = \sum_{j=1}^{P} J_{kj}$, and as it is well known that the arithmetic mean is greater than the geometric mean, i.e., $\frac{1}{P} \sum_{j=1}^{P} J_{kj} \geq \{\prod_{j=1}^{P} J_{kj}\}^{\frac{1}{P}}$, (a minimum is obtained in this case only if $J_{k1} = \ldots = J_{kP}$. Also, substituting Equation 3.15 in Equation 3.17, $P\{\prod_{j=1}^{P} J_{kj}\}^{\frac{1}{P}}$ is obtained so we conclude that this extremum is a minimum. □

Allocation

This step provides an optimal solution to the computation of the matrix of membership degrees of the objects into the fuzzy clusters.

During the allocation step of AFCM-ER, the vector **G** of prototypes, and the matrix **V** of relevance weights of the variables on fuzzy clusters are kept fixed. The objective function (Equation 3.4) is optimized concerning the membership degrees.

Taking the partial derivative of $\mathcal{L}_{AFCM-ER}$ in Equation 3.7 concerning u_{ik} and set the gradient to zero it has:

$$\frac{\partial \mathcal{L}_{AFCM-ER}}{\partial u_{ik}} = \sum_{j=1}^{P} v_{kj}(x_{ij} - g_{kj})^2 + T_u(1 + \ln(u_{ik})) + \lambda_i = 0 \tag{3.19}$$

Solving Equation 3.19

$$u_{ik} = e^{\frac{-\sum_{j=1}^{P} v_{kj}(x_{ij}-g_{kj})^2}{T_u}} e^{\frac{-\lambda_i}{T_u}-1} \tag{3.20}$$

If we have the constraint Equation 3.5 then we can express Equation 3.20 like

$$\sum_{w=1}^{C} u_{iw} = \sum_{w=1}^{C} \left[e^{\frac{-\sum_{j=1}^{P} v_{wj}(x_{ij}-g_{wj})^2}{T_u}} e^{\frac{-\lambda_i}{T_u}-1} \right] = 1 \tag{3.21}$$

Solving Equation 3.21 we have

$$e^{\frac{-\lambda_i}{T_u}-1} = \frac{1}{\sum_{w=1}^{C} e^{\frac{-\sum_{j=1}^{P} v_{wj}(x_{ij}-g_{wj})^2}{T_u}}} \tag{3.22}$$

Substituting Equation 3.22 in Equation 3.20 the formula for computing the object membership function u_{ik} reduces to:

$$u_{ik} = \frac{e^{-\frac{\sum_{j=1}^{P} v_{kj}(x_{ij}-g_{kj})^2}{T_u}}}{\sum_{w=1}^{C} e^{-\frac{\sum_{j=1}^{P} v_{wj}(x_{ij}-g_{wj})^2}{T_u}}} \tag{3.23}$$

Proposition 2. *The updated values of u_{ik} given by Equation 3.23 never increase the objective function in every iteration.*

Proof. Consider the objective function as a function of u_{ik} alone.

The product $v_{kj}(x_{ij} - g_{kj})^2$ may be considered as constant. To prove Proposition 2, it has to be proved that $\mathbf{U}^*$, i.e the updated values of u_{ik} given by Equation 3.23, are a local minima of the objective function $J_{AFCM-ER}(\mathbf{U}^*)$ provided such that the constraints in Equation 3.5 and Equation 3.6 are satisfied. For this it is needed to prove that the Hessian matrix $\partial^2 J_{AFCM-ER}(\mathbf{U}^*)$ is positive definite.

$$\partial^2 J_{AFCM-ER}(\mathbf{U}^*) = \begin{bmatrix} \frac{\partial^2 J_{AFCM-ER}(\mathbf{U})}{\partial u_{11} \partial u_{11}} & \cdots & \frac{\partial^2 J_{AFCM-ER}(\mathbf{U})}{\partial u_{11} \partial u_{NC}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J_{AFCM-ER}(\mathbf{U})}{\partial u_{NC} \partial u_{11}} & \cdots & \frac{\partial^2 J_{AFCM-ER}(\mathbf{U})}{\partial u_{NC} \partial u_{NC}} \end{bmatrix} = \begin{bmatrix} \frac{T_u}{u_{11}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{T_u}{u_{NC}} \end{bmatrix}$$

At $\mathbf{U}^*$, $u_{ik} \geq 0$ and T_u is always assigned a positive value. Therefore the Hessian matrix $\partial^2 J_{AFCM-ER}(\mathbf{U}^*)$ is positive definite. It has been proved the first necessary condition

$$\frac{\partial J_{AFCM-ER}(u_{ik})}{\partial u_{ik}} = 0$$

and the second sufficient condition that $\Delta^2 J_{AFCM-ER}(\mathbf{U}^*)$ is positive definite. Therefore u_{ik}^* update is indeed a local minimal of $J_{AFCM-ER}(\mathbf{U})$ and it never increases the objective function value. □

Proposition 1 and Proposition 2 prove that the updated equations of AFCM-ER point to a local minimal of the objective function.

3.1.2 The algorithm

The K-Means algorithm can be viewed as an EM algorithm and it is convergent because each EM algorithm is convergent (CAMASTRA; VERRI, 2005). As the AFCM-ER algorithm is a modified version of the classical K-Means algorithm its convergence was proved above. The representation, weighting, and allocation steps are repeated until the convergence of theAFCM-ER algorithm is reached. The Algorithm 4 summarizes these steps.

According to the definitions in section 3.1, the computational complexity of AFCM-ER is $O(TCNP)$, where T is the total number of iterations required, and N, P, C indicate the number of data objects, data dimensions and clusters respectively. For the storage, is needed memory to keep:

- The data objects (NP);
- The membership matrix (CN);
- The attribute weight matrix (CP).

Algorithm 4 AFCM-ER Algorithm

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

Output:
        The vector of prototypes $\mathbf{G}$;
        The matrix of membership degrees $\mathbf{U}$;
        The relevance weight vectors $\mathbf{V}$.
Initialization
    Set $t = 0$;
    Randomly select $C$ distinct prototypes $\mathbf{g}_k^{(t)} \in D$ with $k = \{1, ..., C\}$ to obtain the vector of prototypes $\mathbf{G} = (\mathbf{g}_1, ..., \mathbf{g}_C)$;
    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 $\sum_{k=1}^{C} u_{ik}^{(t)} = 1$;
    Initialize the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ with $v_{kj} = 1$, $\forall k, j$;
    Compute $J_{AFCM-ER}$ according to (Equation 3.4).
repeat
    Set $t = t + 1$
    Set $J_{OLD} = J_{AFCM-ER}$
    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 3.9).
    Step 2: weighting.
        Compute the elements $v_{kj}$ of the matrix of relevance weights $\mathbf{V} = (v_{kj})_{\substack{1 \leq k \leq C \\ 1 \leq j \leq P}}$ according to (Equation 3.15).
    Step 3: allocation.
        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 3.23.
        Compute $J_{AFCM-ER}$ according to Equation 3.4.
        Set $J_{NEW} = J_{AFCM-ER}$.
until $|J_{NEW} - J_{OLD}| < \varepsilon$ or $t > T$
```

3.2 Chapter summary

This chapter presented the main contribution of this dissertation, a new FCM-type algorithm for Soft Subspace Clustering based on adaptive Euclidean distances and entropy regularization term in the objective function. The proposed method was developed considering the fuzzy clustering and the clustering in the feature space considering the constraint that the product of the weights of the variables in each group (local adaptive distances) must be equal to one. This type of dissimilarity measure is adequate to learn the variables weights during the clustering process which lead to an improvement in the performance of the algorithms.

A new objective function was formulated, and update rules are derived. For AFCM-ER algorithm was obtained an expression for the best prototypes of the clusters, the best vectors of variable weights and the best fuzzy partition matrix. Convergence properties of the proposed algorithm were also demonstrated.

In the next chapter are presented a set of experiments performed with both simulated and real datasets to demonstrate the effectiveness of the proposed method. The algorithm is also tested for color image segmentation.

4 EXPERIMENTAL EVALUATION

This chapter provides the performance evaluation of the proposed algorithm and comparison with literature methods. A series of experiments are performed with synthetic and real datasets.

The clustering methods are used for different real applications like image processing, data mining, pattern recognition, statistical analysis, and so on. For illustrative purpose, the proposed AFCM-ER algorithm is experimented with image datasets to evaluate its performance in color image segmentation as in (HANMANDLU et al., 2013).

4.1 Experimental setting

To evaluate the performance of the proposed algorithm, several FCM-type clustering algorithms (the FCM, FCM-ER and FCCI algorithms) are chosen for comparative analysis. A series of experiments are performed with various datasets. All data were previously standardized to have zero mean and a standard deviation of one.

The best clustering performance indicated by some index is not always consistent with others indexes, therefore, is necessary to evaluate the performance of a clustering algorithm with different metrics. As result, given a membership matrix **U**, with N objects and C clusters, four measures were used to quantify the separation and the compactness of the clusters according the fuzzy partition: Modified Partition Coefficient (MPC), the Partition Entropy coefficient (PE), Simplified Fuzzy Silhouette (SFS) and Hullermeier and Rifqi index (FRHR).

From the fuzzy partition $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_C)$ is obtained a hard partition $\mathbf{Q} = (\mathbf{Q_1}, ..., \mathbf{Q_C})$, where the cluster $\mathbf{Q_k}$, $k = 1..C$, is defined as:

$$\mathbf{Q_k} = \{i \in \{1, ..., N\} : u_{ik} \geq u_{im}, \forall m \in \{1, ..., C\}\}$$

Hard cluster partitions $\mathbf{Q}_k$ were compared with the known a priori class partition according to the Adjusted Rand Index (ARI), the F-Measure (FM), the Overall Error Rate of Classification (OERC) and the Simplified Crisp Silhouette (SCS).

MPC, PE, SCS, SFS evaluate clustering results without the use of any external information. The evaluation provided by these measures is based on the data itself and its partitioning, as provided by a clustering algorithm. MPC will measures the amount of “overlap” between clusters and PE index the fuzziness of the cluster partition. SCS and SFS measure how close objects are in a cluster and how separated the clusters are, for hard and fuzzy partitions respectively.

ARI, FM, OERC, FRHR are preferable when ground-truth labels are available (DOM, 2002). The ground-truth consists of class labels assigned to each data point. The ideal clustering is selected based on how well the cluster labels produced by the algorithm match to the ground-truth labels. External measures are used to compare the similarity of the two clustering results.

ARI, FM, OERC are measures of correspondence between two partitions of the same data. FRHR is a Rand index (RAND, 1971) extension used to compare two fuzzy partitions.

Each of these measures are described in details in next sections.

4.1.1 Partition Coefficient

Given a membership matrix **U**, with N objects and C groups, Bezdek (BEZDEK, 2013) attempted to define a performance measure based on minimizing the overall content of pairwise fuzzy intersection in the partition matrix U. He proposed a cluster validity index for fuzzy clustering named Partition Coefficient (PC). The index was defined as

$$V_{PC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}(u_{ik})^2 \tag{4.1}$$

The PC index indicates the average relative amount of membership sharing done between pairs of fuzzy subsets in **U**, by combining into a single number, the average contents of pairs of fuzzy algebraic products. The index values range in $\left[\frac{1}{C}, 1\right]$, where C is the number of clusters, the closer the value of PC to 1, the crisper the clustering is.

4.1.2 Partition Entropy coeficient

Bezdek proposed the PE (BEZDEK, 1975) defined as:

$$V_{PE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} u_{ik}\log(u_{ik}) \tag{4.2}$$

The PE index is a scalar measure of the amount of fuzziness in a given **U**. The PE index values range in $[0, \log(C)]$, the closer the value of PE to 0, the crisper the clustering is. The index value close to the upper bound indicates the absence of any clustering structure in the datasets or inability of the algorithm to extract it.

4.1.3 Modified Partition Coefficient

Both PC and PE possess monotonic evolution tendency with the number of groups (C). According to (BEZDEK, 2013), the limitation can be attributed to their apparent monotonicity and an extent, to the heuristic nature of the rationale underlying its formulation. Modification of the V_{PC} index can reduce the monotonic tendency and is define as:

$$V_{MPC} = 1 - \frac{C}{C-1}(1 - V_{PC}) \tag{4.3}$$

The range of MPC is the unit interval $[0, 1]$, where MPC=0 corresponds to maximum fuzziness and MPC=1 to a hard partition.

4.1.4 Crisp silhouette

To define this criterion, consider a data object $\mathbf{x}_i \in D$ belonging to cluster $k \in \{1, ..., C\}$. In the context of crisp partitions produced by a prototype-based clustering algorithm, i.e., C-Means, etc., this means that object $\mathbf{x}_i$ is closer to the prototype of cluster k ($\mathbf{g}_k$) than to any other prototype. In more general context, the membership of the i-th object to the k-th fuzzy cluster (u_{ik}) is higher than the membership of this object to any other fuzzy cluster, i.e., $u_{ik} > u_{iq}$ for every $q \in \{1, ..., C\}, q \neq k$. Let the average distance of object $\mathbf{x}_i$ to all other objects belonging to cluster k be denoted by a_{ki}. Also, let the average distance of this object to all objects belonging to another cluster q, $q \neq k$, be called d_{qi}. Finally, let b_{ki} be the minimum d_{qi} computed over $q = 1, .., C, q \neq k$, which represents the dissimilarity of object $\mathbf{x}_i$ to its closest neighboring cluster. Then, the silhouette of object $\mathbf{x}_i$ is defined as:

$$s_i = \frac{b_{ki} - a_{ki}}{\max\{a_{ki}, b_{ki}\}} \tag{4.4}$$

where the denominator is used just as a normalization term. The higher s_i, the better the assignment of object $\mathbf{x}_i$ to cluster k. In case k is a singleton, i.e., if it is constituted uniquely by object $\mathbf{x}_i$, then the silhouette of this object is defined as $s_i = 0$. This prevents the Crisp Silhouette (CS), defined as the average of s_i over $i = 1, ..., N$, Equation 4.5, to find the trivial solution $C = N$, with each object of the dataset forming a cluster on its own. This way, the best partition is achieved when CS is maximized, which implies minimizing the intra-cluster distance (a_{ki}) while maximizing the inter-cluster distance (b_{ki}).

$$CS = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{4.5}$$

Remark. *An issue with the CS measure is that it depends on the highly intensive computation of all distances among all data objects. In order to get around this problem (HRUSCHKA; CAMPELLO; CASTRO, 2006; HRUSCHKA; CASTRO; CAMPELLO, 2004) propose replace the terms a_{ki} and b_{ki} in Equation 4.5 with simplified versions of them based on the distances between the objects and the prototypes of the corresponding clusters. This modification has shown not to degrade accuracy while being able to significantly reduce the computational burden from $O(N^2)$ to $O(N)$. Moreover, it does not change the dependency of CS on average distances (in this case represented by the prototypes), which is a desirable property concerning robustness to noise in the data. For these reasons, the prototype-based version of CS described above is adopted in this work. Finally, the Simplified Crisp Silhouette SCS takes its values from the* $[-1, 1]$ *(1 is the best and -1 is the worse result)*

4.1.5 Simplified Fuzzy Silhouette

Both the original and prototype-based versions of the CS do not make explicit use of the fuzzy partition matrix in their calculations. In those cases, the fuzzy partition matrix $\mathbf{U}$ is used only to impose on the dataset a crisp partition **Q** to which the CS measure can be applied. Specifically, **Q** is such that $q_{ik} = 1$ if $i = \arg\max_l\{u_{il}\}$ and $q_{ik} = 0$ otherwise. Consequently, CS may not be able to discriminate between overlapped data clusters, even if these clusters have their own (distinct) regions with higher data densities, since it neglects information contained in the fuzzy partition matrix **U** on degrees to which clusters overlap one another. This information can be used to reveal those regions with high data densities by stressing the importance of data objects concentrated in the neighborhood of the cluster prototypes while reducing the importance of objects lying in overlapping areas. To do so, a generalized silhouette criterion, named Fuzzy Silhouette (FS) (CAMPELLO; HRUSCHKA, 2006), is defined as follows:

$$FS = \frac{\sum_{i=1}^{N}(u_{ik} - u_{iq})^{\gamma} s_i}{\sum_{i=1}^{N}(u_{ik} - u_{iq})^{\gamma}} \tag{4.6}$$

where s_i is the silhouette of object $\mathbf{x}_i$ according to Equation 4.4, u_{ik} and u_{iq} are the first and second largest elements of the i-th column of the fuzzy partition matrix, respectively, and $\gamma \geq 0$ is a weighting coefficient. If s_i is calculated according to Simplified Crisp Silhouette (SCS) then fuzzy measure is named Simplify Fuzzy Silhouette (SFS). There is an important aspect regarding Equation 4.6 that deserves particular attention. It differs from Equation 4.5 for being a weighted average (instead of an arithmetic mean) of the individual silhouettes given by Equation 4.4. The weight of each term is determined by the difference between the membership degrees of the corresponding object to its first and second best matching fuzzy clusters, respectively. This way, an object in the near neighborhood of a cluster prototype is given more importance than another object located in an overlapping area (where the membership degrees of the objects to two or more fuzzy clusters are similar). The SFS index values range in $[-1, 1]$, where good partitions are expected to yield larger values.

4.1.6 Adjusted Rand Index

The Adjusted Rand Index (ARI) is the corrected-for-chance version of the Rand index (RAND, 1971). Though the Rand Index may only yield a value between 0 and 1, the ARI can yield negative values if the index is less than the expected index (HUBERT; ARABIE, 1985).

The contingency table

Given a set D of N elements, and two groupings or partitions (i.e. clusterings) of these points, namely $\mathbf{X} = \{C_1, ..., C_r\}$ and $\mathbf{Y} = \{Q_1, ..., Q_s\}$, the overlap between $\mathbf{X}$ and $\mathbf{Y}$ can be summarized in a contingency table $[n_{ij}]$ where each entry n_{ij} denotes the number of objects in common between C_i and Q_j: $n_{ij} = |C_i \bigcap Q_j|$.

The adjusted form of the Rand Index, the ARI, is calculated as:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}} \tag{4.7}$$

X/Y	$Q_1, Q_2 ... Q_s$	Sums
C_1	$n_{11}, n_{12} ... n_{1s}$	a_1
$\vdots$	$\vdots \; \vdots \; \ddots$	
C_r	$n_{r1}, n_{r2} ... n_{rs}$	a_r
Sums	$b_1, b_2 ... b_s$	n

where n_{ij}, a_i , b_j are values from the contingency table. The ARI takes its values on the interval $[-1, 1]$, in which the value 1 indicates perfect agreement between partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance. The ARI index evaluates the degree of agreement (similarity) between an a priori partition and a partition provided by a clustering method. Also, the ARI index is not sensitive to the number of clusters in the partitions or the objects distribution in the groups.

4.1.7 Campello

Campello (CAMPELLO, 2007) presents a method for fuzzifying the indices of Rand, Jaccard, Fowlkes-Mallow, Hubert and (one version of) the adjusted Rand. Campello's scheme is based on writing the equations in an equivalent form using (cardinalities of) intersections of the crisp subsets $D \times D$ corresponding to each of the four totals, and then replacing the crisp sets with fuzzy ones. Campello's generalization for the Rand index equation $a = \frac{1}{2} \sum_{i=1}^{r} \sum_{j=1}^{s} n_{ij}(n_{ij} - 1)$ is:

$$a = \sum_{j=1}^{i-1} \sum_{i=2}^{N} \left(\left(\vee_{k=1}^{r} (u_{ki} \wedge u_{kj}) \right) \wedge \left(\vee_{k=1}^{s} (v_{ki} \wedge v_{kj}) \right) \right) \tag{4.8}$$

Campello's generalization of Rand's index [i.e., fuzzy Rand of Campello (FRC)], $s_{FRC}(U, V)$, is valued in $[0, 1]$; however, $s_{FRC}(\mathbf{U}, \mathbf{V}) = 1$ on $\mathbf{U} = \mathbf{V}$ is only necessary in the fuzzy case and not necessary and sufficient for the index to take the value 1. Curiously, Campello states that "the fuzzy Rand index cannot be seen as a general measure for comparing two fuzzy partitions," even though his index is well defined for this case. Instead, he advocates using it only as a measure to compare a fuzzy partition against a hard partition (possibly with a different number of categories).

4.1.8 Hullermeier and Rifqi

Hullermeier and Rifqi (HULLERMEIER; RIFQI, 2009) argue that Campello's fuzzy Rand index (CAMPELLO, 2007) is in some sense defective because it is not a metric. Instead, their generalization is guided by the fact that Rand's index counts the number of paired agreements divided by the total number of possible pairs, and this leads them to a direct generalization of the Rand index:

$$V_{FRHR}(\mathbf{U}_1, \mathbf{U}_2) = 1 - \left[\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |||\mathbf{U}_2^i - \mathbf{U}_2^j|| - ||\mathbf{U}_1^i - \mathbf{U}_1^j|||/\binom{N}{2} \right] \tag{4.9}$$

The FRHR index is a direct generalization of the original Rand index to the case where $\mathbf{U}_1$ and $\mathbf{U}_2$ are both fuzzy partitions. The FRHR index takes its values from the interval $[0, 1]$, in which the amount 1 indicates perfect agreement between fuzzy partitions, whereas values near 0 correspond to cluster agreement found by chance.

4.1.9 F-Measure

The traditional F-Measure FM between a class $P_i(i = 1, ..., C)$ and a cluster $P_k(k = 1, ..., C)$ is the harmonic mean between Precision and Recall.

$$FM = 2\frac{Precision(P_i, P_k)Recall(P_i, P_k)}{Precision(P_i, P_k) + Recall(P_i, P_k)} \tag{4.10}$$

The precision between a class P_i and a cluster P_k is the fraction between the number of objects in the class P_i and the cluster P_k and the number of objects in the cluster P_k.

$$Precision(P_i, P_k) = \frac{n_{ik}}{a_k} \tag{4.11}$$

The recall is the fraction of the number of objects in the class P_i and the cluster P_k and the number of objects in class P_i.

$$Recall(P_i, P_k) = \frac{n_{ik}}{b_i} \tag{4.12}$$

The FM (BAEZA-YATES; RIBEIRO-NETO et al., 1999) between a priori partition $\mathbf{P} = P_1, ..., P_i, ..., P_C$ and a hard partition $\mathbf{Q} = Q_1, ..., Q_k, ..., Q_C$ given by a clustering algorithm is defined as:

$$FM(\mathbf{P}, \mathbf{Q}) = \frac{1}{N} \sum_{i=1}^{C} b_i \max_{1 \le k \le C} FM(P_i, Q_k) \tag{4.13}$$

The FM is defined as the harmonic mean of pairwise precision and recall and takes its values from the $[0, 1]$ interval, in which the amount 1 indicates perfect agreement between partitions.

4.1.10 Overall Error Rate of Classification

In classification problems, each group Q_k is associated with a priori class P_i and this association must be interpreted as if the true a priori class were P_i. Thus, for an object belonging to a given group Q_k the decision is correct if the a priori class of this object is P_i. To obtain a minimum classification error rate is necessary to find a decision rule that minimizes the probability of error. $p(P_i, Q_k)$ is the posterior probability that a pattern belongs to class P_i when is associated with the group Q_k. $p(Q_k)$ is the probability that the pattern belongs to the group Q_k and function p is the likelihood function.

The estimation of maximum a posteriori probability is the mode of a posteriori probability $p(P_i, Q_k)$ and the index of the a priori class associated to this mode is given by:

$$MAP(Q_k) = \arg\max_{1 \le i \le C} p(P_i, Q_k) \tag{4.14}$$

The Bayes decision rule that minimizes the average probability of error is select the a priori class that maximizes the probability a posteriori. The classification error rate ERC(Q_k) of the group Q_k is $1 - p(\mathbf{P}_{MAP(Q_k)}|Q_k)$ and the Overall Error Rate of Classification OERC is:

$$OERC = \sum_{k=1}^{C} p(Q_k)(1 - p(\mathbf{P}_{MAP(Q_k)}|Q_k)) \tag{4.15}$$

For a sample, $p(P_{MAP(Q_k)}|Q_k) = \max_{1 \le i \le C} \frac{n_{ik}}{a_k}$. OERC (BREIMAN et al., 1984) is designed to measure the ability of a clustering algorithm to find the a priori classes present in a dataset and is calculated as:

$$OERC = \sum_{k=1}^{C} \frac{a_k}{N}(1 - \max_{1 \le i \le C} n_{ik}/a_k) = 1 - \frac{\max_{1 \le i \le C} n_{ik}}{N} \tag{4.16}$$

The OERC takes its values from the interval $[0, 1]$ in which lower OERC values indicate better clustering results.

4.1.11 Parameter setting

Parameter selection is a complicated process in clustering algorithms that result in significant performance improvement or deterioration. In most bibliography methods, parameters adjustment is made using objects labels, which is impracticable in real clustering applications. In this work was adapted the proposal presented by Schwämmle *et al.* (SCHWÄMMLE; JENSEN, 2010) to determine the parameter values in an unsupervised way.

The choice of the parameter α for FCM algorithm was calculated according Schwämmle *et al.* using the function:

$$f(P, N) = 1 + \left(\frac{1418}{N} + 22.05\right)P^{-2} + \left(\frac{12.33}{N} + 0.243\right)P^{-0.0406\ln(N) - 0.1134} \tag{4.17}$$

where N represents the number of objects and P represents the number of variables.

For FCM-ER and AFCM-ER algorithms, T_u was founded without supervision as follows. For each dataset, the value of T_u varied between 0.1 to 100 (with step 0.1), and the threshold for T_u corresponds to the value of the fuzzifier at which the minimum distance between centroids falls under 0.1 for the first time. Based on the premise that increasing T_u increases the fuzziness of the clusters, was used binary search to find the T_u value, decreasing the search to $O(\log(N))$. An important observation is that the T_u value depends on the number of objects and variables in the dataset.

The T_v value for the FCCI algorithm varied between 0.1 to 10^8 (with step 0.1), and for every T_v value, the choice of the parameter T_u followed the same procedure used in the case of the FCM-ER and AFCM-ER algorithms. The selected parameters correspond to pair T_v, T_u with the maximum centroid distance, this means parameter selection is made unsupervised.

For FCM, FCM-ER and the proposed AFCM-ER algorithm, there is only one regularization parameter. Meanwhile, for FCCI algorithm, there is two regularization parameters T_u and T_v.

The parameter ε was set to 10^{-5}, the maximum number of iterations T was 100, and for each dataset, the number of clusters was set equal to the number of a priori classes.

The paired t-test was used to compare the proposed algorithm statistically with the other three clustering methods computed from 100 repetitions. If the p-value was below the threshold of the statistical significance level (usually 0.05), then the null hypothesis was rejected in favor of an alternative hypothesis, which typically states that the two distributions differ. Thus, if the p-value of two approaches was less than 0.05, the difference of the clustering results of the two methods was considered to be significant.

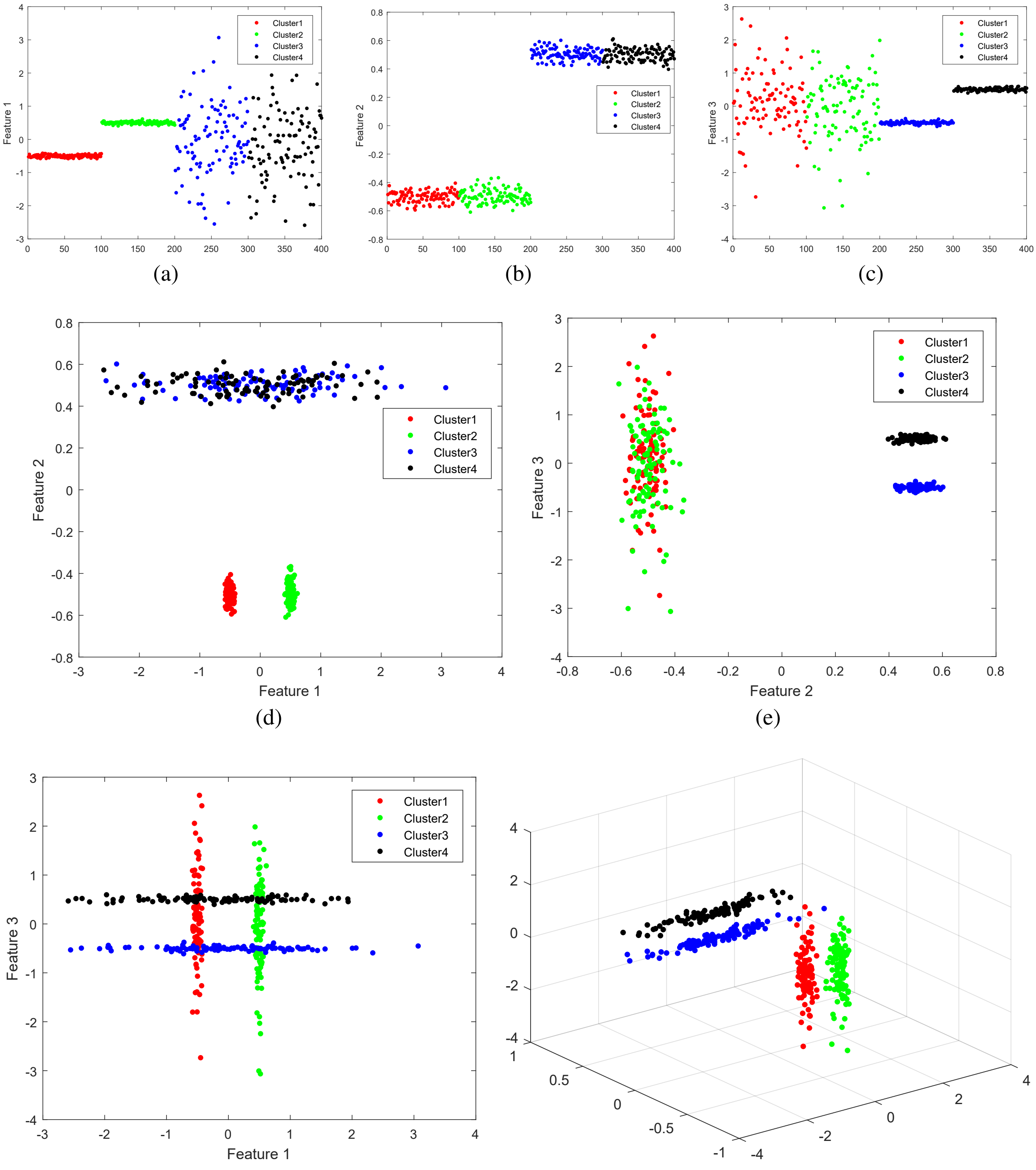
4.2 Experiments on synthetic dataset

To show robustness of the proposed method in Soft Subspace Clustering problems, a synthetic dataset (Gaussian) was created with four hundred instances in three dimensions. The synthetic dataset was divided into four groups of 100 objects each, described by three-dimensional vectors generated randomly from a normal distribution with means and standard deviations showed in Table 2.

Table 2 – Mean vector (μ_1, μ_2, μ_3) and standard deviation vector $(\sigma_1, \sigma_2, \sigma_3)$ for every cluster in synthetic Gaussian dataset.

μ	Cluster 1	Cluster 2	Cluster 3	Cluster 4
μ_1	-0.5	0.5	0.0	0.0
μ_2	-0.5	-0.5	0.5	0.5
μ_3	0.0	0.0	-0.5	0.5
σ_1	0.1	0.1	1.00	1.00
σ_2	0.1	0.1	0.1	0.1
σ_3	1.00	1.00	0.1	0.1

The artificial dataset was generated to demonstrate the capability of the proposed clustering algorithm in finding optimal variables weights for each cluster. By mapping, this dataset into all possible subspaces is observed that in almost every subspace, except Feature1-Feature3, two of the groups are well separated. Figure 5 shows that cluster 1 and 2 are better described by the variables Feature 1 and Feature 2, that is, those variables has higher weights in cluster 1 and cluster 2 definition. Cluster 3 and cluster 4 are better described by the variables Feature 2 and Feature 3, that is, these variables are relevant for the definition of clusters 3 and 4. In conclusion variable Feature 3 represents noise for clusters 1 and 2, whereas for cluster 3 and 4, the variable Feature 1 represents noise.



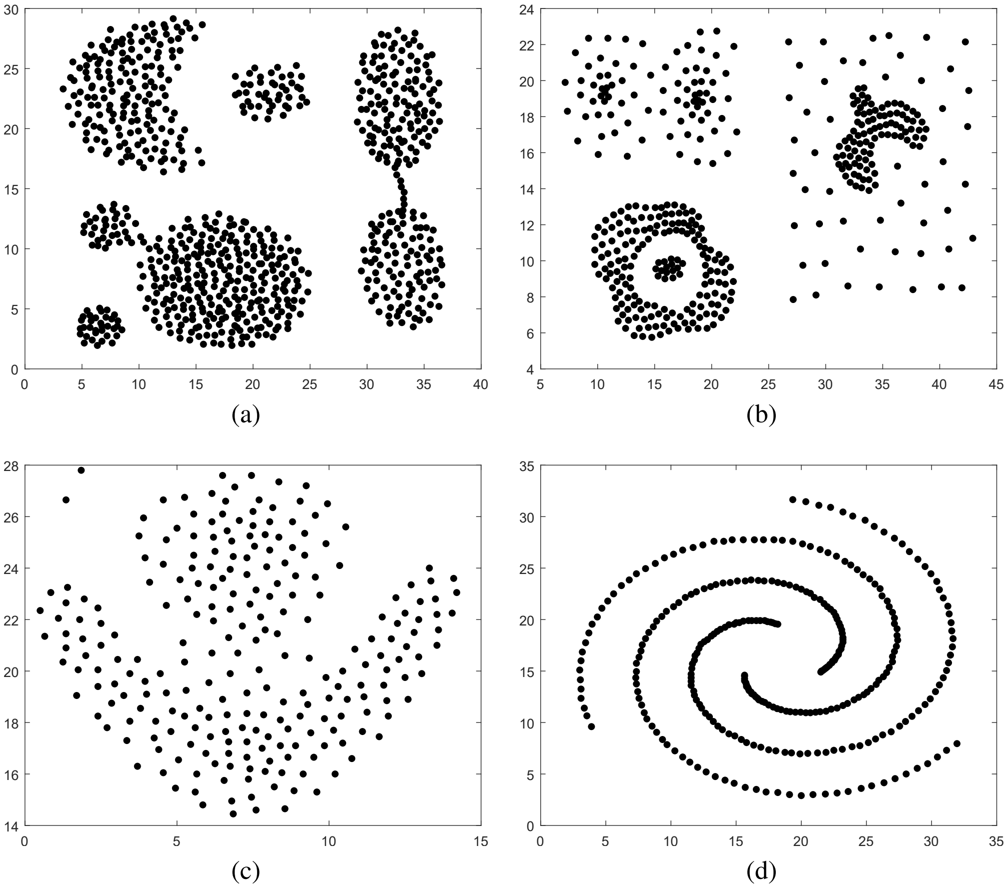
(a) (b) (c)

(d) (e)

(f) (g)

Figure 5 – The projection of the synthetic data in a single dimension (a) Feature 1, (b) Feature 2, (c) Feature 3, two dimensions (d) Feature 1 and Feature 2, (e) Feature 2 and Feature 3, (f) Feature 1 and Feature 3 and three dimensions (g) Feature 1, Feature 2 and Feature 3. In both (d) and (e) we can see that two clusters are properly separated, but the remaining two are mixed together. In (f) the four clusters are more visible, but still overlap each other are are impossible to completely separate

Also, the algorithms were tested with different shape sets to measure the performance in complex datasets. Their plots have been provided in Figure 6. Table 3 describes the datasets briefly, where *N* represents the number of objects, *P* represents the number of variables, and *C* represents the number of a priori clusters.



(a) (b)

(c) (d)

Figure 6 – Plots of synthetic datasets with different shape sets (a) Aggregation, (b) Compound, (c) Flame and (d) Spiral.

Table 3 – Summary of the synthetic datasets.

Dataset	*N*	*P*	*C*	Source
Gaussian	400	3	4	(FERREIRA; CARVALHO, 2014)
Aggregation	788	2	7	(GIONIS; MANNILA; TSAPARAS, 2007)
Compound	399	2	6	(ZAHN, 1971)
Flame	240	2	2	(FU; MEDICO, 2007)
Spiral	312	2	3	(CHANG; YEUNG, 2008)

4.2.1 Results and analysis

The α parameter value for FCM algorithm, T_u for FCM-ER and AFCM-ER algorithms and (T_u,T_v) for FCCI algorithm are shown in Table 4.

Table 4 – Parameters values for each algorithm in synthetic datasets.

Algorithm	Gaussian	Aggregation	Compound	Flame	Spiral
FCM (α)	4.03	7.16	7.61	8.22	7.87
FCM-ER (Tu)	2.20	0.40	0.40	2.00	2.00
FCCI (Tu,T_v)	(0.60,900)	(0.10,20)	(0.20,10)	(1.00,100)	(0.80,80)
AFCM-ER (Tu)	0.30	0.30	0.10	1.90	1.70

Table 5 illustrate the best clustering results (according to the objective function) for fuzzy obtained for the algorithms in synthetic datasets respectively. Also, it is shown (in parenthesis) the performance rank of each algorithm according to considered indices and datasets.

Table 5 – The best clustering results obtained on Synthetic dataset for fuzzy partitions.

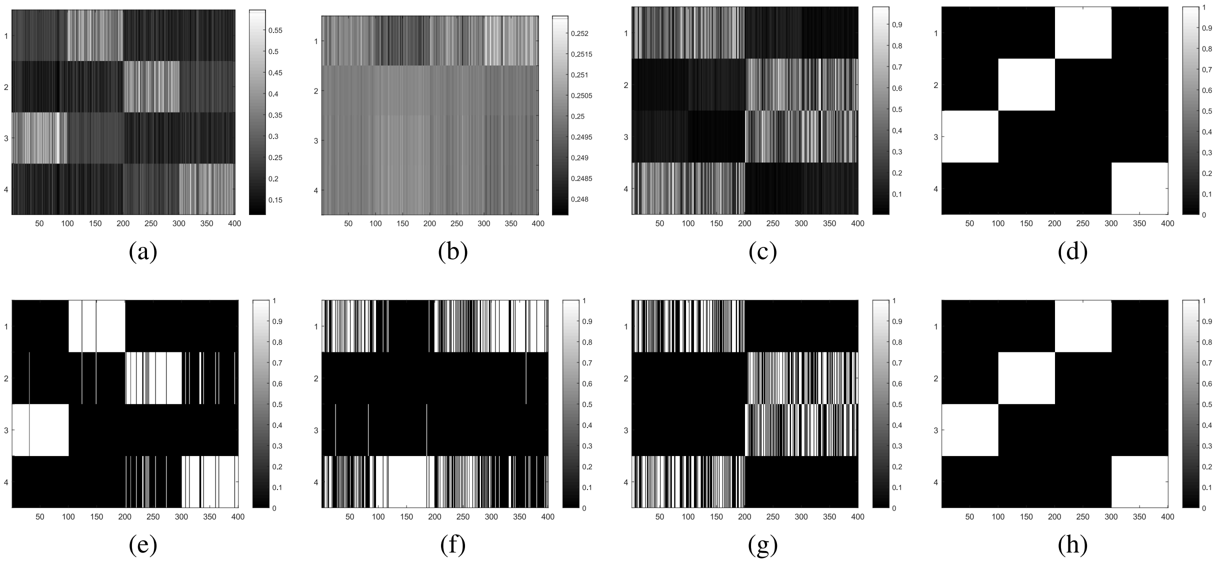
Algorithm	MPC	PE	SFS	FRHR
		Gaussian		
FCM	0.1202(3)	1.2131(3)	**0.5997**(1)	0.5218(3)
FCM-ER	$6.5 \times e^{-7}$(4)	1.3863(4)	0.0007(4)	0.2485(4)
FCCI	0.3448(2)	0.9181(2)	0.4609(3)	0.6187(2)
AFCM-ER	**1.0000**(1)	**0.0000**(1)	0.4931(2)	**1.0000**(1)
		Aggregation		
FCM	0.0077(4)	1.9249(4)	**0.6522**(1)	0.2947(4)
FCM-ER	0.5650(3)	0.6148(3)	0.6272(2)	0.9165(2)
FCCI	**0.5972**(1)	0.5961(2)	0.3188(4)	0.8206(3)
AFCM-ER	0.5850(2)	**0.5536**(1)	0.5750(3)	**0.9382**(1)
		Compound		
FCM	0.0111(4)	1.7657(4)	0.5987(2)	0.3530(4)
FCM-ER	0.5278(2)	0.6488(2)	**0.6024**(1)	**0.8562**(1)
FCCI	0.3310(3)	0.9207(3)	0.0041(4)	0.7427(3)
AFCM-ER	**0.9413**(1)	**0.0806**(1)	0.5115(3)	0.8432(2)
		Flame		
FCM	0.0142(4)	0.6860(4)	**0.6086**(1)	0.5604(2)
FCM-ER	0.0150(3)	0.6856(3)	0.1289(4)	0.5464(3)
FCCI	0.0491(2)	0.6679(2)	0.1480(3)	0.5463(4)
AFCM-ER	**0.2097**(1)	**0.5804**(1)	0.4199(2)	**0.6270**(1)
		Spiral		
FCM	0.0135(3)	1.0857(3)	**0.6628**(1)	0.3619(3)
FCM-ER	2.5×10^{-05}(4)	1.0986(4)	0.0057(4)	0.3328(4)
FCCI	**0.2275**(1)	**0.8520**(1)	0.3996(2)	**0.4637**(1)
AFCM-ER	0.1871(2)	0.8899(2)	0.3494(3)	0.4504(2)

Table 6 presents the clustering algorithms average performance ranking according to considered indices for fuzzy partitions computed from Table 5. It is also shown the performance ranking of the clustering algorithms (in parentheses) according to the average performance ranking and the Overall Average performance Ranking (OAR).

Table 6 – Average performance ranking for fuzzy partitions.

Algorithm	MPC	PE	SFS	FRHR	OAR
FCM	3.6 (4)	3.6 (4)	**1.2** (1)	3.2 (4)	3.3 (4)
FCM-ER	3.2 (3)	3.2 (3)	3.0 (3)	2.8 (3)	3.0 (3)
FCCI	1.8 (2)	2.0 (2)	3.2 (4)	2.6 (2)	2.5 (2)
AFCM-ER	**1.4** (1)	**1.2** (1)	2.6 (2)	**1.4** (1)	**1.3** (1)

It can be observed in Table 6 that the AFCM-ER algorithm presented the best average performance rank (according to the objective function) respect to the fuzzy partitions, obtaining a better quality of clustering in almost all datasets compared to the other approaches. For example, in the Gaussian dataset, for the MPC, PE and FRHR indexes in Table 5, the AFCM-ER obtain ideal clustering solutions because can adaptively identify the importance of each variable. Figure 7 displays the fuzzy (**U**) and hard (**Q**) partition matrices obtained by the algorithms on the Gaussian dataset. In the first row the fuzzy partitions are shown, in the second row the hard. Lighter colors correspond to higher object membership values in the cluster, and darker colors are associated with low membership values. In every case, abscissa axis corresponds to the objects and ordinate axis with retrieved clusters.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7 – Fuzzy and hard partition matrices obtained by (a) and (e) FCM, (b) and (f) FCM-ER, (c) and (g) FCCI and (d) and (h) AFCM-ER algorithms. The first row shows the fuzzy partition matrices and the second row the hard. Lighter colors correspond to higher object membership values in the cluster, and darker colors are associated with low membership values. In every case, abscissa axis corresponds to the objects and ordinate axis with retrieved clusters.

The proposed algorithm provides an excellent partition quality in the synthetic Gaussian dataset. The result for the fuzzy and hard partitions are exactly equal to the ground truth for this dataset. The worst performance rank (Table 6) was presented by FCM algorithm where SSC was not made.

Table 7 illustrate the best clustering results (according to the objective function) for hard partitions obtained for the algorithms in synthetic datasets respectively. Also, it is shown (in parenthesis) the performance rank of each algorithm according to considered indices and datasets.

Table 7 – The best clustering results obtained on Synthetic dataset for hard partitions.

Algorithm	ARI	FM	OERC	SCS
		Gaussian		
FCM	0.7060(2)	0.7789(2)	0.1097(2)	0.4317(2)
FCM-ER	0.1613(4)	0.4379(4)	0.4157(4)	0.0005(4)
FCCI	0.3331(3)	0.4986(3)	0.2489(3)	0.4189(3)
AFCM-ER	**1.0000**(1)	**1.0000**(1)	**0.0000**(1)	**0.4931**(1)
		Aggregation		
FCM	0.5276(3)	0.6244(3)	0.1544(3)	**0.4368**(1)
FCM-ER	0.5677(2)	0.6432(2)	0.1295(2)	0.3598(2)
FCCI	0.3251(4)	0.4597(4)	0.2168(4)	0.2663(4)
AFCM-ER	**0.7115**(1)	**0.7631**(1)	**0.0872**(1)	0.3529(3)
		Compound		
FCM	0.4706(3)	0.5891(2)	0.1857(3)	0.2918(3)
FCM-ER	0.4773(2)	0.5874(3)	0.1777(2)	0.3925(2)
FCCI	0.2350(4)	0.3974(4)	0.2611(4)	-0.0081(4)
AFCM-ER	**0.5417**(1)	**0.6386**(1)	**0.1561**(1)	**0.4842**(1)
		Flame		
FCM	0.4649(2)	0.7412(2)	0.2676(2)	**0.4574**(1)
FCM-ER	0.3676(3)	0.6941(3)	0.3163(3)	0.1293(4)
FCCI	0.0531(4)	0.5627(4)	0.4707(4)	0.1425(3)
AFCM-ER	**0.4880**(1)	**0.7525**(1)	**0.2561**(1)	0.3702(2)
		Spiral		
FCM	-0.0054(4)	0.3281(4)	**0.4457**(1)	**0.4150**(1)
FCM-ER	-0.0049(3)	0.3289(3)	0.4459(2)	0.0053(4)
FCCI	-0.0010(2)	0.3407(2)	0.4505(3)	0.3891(2)
AFCM-ER	**-0.0006**(1)	**0.3444**(1)	0.4528(4)	0.3363(3)

Table 8 presents the clustering algorithms average performance ranking according to considered indices for hard partitions computed from Table 7 respectively. It is also shown the performance ranking of the clustering algorithms (in parentheses) according to the average performance ranking and the OAR.

Table 8 – Average performance ranking for hard partitions.

Algorithm	ARI	FM	OERC	SCS	OAR
FCM	2.8 (3)	2.6 (2)	2.2 (2)	**1.6** (1)	2.0 (2)
FCM-ER	2.8 (3)	3.0 (3)	2.6 (3)	3.2 (4)	3.3 (3)
FCCI	3.4 (4)	3.4 (4)	3.6 (4)	3.2 (4)	4.0 (4)
AFCM-ER	**1.0** (1)	**1.0** (1)	**1.6** (1)	2.0 (2)	**1.3** (1)

For hard partitions, the AFCM-ER algorithm also performed better than FCM, FCM-ER and FCCI algorithms for most of the datasets (according to the objective function). According ARI and FM the proposed method obtains rank (1) for all datasets and for Gaussian and Compound dataset shows excellent results in all four indices. In conclusion, obtained results showed that AFCM-ER algorithm is good for soft subspace clustering problems, even for different shape sets the performance of proposed algorithm stays constant. FCCI shows the worst performance for hard partition according OAR. This is because the AFCM-ER algorithm emphasizes more on the importance of correct attributes for each cluster, obtaining higher clustering accuracy compared with the other approaches. In Gaussian dataset, AFCM-ER algorithm emphasize on the importance of the attributes 1 and 2 for clusters 1 and 2, and 2 and 3 for clusters 3 and 4, obtaining higher performance compared with FCCI algorithm. FCM and FCM-ER assume the same importance of the variables in the clustering task, resulting in the worst performance compared with the proposed algorithm. Table 9 shows the matrix **V** of relevance weights of the variables on fuzzy clusters obtained by FCCI and AFCM-ER algorithms. These matrices correspond to the best solution (according to the objective function) in 100 algorithms executions in the Gaussian dataset.

Table 9 – Attribute weight assignment on Gaussian dataset.

Algorithm	Attribute 1	Attribute 2	Attribute 3
FCCI			
Cluster 1	**0.3396**	**0.3385**	0.3219
Cluster 2	**0.3433**	**0.3405**	0.3162
Cluster 3	0.3190	**0.3416**	**0.3394**
Cluster 4	0.3232	**0.3382**	**0.3386**
AFCM-ER			
Cluster 1	**14.1086**	**3.5036**	0.0202
Cluster 2	**11.2558**	**4.5525**	0.0195
Cluster 3	0.0213	**4.6917**	**10.0054**
Cluster 4	0.0184	**4.9081**	**11.0573**

Table 10 summarizes the results produced by the four clustering algorithms, expressed in terms of the means, standard deviations and p-value (comparing AFCM-ER with the others) of MPC, PE, SFS and FRHR for fuzzy partitions obtained by executing each algorithm 100 times. The first value indicate the mean and the second value is the standard deviation of the results.

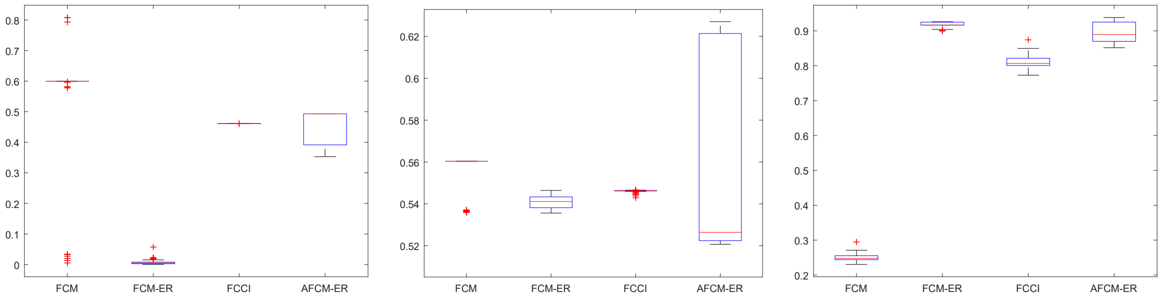
Table 10 – Clustering performance on the synthetic datasets for fuzzy partitions.

Algorithm	MPC	PE	SFS	FRHR
		Gaussian		
FCM	0.1165 ± 0.0111	1.2176 ± 0.0143	**0.5703 ± 0.1436**	0.5180 ± 0.0149
(p-value)	1.3×10^{-82}	1.3×10^{-93}	7.9×10^{-12}	8.4×10^{-106}
FCM-ER	0.0001 ± 0.0001	1.3862 ± 0.0002	0.0066 ± 0.0070	0.2513 ± 0.0024
(p-value)	4.2×10^{-89}	2.3×10^{-100}	5.4×10^{-95}	3.9×10^{-130}
FCCI	0.3448 ± 0.0000	0.9181 ± 0.0000	0.4609 ± 0.0000	0.6187 ± 0.0000
(p-value)	1.2×10^{-68}	7.5×10^{-81}	0.0647	7.4×10^{-100}
AFCM-ER	**0.8903 ± 0.1185**	**0.1197 ± 0.1293**	0.4513 ± 0.0513	**0.9719 ± 0.0365**
		Aggregation		
FCM	0.0015 ± 0.0009	1.9416 ± 0.0026	0.3430 ± 0.1110	0.2494 ± 0.0093
(p-value)	8.8×10^{-132}	3.2×10^{-165}	1.0×10^{-20}	4.5×10^{-130}
FCM-ER	0.5800 ± 0.0270	0.5975 ± 0.0216	**0.6260 ± 0.0321**	**0.9165 ± 0.0074**
(p-value)	6.6×10^{-09}	1.7×10^{-26}	1.9×10^{-29}	3.3×10^{-11}
FCCI	0.5875 ± 0.0546	0.5925 ± 0.0782	0.2750 ± 0.0811	0.8120 ± 0.0177
(p-value)	0.0037	1.9×10^{-08}	6.5×10^{-40}	8.9×10^{-39}
AFCM-ER	**0.6053 ± 0.0295**	**0.5422 ± 0.0312**	0.5004 ± 0.0667	0.8921 ± 0.0313
		Compound		
FCM	0.0058 ± 0.0026	1.7780 ± 0.0061	0.4981 ± 0.1027	0.3129 ± 0.0199
(p-value)	3.2×10^{-185}	8.1×-197	2.1×10^{-04}	5.2×10^{-127}
FCM-ER	0.5335 ± 0.0260	0.6200 ± 0.0346	**0.5601 ± 0.0674**	**0.8631 ± 0.0082**
(p-value)	4.6×10^{-114}	4.0×10^{-116}	0.0231	1.2×10^{-33}
FCCI	0.3479 ± 0.0322	0.9260 ± 0.0371	0.1126 ± 0.1064	0.7470 ± 0.0131
(p-value)	1.9×10^{-122}	6.0×10^{-131}	5.4×10^{-61}	8.2×10^{-57}
AFCM-ER	**0.9258 ± 0.0132**	**0.1029 ± 0.0181**	0.5413 ± 0.0446	0.8280 ± 0.0194
		Flame		
FCM	0.0127 ± 0.0044	0.6868 ± 0.0022	**0.5538 ± 0.1567**	**0.5577 ± 0.0075**
(p-value)	2.1×10^{-63}	5.9×10^{-62}	1.3×10^{-23}	0.6050
FCM-ER	0.0043 ± 0.0039	0.6910 ± 0.0020	0.0622 ± 0.0342	0.5408 ± 0.0029
(p-value)	6.5×10^{-66}	2.6×10^{-64}	1.8×10^{-70}	6.9×10^{-05}
FCCI	0.0485 ± 0.0012	0.6682 ± 0.0006	0.1442 ± 0.0053	0.5461 ± 0.0006
(p-value)	$2.2 \times 10-51$	2.5×10^{-50}	3.0×10^{-64}	0.0034
AFCM-ER	**0.1464 ± 0.0328**	**0.6156 ± 0.0181**	0.3382 ± 0.0469	**0.5602 ± 0.0468**
		Spiral		
FCM	0.0096 ± 0.0054	1.0894 ± 0.0051	**0.5428 ± 0.1746**	0.3550 ± 0.0090
(p-value)	7.2×10^{-136}	1.6×10^{-150}	9.8×10^{-18}	3.8×10^{-103}
FCM-ER	0.0004 ± 0.0004	1.0982 ± 0.0004	0.0225 ± 0.0089	0.3365 ± 0.0022
(p-value)	5.3×10^{-146}	$1.1 \times 10-165$	3.8×10^{-110}	1.7×10^{-147}
FCCI	**0.2241 ± 0.0110**	**0.8557 ± 0.0091**	0.3730 ± 0.0342	**0.4610 ± 0.0044**
(p-value)	5.6×10^{-48}	1.6×10^{-54}	7.6×10^{-05}	1.6×10^{-34}
AFCM-ER	0.1876 ± 0.0065	0.8898 ± 0.0046	0.3546 ± 0.0253	0.4503 ± 0.0031

The clustering results regarding the means and standard deviations (Table 10) of the MPC, PE, indicate robustness in the performance of the AFCM-ER algorithm in fuzzy partitions obtaining significantly better results over other compared approaches. For example, the best solution of the proposed algorithm in Spiral dataset according to the objective function acquire the second rank, but respect to the mean and standard deviation achieve the best performance over the others methods. These results suggest that the proposed algorithm has a good performance in datasets with complex structures and detecting relevant and irrelevant features for the determination of the clusters.

For SFS index the proposed algorithm obtains the second rank in the Table 6 and maintains this same behavior in Table 10. For Gaussian dataset case, the rank for proposed algorithm was (1) for SFS index in Table 6 and the third according to the mean. This result was obtained because the proposed algorithm has a high degree of dispersion (Figure 8 (a)) so is less robust for SFS index in the Gaussian dataset.

Paying attention to the obtained p-values, in Flame dataset for the FRHR index in FCM algorithm, the results are not significantly different respect to proposed algorithm. For Aggregation dataset, the proposed algorithm obtains the rank (1) in Table 6, but respect to the mean and standard deviation get the rank (2). These results are obtained because the AFCM-ER algorithm has a high degree of dispersion concerning the other approaches. The 100 iterations results for FRHR for Flame and Aggregation datasets are shown in Figure 8 (b) and (c). However, in real applications ground truth labels are not available, and from 100 algorithms executions, one response has to be given. Because the problem nature is unsupervised, the best option is lower objective function value result.



(a) Gaussian (b) Flame (c) Aggregation

Figure 8 – Box plots for the obtained values of (a) SFS in Gaussian dataset, (b) FRHR in Flame dataset and (c) FRHR in Aggregation dataset.

Table 11 summarizes the results produced by the four clustering algorithms, expressed in terms of the means, standard deviations and p-value (comparing AFCM-ER with the others) of ARI, FM, OERC and SCS indices for hard partitions obtained by executing each algorithm 100 times. The first value indicate the mean and the second value is the standard deviation of the results.

Table 11 – Clustering performance on the synthetic datasets for hard partitions.

Algorithm	ARI	FM	OERC	SCS
		Gaussian		
FCM	0.7048 ± 0.0773	0.7789 ± 0.0559	0.1112 ± 0.0322	**0.4024 ± 0.1013**
(p-value)	1.1×10^{-12}	3.4×10^{-14}	2.0×10^{-09}	0.2111
FCM-ER	0.1667 ± 0.0141	0.4366 ± 0.0082	0.4027 ± 0.0119	0.0035 ± 0.0028
(p-value)	2.0×10^{-64}	2.7×10^{-61}	8.3×10^{-72}	2.3×10^{-72}
FCCI	0.3331 ± 0.0000	0.4986 ± 0.0000	0.2489 ± 0.0000	**0.4189 ± 0.0000**
(p-value)	1.1×10^{-53}	8.3×10^{-56}	1.9×10^{-48}	0.9008
AFCM-ER	**0.8464 ± 0.1619**	**0.8895 ± 0.1167**	**0.0634 ± 0.0670**	**0.4199 ± 0.0827**
		Aggregation		
FCM	**0.5622 ± 0.0767**	**0.6619 ± 0.0601**	0.1550 ± 0.0279	0.1736 ± 0.0669
(p-value)	0.0728	0.9102	7.2×10^{-10}	2.3×10^{-34}
FCM-ER	**0.6033 ± 0.0432**	**0.6740 ± 0.0368**	**0.1199 ± 0.0121**	**0.3778 ± 0.0321**
(p-value)	0.1533	0.1633	0.1348	0.5228
FCCI	0.3380 ± 0.0480	0.4697 ± 0.0381	0.2126 ± 0.0176	0.1673 ± 0.0735
(p-value)	1.8×10^{-38}	7.2×10^{-36}	1.1×10^{-43}	3.3×10^{-37}
AFCM-ER	**0.5868 ± 0.1052**	**0.6607 ± 0.0864**	**0.1251 ± 0.0318**	**0.3723 ± 0.0769**
		Compound		
FCM	**0.5724 ± 0.0902**	**0.6792 ± 0.0741**	**0.1609 ± 0.0275**	0.2784 ± 0.0726
(p-value)	1.1×10^{-10}	8.6×10^{-14}	5.7×10^{-04}	1.6×10^{-49}
FCM-ER	0.5127 ± 0.0509	0.6167 ± 0.0417	0.1668 ± 0.0164	0.3958 ± 0.0336
(p-value)	0.0061	0.0056	0.0083	8.6×10^{-38}
FCCI	0.2764 ± 0.0307	0.4334 ± 0.0287	0.2498 ± 0.0085	0.0072 ± 0.0311
(p-value)	7.6×10^{-53}	7.5×10^{-49}	2.9×10^{-59}	6.7×10^{-101}
AFCM-ER	0.4923 ± 0.0651	0.5994 ± 0.0546	0.1728 ± 0.0197	**0.5081 ± 0.0410**
		Flame		
FCM	**0.4415 ± 0.0921**	**0.7299 ± 0.0446**	**0.2793 ± 0.0461**	**0.4123 ± 0.1282**
(p-value)	9.8×10^{-14}	1.4×10^{-13}	1.0×10^{-13}	2.3×10^{-13}
FCM-ER	0.3588 ± 0.0654	0.6899 ± 0.0317	0.3207 ± 0.0327	0.0633 ± 0.0344
(p-value)	1.9×10^{-07}	3.1×10^{-07}	2.1×10^{-07}	6.1×10^{-67}
FCCI	0.0613 ± 0.0043	0.5646 ± 0.0010	0.4669 ± 0.0020	0.1348 ± 0.0055
(p-value)	1.0×10^{-11}	1.6×10^{-08}	2.9×10^{-11}	8.9×10^{-62}
AFCM-ER	0.2292 ± 0.2184	0.6289 ± 0.1045	0.3853 ± 0.1091	0.2988 ± 0.0419
		Spiral		
FCM	−0.0050 ± 0.0010	0.3351 ± 0.0116	**0.4505 ± 0.0082**	0.3263 ± 0.1155
(p-value)	1.8×10^{-29}	1.6×10^{-10}	0.0375	0.9663
FCM-ER	−0.0050 ± 0.0007	**0.3499 ± 0.0197**	0.4621 ± 0.0154	0.0189 ± 0.0074
(p-value)	1.4×10^{-30}	0.0055	5.7×10^{-09}	3.9×10^{-92}
FCCI	**0.0022 ± 0.0042**	0.3477 ± 0.0079	0.4528 ± 0.0029	**0.3574 ± 0.0377**
(p-value)	4.4×10^{-05}	0.0015	0.2809	3.0×10^{-07}
AFCM-ER	-3.8×10^{-05} ± 0.0029	0.3443 ± 0.0055	0.4523 ± 0.0024	0.3268 ± 0.0370

The clustering results in terms of the means and standard deviations of the ARI, FM, OERC and SCS values, indicate robustness in the performance of the AFCM-ER algorithm for hard partitions in Gaussian dataset. For datasets with different shape sets the performance of proposed method is less robust than FCM algorithm. In the Aggregation dataset, for FCM and FCM-ER algorithms, the results are not significant different respect to the proposed algorithm because the algorithm, despite the difference in the constraints, is a FCM algorithm variation.

In general, the proposed algorithm shows the best performance in comparison with the other approaches in the defined synthetic datasets for almost all indexes. Selecting the best solution according to the objective function, showed a high performance. The results obtained according to the mean and the variance for fuzzy partitions, the proposed method achieves the best performance, although for the hard partitions accuracy decreased.

4.3 Experiments on real datasets

To show the effectiveness of the proposed method, ten real datasets available at the *UCI* machine learning repository (ASUNCION; NEWMAN, 1994) were chosen for comparative analysis: Abalone, Breast tissue, Haberman, Iris plant, Led7, Scale, Sonar, Thyroid gland, UKM, and Vehicle. Table 12 describes briefly the datasets.

Table 12 – Summary of the real datasets available at the *UCI* machine learning repository.

Dataset	N	P	C	Dataset	N	P	C
Abalone	4177	8	3	Scale	625	4	3
Breast tissue	106	9	6	Sonar	208	60	2
Haberman	306	3	2	Thyroid gland	215	5	3
Iris plant	150	4	3	UKM	403	5	4
Led7	3200	7	10	Vehicle	846	18	4

Abalone: The objective is to predict the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope. This process is a tedious and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

Breast tissue: Impedance measurements of freshly excised breast tissue were made at the following frequencies: 15.625, 31.25, 62.5, 125, 250, 500, 1000 KHz. These measurements plotted in the (real, imaginary) plane constitute the impedance spectrum from where the breast tissue features are computed. The dataset can be used for predicting the classification of either the original six classes or of 4 classes by merging the fibro-adenoma, mastopathy and glandular classes whose discrimination is not necessary (they cannot be accurately discriminated anyway).

Haberman: This dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The task is to determine if the patient survived five years or longer (positive) or if the patient died within five years (negative).

Iris plant: This dataset is perhaps the best-known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is frequently referenced to this day. The dataset contains three classes of 50 instances each, where each category refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The objective is to predict the class of iris plant.

Led7: This simple dataset domain contains 7 Boolean attributes and ten concepts, the set of decimal digits. The LED displays contain seven light-emitting diodes, hence the reason for seven characteristics. The task is to determine which digit is shown in the display. The problem would be easy if not for the introduction of noise. In this case, each attribute value has the 10% probability of having its value inverted. This dataset is a sample of 500 instances obtained from the original data generator.

Scale: This dataset was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the proper distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.

Sonar: The file "sonar.mines"contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. The file "sonar.rocks"contains 97 patterns obtained from rocks under similar conditions. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The dataset contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. Each pattern is a set of 60 numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period. The integration aperture for higher frequencies occurs later in the time since these frequencies are transmitted later during the chirp. The label associated with each record contains the letter "R"if the object is a rock and "M"if it is a mine (metal cylinder). The numbers in the labels are in increasing order of aspect angle, but they do not encode the angle directly.

Thyroid gland: The original thyroid disease (ann-thyroid) dataset from UCI machine learning repository is a classification dataset, which is suited for training ANNs. It has 3772 training instances and 3428 testing instances. It has 15 categorical and six real attributes. The problem is to determine whether a patient referred to the clinic is hypothyroid. Therefore three classes are built: normal (not hypothyroid), hyperfunction and subnormal functioning. For outlier detection, 3772 training instances are used, with only six real attributes. The hyperfunction class is treated as outlier class, and other two classes are inliers because hyperfunction is a clear minority class.

UKM: It is the real dataset about the students' knowledge status about the subject of Electrical DC Machines. The dataset had been obtained from Ph.D. Thesis. The users' knowledge class were classified by the authors using intuitive knowledge classifier (a hybrid ML technique of *K-NN* and meta-heuristic exploring methods), k-nearest neighbor algorithm

Vehicle: The purpose is to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles.

4.3.1 Results and analysis

The choice of the parameter value for the four algorithms followed the same procedure used in subsection 4.1.11. Table 13 shows the parameters values for each real dataset.

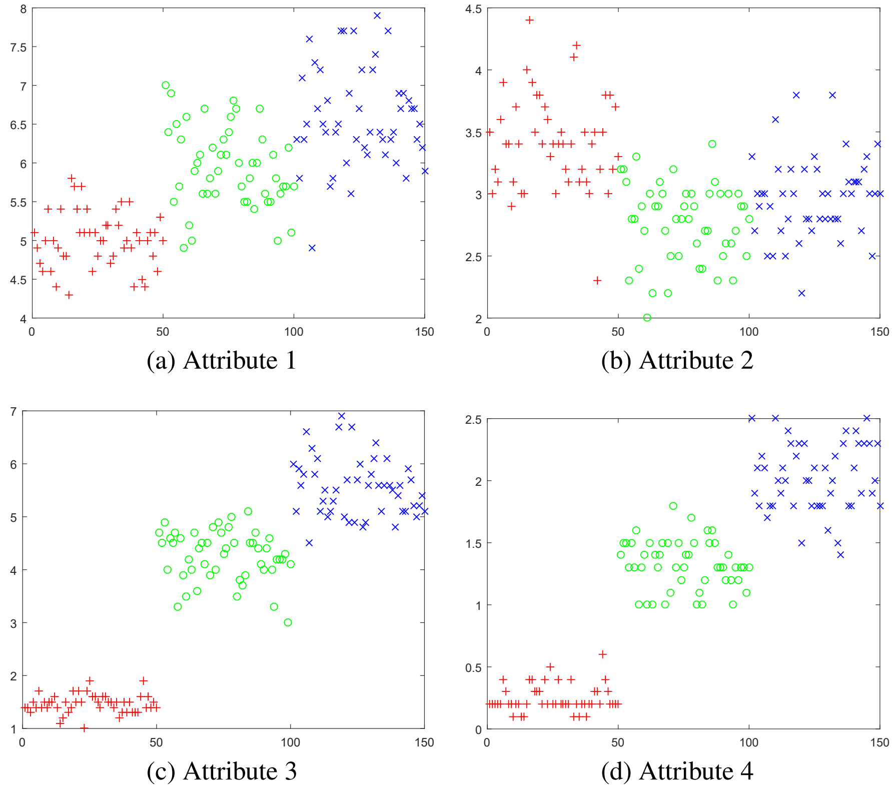
Table 13 – Parameters values in real datasets.

Algorithm	Abalone	Breast tissue	Haberman	Iris plant	Led7
FCM (α)	1.45	1.62	4.16	3.18	1.56
FCM-ER (Tu)	13.10	1.20	2.20	2.60	2.00
FCCI (Tu,T_v)	(1.60,500)	(0.10,10^8)	(0.60,2.5)	(1.00,4.5)	(0.20,900)
AFCM-ER (Tu)	12.00	0.60	9.30	2.30	0.10

Algorithm	Scale	Sonar	Thyroid gland	UKM	Vehicle
FCM (m)	2.68	1.09	2.32	2.18	1.16
FCM-ER (Tu)	2.00	24.20	5.30	2.00	5.70
FCCI (Tu,T_v)	(0.40,1.5)	(0.40,1.5)	(1.00,4.5)	(0.40,80)	(0.20,600)
AFCM-ER (Tu)	1.90	50.00	2.60	2.30	15.50

The fuzzy clustering algorithms were applied to the datasets to obtain a C-cluster partitions. For each dataset, the algorithms were run 100 times, and the best results were selected according to the minimum value of their objective function. Clustering results are also calculated, expressed in terms of the means, standard deviations and the p-value of the indices values.

To have an intuitive understanding of the physical properties of the attribute weight assignment was plotted the distribution of four attributes of the Iris plant dataset shown in Figure 9. It can be seen that attributes 3 and 4 are more compact in each cluster. For this reason, they should be more important and contribute more in clustering, so that higher weights should be assigned to these two attributes.



(a) Attribute 1

(b) Attribute 2

(c) Attribute 3

(d) Attribute 4

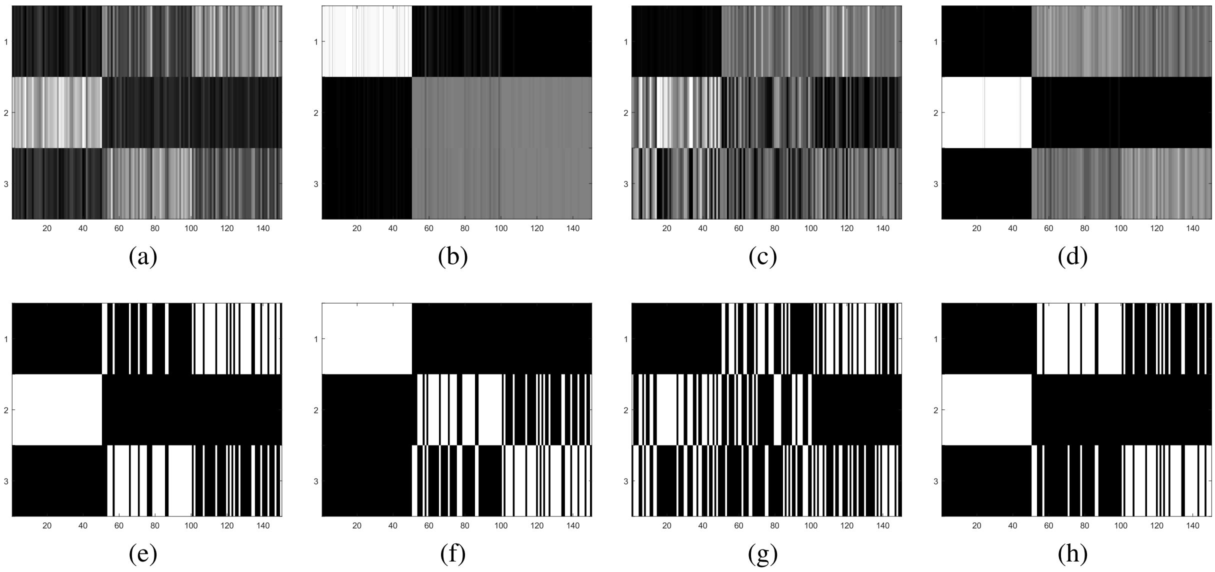
Figure 9 – Distribution of different attributes of Iris plant dataset.

Table 14 shows the matrix **V** of relevance weights of the variables on fuzzy clusters obtained by FCCI and AFCM-ER algorithms. These matrices correspond to the best result according to the clustering adequacy criterion in 100 algorithms executions. FCM and FCM-ER algorithms assume the same importance of the variables for the clustering task. Compared with the FCCI method, the proposed method emphasizes the importance of these two attributes, to achieve higher clustering accuracy. This further verifies the physical meaning of the maximum entropy method for attribute weight assignment which is consistent with the available data.

Table 14 – Attribute weight assignment on Iris plant dataset.

Algorithm	Attribute 1	Attribute 2	Attribute 3	Attribute 4
FCCI				
Cluster 1	0.0120	0.0059	**0.6500**	**0.3321**
Cluster 2	**0.9597**	0.0000	0.0183	**0.0220**
Cluster 3	0.0000	**0.9998**	**0.0001**	**0.0001**
AFCM-ER				
Cluster 1	0.6170	0.6890	**1.8150**	**1.2959**
Cluster 2	0.4632	0.1053	**5.8954**	**3.4787**
Cluster 3	0.6520	0.6795	**1.7794**	**1.2684**

Figure 10 shows the fuzzy (**U**) and hard (**Q**) partition matrices obtained by the algorithms on Iris plant dataset. Abscissa axis corresponds to the objects and ordinate axis with retrieved clusters. The first row in Figure 10 corresponds to the fuzzy partition matrices and the second row the hard. It can be observed that the AFCM-ER algorithm provides a better quality fuzzy and hard partitions in comparison with results obtained by the others approaches. This result is achieved because AFCM-ER algorithm emphasizes more on the importance of attributes 3 and 4.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 10 – Fuzzy and hard partition matrices in real datasets obtained by (a) and (e) FCM, (b) and (f) FCM-ER, (c) and (g) FCCI and (d) and (h) AFCM-ER algorithms. The first row corresponds to the fuzzy partition matrices and the second row the hard. Lighter colors correspond to higher object membership values in the cluster, and darker colors are associated with low membership values. In every case, abscissa axis corresponds to the objects and ordinate axis with retrieved clusters.

Table 15 shows the best clustering results (according to the objective function) for fuzzy partitions. Also, it is shown (in parenthesis) the performance rank of each algorithm according to the indices and considered datasets.

Table 16 presents the average performance ranking of the clustering algorithms according to the indices considered computed from Table 15. It is also shown the performance ranking of the clustering algorithms (in parentheses) according to the average performance ranking and the OAR.

It can be observed that FCCI and AFCM-ER algorithms present the best average performance rank (according to the objective function) respect to fuzzy partitions, obtaining a good quality of clustering in real datasets. Moreover FCM algorithm achieved the third best average rankings. FCM-ER algorithm presented the worst performance.

Table 15 – Cluster validity measures on real datasets for fuzzy partitions.

Algorithms	MPC	PE	SFS	FRHR	Algorithms	MPC	PE	SFS	FRHR
		Abalone					Scale		
FCM	**0.8131**(1)	**0.2151**(1)	**0.6155**(1)	**0.6026**(1)	FCM	0.0001(4)	1.0986(4)	**0.2484**(1)	0.4303(4)
FCM-ER	0.0081(4)	1.0905(4)	0.1493(4)	0.3657(4)	FCM-ER	0.0001(4)	1.0985(3)	0.0063(4)	0.4312(3)
FCCI	0.0880(3)	1.0067(3)	0.3463(3)	0.4306(3)	FCCI	**0.5133**(1)	**0.4996**(1)	0.0986(3)	**0.5389**(1)
AFCM-ER	0.0968(2)	0.9963(2)	0.4664(2)	0.4446(2)	AFCM-ER	0.3364(2)	0.6951(2)	0.1692(2)	0.5272(2)
		Breast tissue					Sonar		
FCM	0.6659(3)	0.5560(4)	**0.5632**(1)	0.7360(4)	FCM	**0.8801**(1)	**0.1010**(1)	**0.2811**(1)	**0.5036**(1)
FCM-ER	0.6616(4)	0.4451(3)	0.4750(4)	0.7415(3)	FCM-ER	0.0037(3)	0.6913(3)	0.0271(3)	0.5001(2)
FCCI	0.7866(2)	0.2892(2)	0.5153(3)	**0.7551**(1)	FCCI	0.3906(2)	0.4427(2)	0.0584(2)	0.4998(3)
AFCM-ER	**0.9376**(1)	**0.0925**(1)	0.5239(2)	0.7423(2)	AFCM-ER	0.0025(4)	0.6919(4)	0.0086(4)	0.4996(4)
		Haberman					Thyroid gland		
FCM	0.0689(2)	0.6578(2)	**0.5980**(1)	0.5752(4)	FCM	0.2364(2)	0.8347(2)	**0.4691**(1)	0.5787(3)
FCM-ER	0.0064(4)	0.6900(4)	0.0710(3)	0.5990(2)	FCM-ER	0.0033(4)	1.0953(4)	0.0767(4)	0.5549(4)
FCCI	**0.4146**(1)	**0.4335**(1)	0.1841(2)	0.5760(3)	FCCI	0.1036(3)	0.9780(3)	0.2513(3)	0.6659(2)
AFCM-ER	0.0096(3)	0.6879(3)	0.0401(4)	**0.6086**(1)	AFCM-ER	**0.2423**(1)	**0.7973**(1)	0.3206(2)	**0.6953**(1)
		Iris plant					UKM		
FCM	0.2437(3)	0.8623(3)	0.7288(3)	0.6157(3)	FCM	0.0008(4)	1.3851(4)	0.0450(4)	0.2860(4)
FCM-ER	0.4433(2)	0.5744(2)	0.7609(2)	0.7516(2)	FCM-ER	0.1612(3)	1.1455(3)	0.1918(2)	0.4943(3)
FCCI	0.2020(4)	0.9029(4)	0.6895(4)	0.5725(4)	FCCI	**0.2566**(1)	0.9822(2)	**0.1919**(1)	**0.5815**(1)
AFCM-ER	**0.4905**(1)	**0.4836**(1)	**0.7640**(1)	**0.7718**(1)	AFCM-ER	0.2436(2)	**0.9611**(1)	0.1714(3)	0.5580(2)
		Led7					Vehicle		
FCM	0.6165(3)	0.8353(3)	**0.8049**(1)	0.8180(3)	FCM	**0.8807**(1)	**0.1549**(1)	0.4131(3)	**0.6478**(1)
FCM-ER	0.4185(4)	1.1055(4)	0.4855(3)	0.7814(4)	FCM-ER	0.4654(3)	0.6896(3)	**0.5250**(1)	0.5660(3)
FCCI	0.7687(2)	0.4302(2)	0.5375(2)	**0.8788**(1)	FCCI	0.6624(2)	0.4329(2)	0.4289(2)	0.6206(2)
AFCM-ER	**0.9990**(1)	**0.0030**(1)	0.3915(4)	0.8569(2)	AFCM-ER	0.1434(4)	1.1347(4)	0.0342(4)	0.4567(4)

Table 16 – Average performance ranking for fuzzy partitions.

Algorithm	MPC	PE	SFS	FRHR	OAR
FCM	2.4 (3)	2.5 (3)	**1.7** (1)	2.8 (3)	2.5 (3)
FCM-ER	3.5 (4)	3.3 (4)	3.0 (4)	3.0 (4)	4.0 (4)
FCCI	**2.1** (1)	2.2 (2)	2.5 (2)	**2.1** (1)	**1.5** (1)
AFCM-ER	**2.1** (1)	**2.0** (1)	2.8 (3)	**2.1** (1)	**1.5** (1)

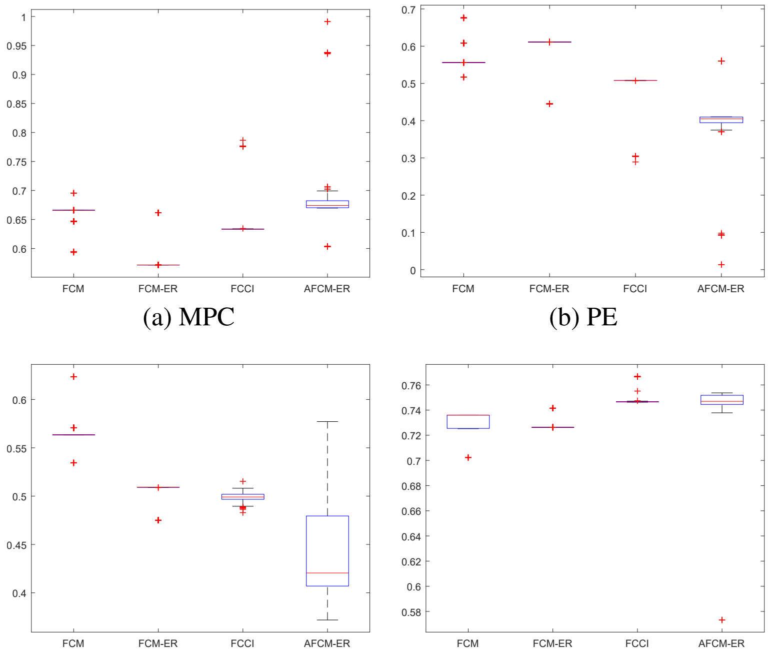
The experimental results are also reported regarding the mean and standard deviation of MPC, PE, SFS, and FRHR indexes for 100 runs of the corresponding algorithm for fuzzy partitions. In particular, a paired two-tailed t-test has been adopted as the statistical test to compare the performance of the proposed AFCM-ER algorithm and the other four methods on real datasets. The p-value of t-test represents the probability that two sets of compared samples come from distributions with equal means. The smaller the p-value, the more significant the difference between the two average values is. Table 17 reports the means, standard deviations and p-values of the index values obtained by the clustering algorithms on considered datasets for internal indices (YEUNG; HAYNOR; RUZZO, 2001).

The results expressed regarding the means and standard deviations for fuzzy partitions, show that the proposed algorithm usually demonstrates much better performance than the other approaches in almost all considered indices proving more robustness. Despite the good perfor-

Table 17 – Clustering performance on real datasets for fuzzy partitions. The first value corresponds to the mean and the second value is the standard deviation of 100 results.

Algorithm	MPC	PE	SFS	FRHR	Algorithm	MPC	PE	SFS	FRHR
Abalone					Scale				
FCM	**0.8131 ± 0.0000**	**0.2151 ± 0.0000**	**0.6155 ± 0.0000**	**0.6026 ± 0.0000**	FCM	0.0001 ± 0.0001	1.0985 ± 0.0001	**0.2565 ± 0.2040**	0.4304 ± 0.0003
(p-value)	4.6×10^{-279}	8.2×10^{-280}	2.7×10^{-70}	3.3×10^{-173}	(p-value)	4.6×10^{-106}	1.1×10^{-119}	2.3×10^{-04}	3.1×10^{-122}
FCM-ER	0.0027 ± 0.0020	1.0959 ± 0.0020	0.0813 ± 0.0331	0.3509 ± 0.0065	FCM-ER	0.0005 ± 0.0002	1.0981 ± 0.0002	0.0133 ± 0.0032	0.4340 ± 0.0014
(p-value)	8.3×10^{-157}	3.8×10^{-157}	5.7×10^{-75}	7.8×10^{-103}	(p-value)	4.7×10^{-106}	1.1×10^{-119}	8.3×10^{-78}	1.2×10^{-119}
FCCI	0.0876 ± 0.0004	1.0068 ± 0.0002	0.3709 ± 0.0346	0.4324 ± 0.0019	FCCI	**0.4639 ± 0.0175**	**0.5861 ± 0.0306**	−0.0092 ± 0.0381	0.5107 ± 0.0103
(p-value)	2.8×10^{-87}	2.0×10^{-92}	7.6×10^{-15}	1.4×10^{-47}	(p-value)	1.8×10^{-56}	5.6×10^{-45}	2.5×10^{-61}	9.0×10^{-26}
AFCM-ER	0.0967 ± 0.0011	0.9966 ± 0.0012	0.4181 ± 0.0412	0.4415 ± 0.0030	AFCM-ER	0.3409 ± 0.0304	0.6921 ± 0.0264	0.1783 ± 0.0289	**0.5285 ± 0.0060**
Breast tissue					Sonar				
FCM	0.6599 ± 0.0182	0.5704 ± 0.0332	**0.5657 ± 0.0139**	0.7321 ± 0.0076	FCM	**0.8738 ± 0.0117**	**0.1056 ± 0.0085**	**0.2796 ± 0.0026**	**0.5031 ± 0.0010**
(p-value)	2.7×10^{-05}	7.3×10^{-26}	3.9×10^{-46}	9.2×10^{-06}	(p-value)	1.7×10^{-187}	5.6×10^{-184}	1.1×10^{-201}	1.0×10^{-59}
FCM-ER	0.5841 ± 0.0314	0.5879 ± 0.0579	0.5044 ± 0.0119	0.7285 ± 0.0053	FCM-ER	0.0013 ± 0.0012	0.6925 ± 0.0006	0.0141 ± 0.0077	0.5000 ± 0.0001
(p-value)	4.3×10^{-20}	2.2×10^{-27}	5.7×10^{-26}	1.2×10^{-08}	(p-value)	3.9×10^{-17}	6.6×10^{-19}	1.8×10^{-10}	7.0×10^{-81}
FCCI	0.6534 ± 0.0501	0.4795 ± 0.0715	0.4988 ± 0.0052	**0.7492 ± 0.0068**	FCCI	0.3826 ± 0.0442	0.4545 ± 0.0293	0.0289 ± 0.0180	0.5012 ± 0.0040
(p-value)	2.3×10^{-05}	1.4×10^{-09}	7.8×10^{-24}	0.0515	(p-value)	6.9×10^{-95}	2.5×10^{-92}	1.7×10^{-19}	2.0×10^{-04}
AFCM-ER	**0.7052 ± 0.1006**	**0.3766 ± 0.1306**	0.4353 ± 0.0474	**0.7441 ± 0.0250**	AFCM-ER	0.0025 ± 0.0000	0.6919 ± 0.0000	0.0086 ± 0.0000	0.4996 ± 0.0000
Haberman					Thyroid gland				
FCM	0.0689 ± 0.0000	0.6578 ± 0.0000	**0.5980 ± 0.0000**	0.5752 ± 0.0000	FCM	0.2362 ± 0.0012	0.8347 ± 0.0011	**0.4679 ± 0.0034**	0.5791 ± 0.0019
(p-value)	0	0	0	0	(p-value)	3.1×10^{-70}	1.1×10^{-10}	1.0×10^{-61}	5.3×10^{-109}
FCM-ER	0.0016 ± 0.0015	0.6924 ± 0.0008	0.0308 ± 0.0179	0.6050 ± 0.0026	FCM-ER	0.0012 ± 0.0008	1.0974 ± 0.0008	0.0317 ± 0.0140	0.5439 ± 0.0053
(p-value)	4.7×10^{-74}	2.6×10^{-78}	1.2×10^{-06}	1.6×10^{-24}	(p-value)	1.0×10^{-243}	5.7×10^{-116}	2.2×10^{-64}	1.3×10^{-110}
FCCI	**0.4251 ± 0.0144**	**0.4283 ± 0.0083**	0.1801 ± 0.0312	0.5715 ± 0.0085	FCCI	0.0882 ± 0.0114	0.9916 ± 0.0112	0.1639 ± 0.0552	0.6492 ± 0.0131
(p-value)	1.1×10^{-146}	6.9×10^{-150}	1.5×10^{-67}	1.9×10^{-66}	(p-value)	1.9×10^{-114}	8.5×10^{-93}	9.4×10^{-23}	8.9×10^{-54}
AFCM-ER	0.0096 ± 0.0000	0.6879 ± 0.0000	0.0400 ± 0.0000	**0.6086 ± 0.0000**	AFCM-ER	**0.2420 ± 0.0004**	**0.8205 ± 0.0198**	0.2572 ± 0.0547	**0.7080 ± 0.0109**
Iris plant					UKM				
FCM	0.2020 ± 0.0000	0.9029 ± 0.0000	0.6895 ± 0.0000	0.5725 ± 0.0000	FCM	0.0003 ± 0.0002	1.3858 ± 0.0003	0.0249 ± 0.0133	0.2810 ± 0.0034
(p-value)	1.6×10^{-229}	3.5×10^{-261}	7.7×10^{-111}	1.6×10^{-242}	(p-value)	1.2×10^{-91}	2.8×10^{-86}	6.7×10^{-103}	1.7×10^{-106}
FCM-ER	0.4443 ± 0.0004	0.5737 ± 0.0003	**0.7138 ± 0.0122**	0.7520 ± 0.0003	FCM-ER	0.1548 ± 0.0073	1.1535 ± 0.0102	0.1913 ± 0.0047	0.4907 ± 0.0072
(p-value)	9.5×10^{-161}	1.0×10^{-198}	5.7×10^{-78}	1.3×10^{-171}	(p-value)	3.7×10^{-43}	5.6×10^{-48}	2.9×10^{-61}	4.0×10^{-39}
FCCI	0.2226 ± 0.0286	0.8410 ± 0.0420	0.1146 ± 0.0211	0.5167 ± 0.0455	FCCI	**0.2451 ± 0.0082**	**1.0015 ± 0.0109**	**0.1963 ± 0.0050**	**0.5555 ± 0.0128**
(p-value)	6.1×10^{-101}	5.0×10^{-96}	2.9×10^{-140}	3.7×10^{-79}	(p-value)	2.6×10^{-11}	0.8833	2.3×10^{-63}	4.7×10^{-07}
AFCM-ER	**0.5084 ± 0.0015**	**0.4687 ± 0.0010**	0.6392 ± 0.0040	**0.7867 ± 0.0008**	AFCM-ER	0.2241 ± 0.0280	**1.0023 ± 0.0546**	0.1696 ± 0.0032	0.5426 ± 0.0229
Led7					Vehicle				
FCM	0.6165 ± 0.0000	0.8353 ± 0.0000	**0.8049 ± 0.0000**	0.8180 ± 0.0000	FCM	**0.8821 ± 0.0045**	**0.1535 ± 0.0045**	0.4148 ± 0.0055	**0.6481 ± 0.0013**
(p-value)	4.4×10^{-99}	5.8×10^{-121}	6.4×10^{-113}	1.5×10^{-08}	(p-value)	4.9×10^{-217}	3.1×10^{-231}	1.8×10^{-90}	7.4×10^{-191}
FCM-ER	0.4169 ± 0.0020	1.1047 ± 0.0089	0.4849 ± 0.0116	0.7816 ± 0.0028	FCM-ER	0.4663 ± 0.0005	0.6886 ± 0.0005	**0.5015 ± 0.0085**	0.5669 ± 0.0004
(p-value)	3.1×10^{-118}	4.6×10^{-133}	3.1×10^{-64}	1.0×10^{-37}	(p-value)	2.5×10^{-222}	5.3×10^{-254}	4.3×10^{-101}	1.7×10^{-172}
FCCI	0.7491 ± 0.0194	0.4608 ± 0.0280	0.5429 ± 0.0214	**0.8760 ± 0.0044**	FCCI	0.6558 ± 0.0071	0.4396 ± 0.0072	0.4248 ± 0.0046	0.6194 ± 0.0024
(p-value)	3.5×10^{-76}	8.4×10^{-91}	9.4×10^{-73}	1.4×10^{-31}	(p-value)	3.4×10^{-184}	5.4×10^{-198}	6.0×10^{-92}	3.4×10^{-171}
AFCM-ER	**0.9653 ± 0.0367**	**0.0504 ± 0.0495**	0.3342 ± 0.0358	0.8332 ± 0.0246	AFCM-ER	0.1446 ± 0.0018	1.1338 ± 0.0011	0.0678 ± 0.0440	0.4559 ± 0.0021

mance of the proposed algorithm, in the Breast tissue and UKM datasets, there is no significant difference with the FCCI algorithm for FRHR and PE values respectively. Figure 11 and Figure 12 show the performance behavior of applying the four algorithms on Breast tissue and UKM datasets as box plots for MPC, PE, SFS and FRHR values.



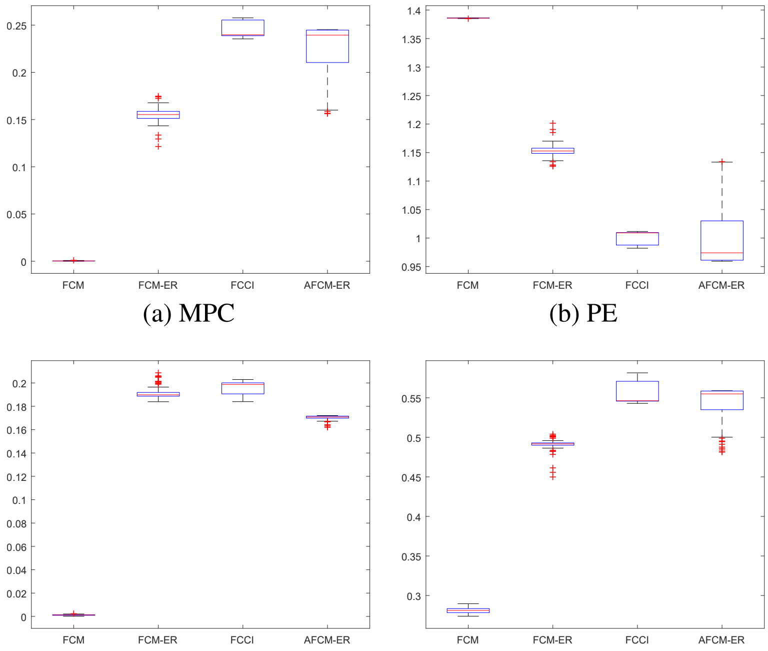
(a) MPC

(b) PE

(c) SFS

(d) FRHR

Figure 11 – Clustering results on Breast tissue dataset using different metrics.



(a) MPC

(b) PE

(c) SFS

(d) FRHR

Figure 12 – Clustering results on UKM dataset using different metrics.

Concerning to the comparison between the hard partition and the a priori partition, Table 18 shows the results considering ARI, FM, OERC, SCS and the performance rank (in parenthesis) of each algorithm on real datasets.

By comparing the results in Table 18, it is further noticed that the best clustering performance indicated by some index is not always consistent with others indexes, i.e., an algorithm showing good clustering performance with a high index value may not have a high value in the other indexes as well. Example of this can be observed in Breast tissue dataset where the proposed algorithm performance rank for ARI is (2), for FM is (1), (4) for OERC and (1) for SCS. Therefore, it is necessary to evaluate the performance of a clustering algorithm with different metrics.

Table 18 – Cluster validity measures on real datasets for hard partitions.

Algorithms	ARI	FM	OERC	SCS	Algorithms	ARI	FM	OERC	SCS
		Abalone					Scale		
FCM	0.1333(4)	0.4285(4)	0.3894(3)	**0.5471**(1)	FCM	**0.1279**(1)	**0.4628**(1)	**0.4173**(1)	0.0008(4)
FCM-ER	**0.1938**(1)	**0.5041**(1)	0.3905(4)	0.1300(4)	FCM-ER	0.0062(4)	0.3803(4)	0.4739(4)	0.0047(3)
FCCI	0.1904(2)	0.4811(2)	0.3746(2)	0.3443(2)	FCCI	0.0574(3)	0.4152(3)	0.4501(3)	0.0669(2)
AFCM-ER	0.1855(3)	0.4730(3)	**0.3731**(1)	0.2835(3)	AFCM-ER	0.1131(2)	0.4601(2)	0.4259(2)	**0.1559**(1)
		Breast tissue					Sonar		
FCM	0.3019(3)	0.4286(3)	0.2099(2)	0.4782(2)	FCM	0.0085(2)	0.5036(3)	0.4957(2)	**0.2626**(1)
FCM-ER	**0.3208**(1)	0.4422(2)	**0.2013**(1)	0.4105(4)	FCM-ER	-0.0015(3)	0.5063(2)	0.5007(3)	0.0211(3)
FCCI	0.2862(4)	0.4213(4)	0.2246(3)	0.4691(3)	FCCI	-0.0047(4)	0.4967(4)	0.5024(4)	0.0281(2)
AFCM-ER	0.3031(2)	**0.4520**(1)	0.2584(4)	**0.5042**(1)	AFCM-ER	**0.0190**(1)	**0.5268**(1)	**0.4905**(1)	0.0068(4)
		Haberman					Thyroid gland		
FCM	-0.0011(3)	0.5480(3)	0.5009(3)	**0.4839**(1)	FCM	0.1801(2)	0.5640(2)	0.4140(2)	**0.3241**(1)
FCM-ER	-0.0027(4)	0.5473(4)	0.5016(4)	0.0634(3)	FCM-ER	0.0577(3)	0.4670(4)	0.4788(3)	0.0455(3)
FCCI	**0.1530**(1)	0.6641(2)	0.4054(2)	0.1663(2)	FCCI	0.0464(4)	0.4851(3)	0.4822(4)	-0.0174(4)
AFCM-ER	0.1456(2)	**0.7070**(1)	**0.3874**(1)	0.0522(4)	AFCM-ER	**0.6933**(1)	**0.8510**(1)	**0.1534**(1)	0.1611(2)
		Iris plants					UKM		
FCM	0.6303(2)	0.7520(2)	0.1632(2)	**0.5679**(1)	FCM	**0.2630**(1)	**0.5156**(1)	0.3563(4)	0.0249(4)
FCM-ER	0.6199(3)	0.7449(3)	0.1678(3)	0.2599(3)	FCM-ER	0.1644(4)	0.3820(4)	0.3224(3)	**0.1336**(1)
FCCI	0.1853(4)	0.4534(4)	0.3597(4)	0.0577(4)	FCCI	0.2349(3)	0.4394(3)	0.2997(2)	0.1256(2)
AFCM-ER	**0.6882**(1)	**0.7909**(1)	**0.1377**(1)	0.3977(2)	AFCM-ER	0.2565(2)	0.4596(2)	**0.2952**(1)	0.1200(3)
		Led7					Vehicle		
FCM	**0.5142**(1)	**0.5631**(1)	**0.0879**(1)	**0.5563**(1)	FCM	0.0770(2)	0.3082(4)	**0.3466**(1)	**0.3754**(1)
FCM-ER	0.4833(2)	0.5358(2)	0.0944(2)	0.3481(4)	FCM-ER	0.0736(4)	0.3085(2)	0.3507(3)	0.2676(3)
FCCI	0.4589(3)	0.5138(3)	0.0988(3)	0.4889(2)	FCCI	0.0767(3)	0.3084(3)	0.3472(2)	0.3455(2)
AFCM-ER	0.3139(4)	0.3930(4)	0.1431(4)	0.3919(3)	AFCM-ER	**0.1143**(1)	**0.3558**(1)	0.3543(4)	0.0221(4)

The average performance ranking of the clustering algorithms for hard partitions was computed from Table 18. The results according to the performance ranking of the clustering algorithms, the average performance ranking and the OAR are presented in Table 19.

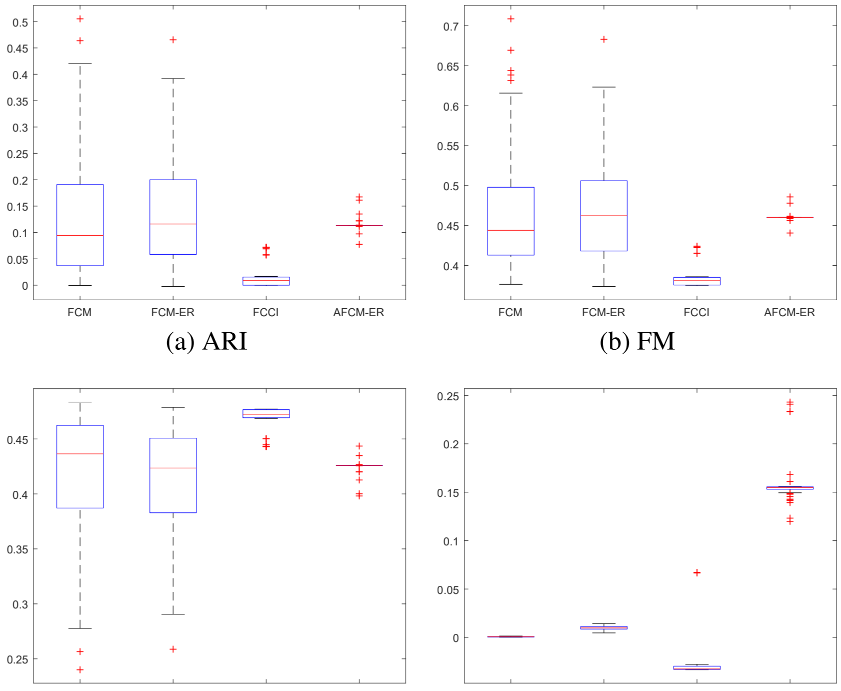
Table 19 – Average performance ranking for hard partition on real datasets.

Algorithm	ARI	FM	OERC	SCS	OAR
FCM	2.1(2)	2.4(2)	2.1(2)	**1.7**(1)	1.8(2)
FCM-ER	2.9(3)	2.8(3)	3.0(4)	3.1(4)	3.5(4)
FCCI	3.1(4)	3.1(4)	2.9(3)	2.5(2)	3.3(3)
AFCM-ER	**1.9**(1)	**1.7**(1)	**2.0**(1)	2.7(3)	**1.5**(1)

The AFCM-ER algorithm for hard partitions, performed better than the other three algorithms for most of the datasets. For example AFCM-ER algorithm in the Thyroid gland dataset performed more than 25% better in ARI, FM and OERC. Moreover, the FCM and FCCI algorithms achieved, respectively, the second and third best average rankings. The FCM-ER algorithm presented the worst performance according to the average ranking. Finally, the proposed algorithm, in general, performed better with a higher frequency according to considered indexes.

Table 20 reports the clustering results expressed in terms of the means, standard deviations and *p*-values of the considered index values obtained by the fuzzy clustering algorithms on the datasets for hard partitions.

The results expose that AFCM-ER algorithm significantly outperformed the other three algorithms in most results. However, even though FCM, FCM-ER and FCCI are in general inferior, they are able to achieve the good clustering performances for the Abalone, Led7, Scale and UKM datasets. In the case of Scale dataset, results in Table 20 show that the AFCM-ER algorithm is not significantly different in comparison with FCM algorithm for ARI, FM and OERC indexes. Figure 13 shows the box plot of ARI, FM, OERC, and SCS values for Scale dataset. Although there is no significant difference, the proposed method shows less degree of dispersion than the FCM algorithm, so it is much more robust (stable) for this dataset.



(a) ARI

(b) FM

(c) OERC

(d) SCS

Figure 13 – Box plot for Scale dataset.

On the other hand, FCM-ER algorithm shows the best performance in ARI for Breast tissue dataset (Table 18). However in Table 20, AFCM-ER algorithm significantly outperformed in ARI. These results indicate that there is no single algorithm that is always superior to the others for all datasets. In conclusion, the proposed algorithm obtained the best clustering results for hard partitions compared with the literature proposals, indicating that automatic variable selection and entropy regularization is an effective way to enhance the performance of the clustering algorithms.

Table 20 – Clustering performance on real datasets for hard partitions. The first value corresponds to the mean and the second value is the standard deviation of 100 results.

Algorithms	ARI	FM	OERC	SCS	Algorithms	ARI	FM	OERC	SCS
		Abalone					Scale		
FCM	0.1331 ± 0.0001	0.4284 ± 0.0001	0.3895 ± 0.0000	**0.5471 ± 0.0000**	FCM	**0.1265 ± 0.1159**	**0.4657 ± 0.0704**	**0.4190 ± 0.0560**	0.0007 ± 0.0003
(p-value)	7.4×10^{-132}	8.8×10^{-151}	1.2×10^{-104}	1.4×10^{-79}	(p-value)	0.2916	0.4535	0.2607	6.1×10^{-96}
FCM-ER	**0.1911 ± 0.0009**	**0.5079 ± 0.0030**	0.3970 ± 0.0030	0.0745 ± 0.0287	FCM-ER	0.0144 ± 0.0198	0.3849 ± 0.0135	0.4699 ± 0.0092	−0.0211 ± 0.0311
(p-value)	6.3×10^{-23}	2.7×10^{-98}	6.1×10^{-102}	1.6×10^{-87}	(p-value)	0.0410	0.2246	0.0231	1.1×10^{-93}
FCCI	0.1902 ± 0.0013	0.4812 ± 0.0003	0.3749 ± 0.0011	0.3235 ± 0.0283	FCCI	0.0144 ± 0.0198	0.3849 ± 0.0135	0.4699 ± 0.0092	−0.0211 ± 0.0311
(p-value)	1.3×10^{-13}	1.8×10^{-75}	1.1×10^{-33}	0.5434	(p-value)	1.6×10^{-70}	5.6×10^{-77}	5.9×10^{-68}	9.1×10^{-71}
AFCM-ER	0.1877 ± 0.0027	0.4733 ± 0.0014	**0.3713 ± 0.0017**	0.3263 ± 0.0369	AFCM-ER	**0.1140 ± 0.0086**	**0.4603 ± 0.0037**	**0.4254 ± 0.0045**	**0.1565 ± 0.0177**
		Breast tissue					Sonar		
FCM	0.2942 ± 0.0136	0.4230 ± 0.0101	0.2138 ± 0.0110	**0.4808 ± 0.0140**	FCM	0.0061 ± 0.0043	0.5049 ± 0.0030	0.4969 ± 0.0022	**0.2600 ± 0.0046**
(p-value)	1.0×10^{-23}	2.3×10^{-37}	0.8425	3.1×10^{-37}	(p-value)	6.0×10^{-51}	4.8×10^{-88}	7.6×10^{-51}	2.4×10^{-174}
FCM-ER	0.2964 ± 0.0099	0.4187 ± 0.0095	0.2030 ± 0.0007	0.3242 ± 0.0350	FCM-ER	−0.0014 ± 0.0006	0.5065 ± 0.0011	0.5007 ± 0.0003	0.0110 ± 0.0060
(p-value)	1.7×10^{-22}	7.3×10^{-40}	0.0021	0.0043	(p-value)	1.7×10^{-155}	2.3×10^{-127}	2.2×10^{-155}	4.6×10^{-10}
FCCI	0.3046 ± 0.0086	0.4237 ± 0.0045	**0.1980 ± 0.0071**	0.3452 ± 0.0459	FCCI	0.0085 ± 0.0167	0.5084 ± 0.0119	0.4957 ± 0.0084	0.0173 ± 0.0129
(p-value)	9.9×10^{-18}	5.7×10^{-38}	3.2×10^{-05}	7.2×10^{-06}	(p-value)	1.1×10^{-08}	4.9×10^{-28}	1.1×10^{-08}	1.2×10^{-12}
AFCM-ER	**0.3443 ± 0.0366**	**0.4714 ± 0.0224**	0.2145 ± 0.0367	0.2962 ± 0.0888	AFCM-ER	**0.0190 ± 0.0000**	**0.5268 ± 0.0000**	**0.4905 ± 0.0000**	0.0068 ± 0.0000
		Haberman					Thyroid gland		
FCM	−0.0011 ± 0.0000	0.5480 ± 0.0000	0.5009 ± 0.0000	**0.4838 ± 0.0000**	FCM	0.1859 ± 0.0184	0.5673 ± 0.0101	0.4111 ± 0.0093	**0.3241 ± 0.0010**
(p-value)	0	0	0	0	(p-value)	1.7×10^{-07}	7.1×10^{-08}	9.8×10^{-08}	2.4×10^{-86}
FCM-ER	−0.0027 ± 0.0000	0.5473 ± 0.0000	0.5016 ± 0.0000	0.0276 ± 0.0160	FCM-ER	0.0514 ± 0.0068	0.4882 ± 0.0183	0.4796 ± 0.0046	0.0281 ± 0.0105
(p-value)	0	0	0	4.8×10^{-28}	(p-value)	1.5×10^{-17}	1.1×10^{-17}	1.5×10^{-17}	2.9×10^{-58}
FCCI	0.1320 ± 0.0313	0.6573 ± 0.0198	0.4147 ± 0.0173	0.1680 ± 0.0301	FCCI	0.0610 ± 0.0286	0.5248 ± 0.0253	0.4716 ± 0.0154	−0.0343 ± 0.0100
(p-value)	3.5×10^{-05}	1.2×10^{-44}	9.1×10^{-29}	2.8×10^{-61}	(p-value)	1.1×10^{-16}	4.8×10^{-13}	2.9×10^{-16}	7.2×10^{-77}
AFCM-ER	**0.1456** ± 0.0000	**0.7070 ± 0.0000**	**0.3874 ± 0.0000**	0.0522 ± 0.0000	AFCM-ER	**0.3545 ± 0.2911**	**0.6625 ± 0.1620**	**0.3253 ± 0.1477**	0.1276 ± 0.0280
		Iris plant					UKM		
FCM	0.6303 ± 0.0000	0.7520 ± 0.0000	0.1632 ± 0.0000	**0.5679 ± 0.0000**	FCM	**0.2582 ± 0.0168**	**0.5018 ± 0.0190**	0.3428 ± 0.0099	0.0126 ± 0.0062
(p-value)	0	0	0	1.9×10^{-141}	(p-value)	5.5×10^{-16}	1.7×10^{-39}	6.1×10^{-52}	4.2×10^{-104}
FCM-ER	0.6199 ± 0.0000	0.7449 ± 0.0000	0.1678 ± 0.0000	0.3022 ± 0.0115	FCM-ER	0.1613 ± 0.0064	0.3807 ± 0.0053	0.3245 ± 0.0030	0.1189 ± 0.0122
(p-value)	0	0	0	7.4×10^{-86}	(p-value)	3.8×10^{-49}	6.6×10^{-51}	2.3×10^{-42}	0.0043
FCCI	0.1408 ± 0.0571	0.4268 ± 0.0341	0.3817 ± 0.0282	0.0236 ± 0.0437	FCCI	0.1909 ± 0.0449	0.4056 ± 0.0362	0.3157 ± 0.0149	**0.1379 ± 0.0082**
(p-value)	1.7×10^{-99}	4.6×10^{-104}	3.8×10^{-95}	2.4×10^{-93}	(p-value)	2.4×10^{-13}	5.7×10^{-15}	2.2×10^{-09}	7.2×10^{-36}
AFCM-ER	**0.6882 ± 0.0000**	**0.7909 ± 0.0000**	**0.1377 ± 0.0000**	0.3958 ± 0.0067	AFCM-ER	0.2290 ± 0.0228	0.4395 ± 0.0191	**0.3061 ± 0.0071**	0.1149 ± 0.0067
		Led7					Vehicle		
FCM	**0.5142 ± 0.0000**	**0.5631 ± 0.0000**	**0.0879 ± 0.0000**	**0.5563 ± 0.0000**	FCM	0.0790 ± 0.0079	0.3103 ± 0.0080	**0.3464 ± 0.0008**	**0.3774 ± 0.0065**
(p-value)	1.3×10^{-84}	4.9×10^{-90}	4.3×10^{-53}	4.5×10^{-76}	(p-value)	3.7×10^{-68}	1.2×10^{-75}	4.4×10^{-33}	6.6×10^{-112}
FCM-ER	0.4818 ± 0.0047	0.5345 ± 0.0042	0.0947 ± 0.0009	0.3371 ± 0.0145	FCM-ER	0.0729 ± 0.0003	0.3075 ± 0.0003	0.3505 ± 0.0001	0.2784 ± 0.0041
(p-value)	7.5×10^{-79}	7.5×10^{-84}	1.2×10^{-49}	0.2768	(p-value)	1.3×10^{-126}	1.3×10^{-125}	0.0167	5.2×10^{-97}
FCCI	0.4606 ± 0.0113	0.5159 ± 0.0097	0.0995 ± 0.0032	0.4835 ± 0.0278	FCCI	0.0754 ± 0.0047	0.3076 ± 0.0035	0.3478 ± 0.0018	0.3417 ± 0.0041
(p-value)	1.2×10^{-71}	2.3×10^{-75}	7.8×10^{-47}	7.7×10^{-52}	(p-value)	4.0×10^{-87}	1.2×10^{-102}	1.3×10^{-17}	1.2×10^{-108}
AFCM-ER	0.2760 ± 0.0353	0.3641 ± 0.0259	0.1639 ± 0.0243	0.3323 ± 0.0407	AFCM-ER	**0.1161 ± 0.0024**	**0.3550 ± 0.0027**	0.3511 ± 0.0024	0.0534 ± 0.0238

4.4 Color segmentation

The segmentation of color images is a potential area of research due to its practical significance in various fields. Image segmentation partitions the image into regions/segments such that pixels belonging to a region are more similar to each other than those belonging to the different regions. Clustering is a well-known approach for segmenting images. It strives to assess the relationships among patterns of the dataset by organizing them into groups or clusters such that objects within a group are more similar to each other than those belonging to different clusters. Clustering in the color domain gives improved segmentation results since color components carry more information than the grayscale components.

In this section, the proposed algorithm is experimented with the Berkeley segmentation database (ARBELAEZ; FOWLKES; MARTIN, 2007) to evaluate its clustering performance and robustness in image segmentation. It is also carried out the experiments with the other approaches on the image dataset. This segmentation application was performed as in the (HANMANDLU et al., 2013) work.

All images are digitized to 24 bits per pixel in the RGB format. Since the distance between any two points in the RGB space is not proportional to their color difference, the transformation from the RGB space to a uniform color space: $CIELAB$ is performed. The vector $\{a^*, b^*\}$ of $CIELAB$ color space contains the total chrominance color information of pixels and is the feature space for our color segmentation experiments. The vector $\{L^*\}$ or luminance vector which decides the darkness or fairness of the image segments is discarded in the clustering process to ensure that the illumination effects do not affect the segmentation process.

4.4.1 Segmentation evaluation indices

The quality of the segmentation is generally judged by two types of indices: the goodness methods such as Liu's F-measure which ascertains the color difference in the $CIELAB$ color space and also penalizes the formation of large number of segments, and the discrepancy methods which ascertain the quality with respect to some reference result like ground truth images for example the RAND index. The above two types of quality measures are used together to judge the efficiency and practicality of the proposed algorithms.

Liu's Evaluation Measure (F): The performance of color segmentation is evaluated using Liu and Yang's (LIU; YANG, 1994) evaluation function F:

$$F(I) = \frac{1}{1000(N_1 \times N_2)} \sqrt{G} \sum_{i=1}^{G} \frac{e_i^2}{\sqrt{A_i}} \tag{4.18}$$

where I is the segmented image and $N_1 \times N_2$ is the image size, G is the number of regions of the segmented image, A_i is the area and e_i is the average color error of the i-th region where e_i is defined as the sum of Euclidean distances between the $\{a^*, b^*\}$ color vector of the pixels of region i and the color centroid attributed to region i in the segmented image. The smaller the value of $F(I)$ the better the segmentation result. Liu's F-factor was chosen as one of our evaluation criteria since it gives an accurate measure of the color differencing achieved by the segmentation algorithm and at the same time penalizes large number of regions formed. Also is used for comparative proposed algorithms with FCCI algorithm.

Probabilistic Rand index (PR): The PR index is a generalization of the RAND index (RAND, 1971). It allows a comparison of the test segmentation with multiple ground truth images through soft non-uniform weighting of pixel pairs as a function of the variability in the ground truth sets. Consider a set of manually segmented (ground truth) images $\{S_1, ..., S_k\}$ corresponding to an image $X = \{x_1, ..., x_N\}$, where a subscript indexes one of N pixels. Let S_{test} be the obtained segmentation compared with the manually labeled set. Let $l_i^{S_{test}}$ be the label of pixel x_i in segmentation S_{test} and $l_i^{S_k}$ the label of same pixel in the manually segmented image S_k. It is assumed that each label $l_i^{S_k}$ can take values in a discrete set of size L_k, and correspondingly $l_i^{S_{test}}$ takes one of L_{test} values.

Each human segmenter provides information about the segmentation S_k of the image in the form of binary numbers $I(l_i^{S_k} = l_j^{S_k})$ for each pair of pixels (x_i, x_j) (UNNIKRISHNAN; HEBERT, 2005). The set of all perceptually correct segmentation defines a Bernoulli distribution giving a random variable with the expected value denoted as h_{ij}. The PR is then defined as:

$$PR(S_{test}, \{S_k\}) = \frac{2}{N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} [m_{ij} h_{ij} + (1 - m_{ij})(1 - h_{ij})] \tag{4.19}$$

where m_{ij} denotes the event of a pair of pixels i and j having the same label in the test image S_{test} and h_{ij} denotes the event of a pair of pixels i and j having the same label in $\{S_k\}$:

$$m_{ij} = I(l_i^{S_{test}} = l_j^{S_{test}}) \tag{4.20}$$

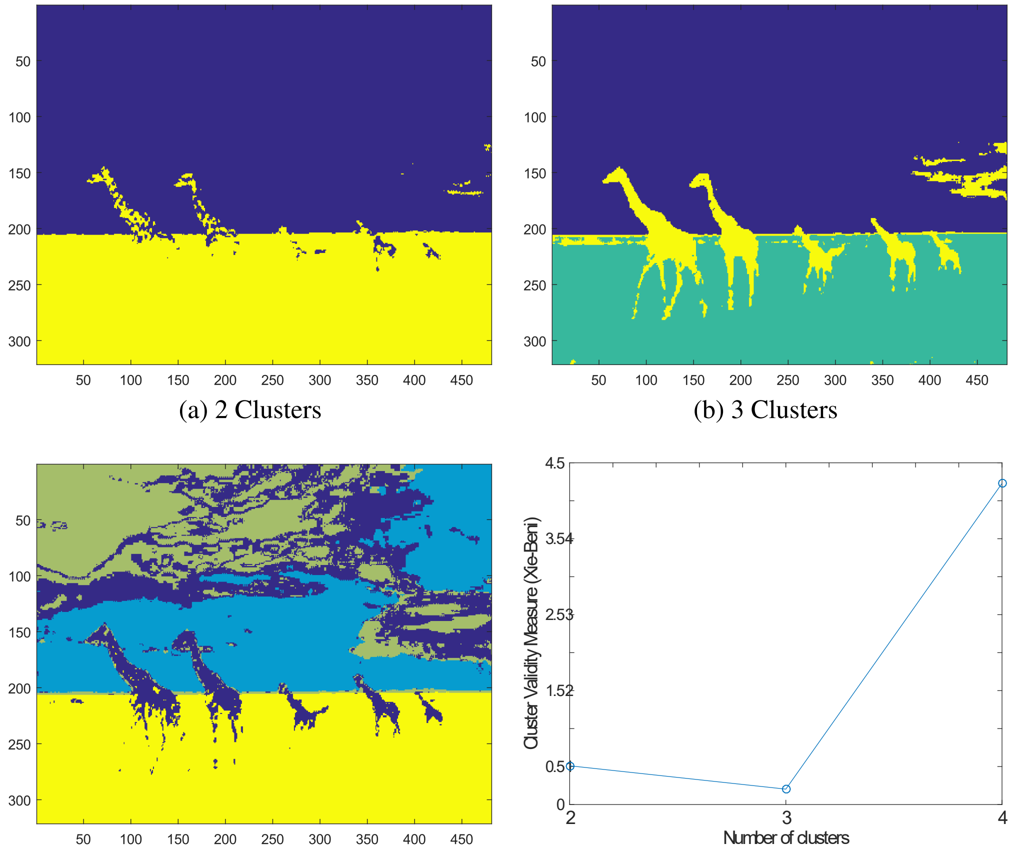
This measure takes values $[0, 1]$, where 0 means no similarities between S_{test} and $\{S_1, S_2, ..., S_k\}$, and 1 means all segmentations are identical.

4.4.2 Experimental setting

A set of ten test images is taken from the Berkeley segmentation database (ARBELAEZ; FOWLKES; MARTIN, 2007) along with 5–7 ground truth segmentations available for each image in the database for the evaluation of the results. The size of each image is either 321 × 481 or 481 × 321. In these experiments, all the datasets are preprocessed by normalizing the feature in each dimension into the interval $[-1, 1]$.

For each dataset, the value of the T_u parameter of proposed algorithm was varied between 10^{-7} to 10 (with step 0.1). The algorithms were executed on each dataset 100 times, and the cluster centers were randomly initialized at each time.

The number of clusters C for each image was determined from the first local minimum in the cluster validity according to Xie-Beni index as demonstrated by the example shown in Figure 14. In this example, the number of clusters C was varied from 2 to 4. The corresponding values of Xie-Beni index for $C = 2, ..., 4$ are 0.5103, 0.2028 and 4.2291 respectively. It can be observed a local minimum in $C = 3$; therefore the selected C value in this image is equal to 3.



(a) 2 Clusters

(b) 3 Clusters

(c) 4 Clusters

(d) Xie-Beni values

Figure 14 – Segmentation results (a) for two clusters, (b) for three clusters, (c) for four clusters and (d) the corresponding Clustering Validity Measure (Xie-Beni index).

The steps for color image segmentation with automatic cluster amount selection are outlined in Algorithm 5:

The results shows that the AFCM-ER algorithm yields highly crisp values of object membership function u_{ik} (close to 0 and 1). On the other hand the feature membership values v_{kj} are highly fuzzy for high values due to restriction Equation 3.6 for AFCM-ER algorithm. The v_{kj} values however are accentuated by the lower values of parameter T_u in Equation 3.4 creating a considerable influence in the computation of u_{ik}, eventually leading to crisp values of u_{ik} after the iterative procedure. This helps in crisp classification during the defuzzification process.

Algorithm 5 Algorithm for color image segmentation with automatic cluster amount selection.

```
Obtain the three dimensional $RGB$ input image;
Convert $RGB$ color space into the $CIELAB$ color space with color dimensions $P = 2$, i.e,
  $\{a^*, b^*\}$;
Perform $2D$ to $1D$ transformation (CHACON; AGUILAR; DELGADO, 2002) (by lifting the
  elements column wise) to generate data point $x_{ij}$ in the $j$-th dimension, $j = 1, 2$ for each
  pixel $i = 1, ..., N$, where $N$ is the size of the data;
    $C = 1$;
    $S_{old} = \infty$;
    $S_{new} = \infty$;
while $S_{old} > S_{new}$ do
    $S_{old} = S_{new}$;
    $C = C + 1$;
    Find $T_u$ following the same procedure used in subsection 4.1.11;
    Run the algorithm and obtain fuzzy object membership function $u_{ik}$;
    Calculate Xie and Beni's (XIE; BENI, 1991) cluster validity and assign value to $S_{new}$;
end while
Run the algorithm for $C$-1 clusters and obtain the object $u_{ik}$ membership function.
Defuzzify $u_{ik}$ into clusters.
```

4.4.3 Results and analysis

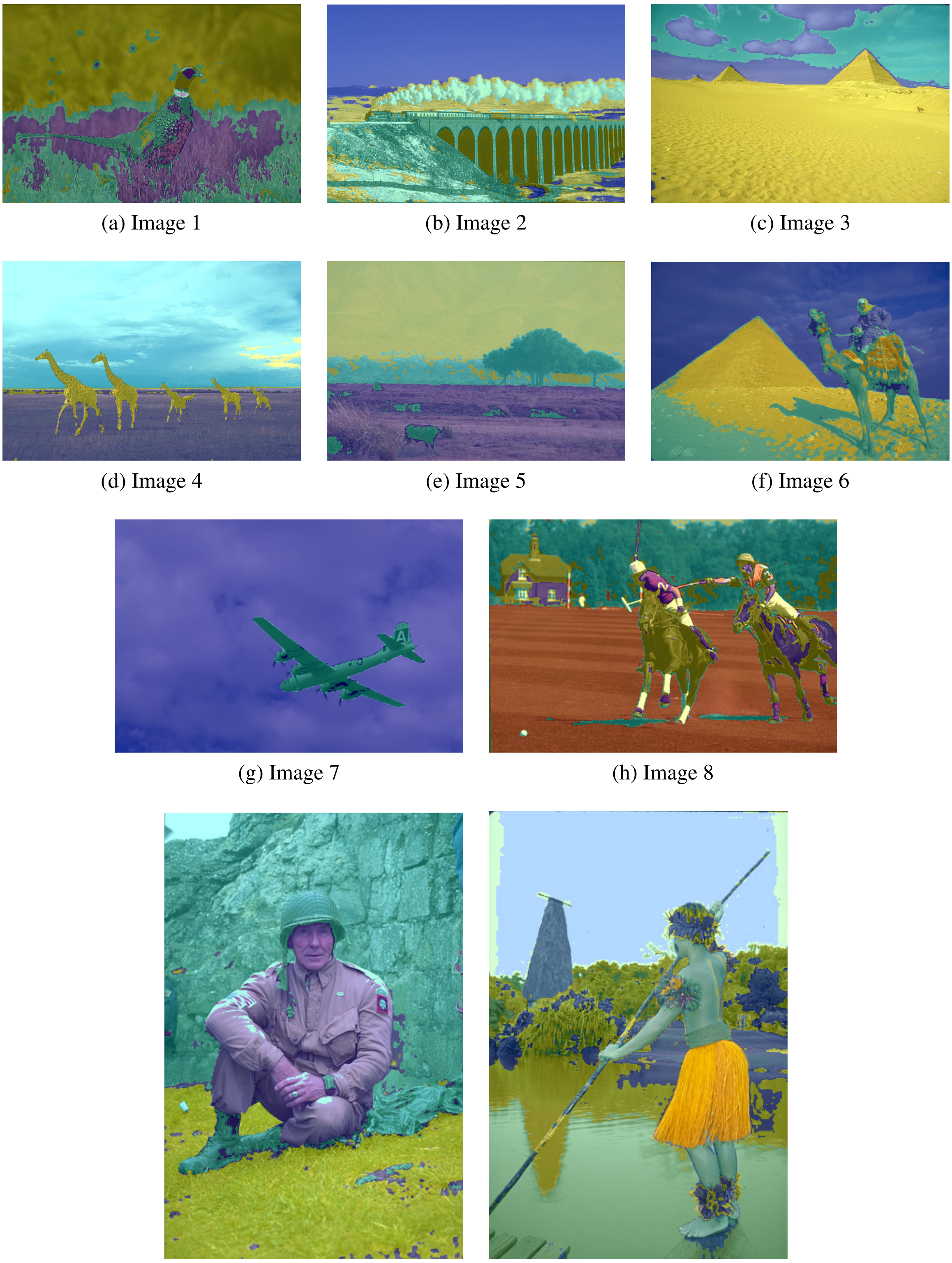
In this subsection are shown and analyzed the segmentation results obtained by the FCM, the FCM-ER, the FCCI and the proposed AFCM-ER algorithms on input images.

The obtained segmentation results by the proposed algorithm are presented in Figure 15. The segmentation shows that proposed method has a good match with human ground truth segmentation as indicated by a high value of PR index (there is an excellent correspondence with human perception), and also efficient color differentiation as indicated by a low value of Liu's evaluation measure F. The algorithm is also able to segment natural scenes containing non-uniform illumination efficiently (Figure 15) by segregating shadows from sunlit portions thus agreeing with human perception.

Table 21 shows Liu's F-factor and PR values obtained by FCM, FCM-ER, FCCI and AFCM-ER algorithms. In this table it is also presented the number of clusters obtained automatically for the segmentation algorithm using the Xie-Beni index.

Table 21 clearly indicates that the AFCM-ER algorithm is the best among all these algorithms, and respect to the PR index, the proposed approach gets the best performance (higher PR) for all images. For PR index the worst result was obtained by FCM algorithm and for Liu's evaluation measure FCM-ER algorithm.

Figure 16 and Figure 17 show the color segmented results of the four methods for ten images from the Berkeley Segmentation dataset. It is observed from the segmentation results that the proposed method forms well defined and interpretable clusters even when the color difference between two regions is not too distinct as in the case of the last image.



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

(e) Image 5

(f) Image 6

(g) Image 7

(h) Image 8

(i) Image 9

(j) Image 10

Figure 15 – Color segmentation results of 10 test images from Berkeley segmentation dataset by the proposed method.

Table 21 – PR and Liu's F-measure values for the obtained images segmentation from Berkeley dataset by the proposed algorithm (AFCM-ER) and the other approaches.

Algorithms	PR	Liu's F-measure	Number of clusters	Algorithms	PR	Liu's F-measure	Number of clusters
Image 1				Image 6			
FCM	0.7954	0.0079	2	FCM	0.7827	**0.0039**	2
FCM-ER	0.7890	0.0083	2	FCM-ER	0.7819	0.0116	2
FCCI	0.7139	**0.0041**	4	FCCI	0.7819	0.0091	2
AFCM-ER	**0.8249**	0.0065	3	AFCM-ER	**0.8224**	0.0048	3
Image 2				Image 7			
FCM	0.7048	0.0065	2	FCM	0.4245	0.0011	3
FCM-ER	0.8086	0.0034	3	FCM-ER	0.5301	0.0012	2
FCCI	0.8076	0.0034	3	FCCI	0.6731	**0.0008**	3
AFCM-ER	**0.8148**	**0.0030**	3	AFCM-ER	**0.8712**	0.0010	2
Image 3				Image 8			
FCM	0.8889	**0.0013**	2	FCM	0.7404	0.0070	2
FCM-ER	0.7714	0.0059	3	FCM-ER	0.7970	0.0056	3
FCCI	0.8948	0.0019	3	FCCI	0.7987	0.0057	3
AFCM-ER	**0.8970**	0.0016	3	AFCM-ER	**0.8695**	**0.0037**	4
Image 4				Image 9			
FCM	0.9426	0.0040	2	FCM	0.8447	0.0082	3
FCM-ER	0.7605	0.0023	3	FCM-ER	0.8455	0.0082	3
FCCI	0.7830	**0.0022**	4	FCCI	0.8400	0.0107	3
AFCM-ER	**0.9633**	0.0076	3	AFCM-ER	**0.8497**	**0.0075**	3
Image 5				Image 10			
FCM	0.6955	**0.0026**	2	FCM	0.6201	0.0096	3
FCM-ER	0.7002	0.0069	2	FCM-ER	0.6642	0.0049	4
FCCI	0.7002	0.0069	2	FCCI	0.6581	**0.0044**	4
AFCM-ER	**0.8282**	0.0030	3	AFCM-ER	**0.8347**	0.0045	6

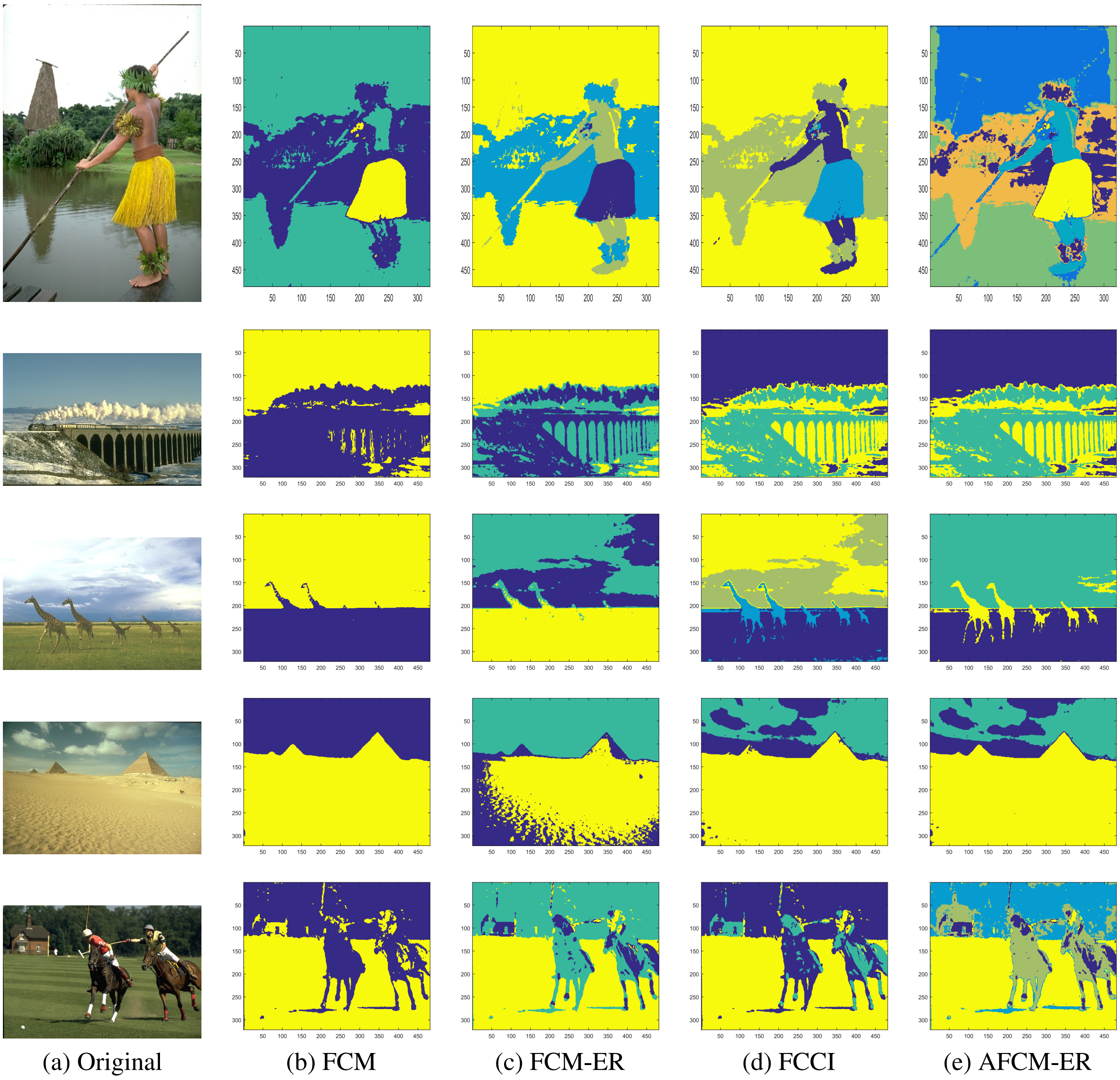
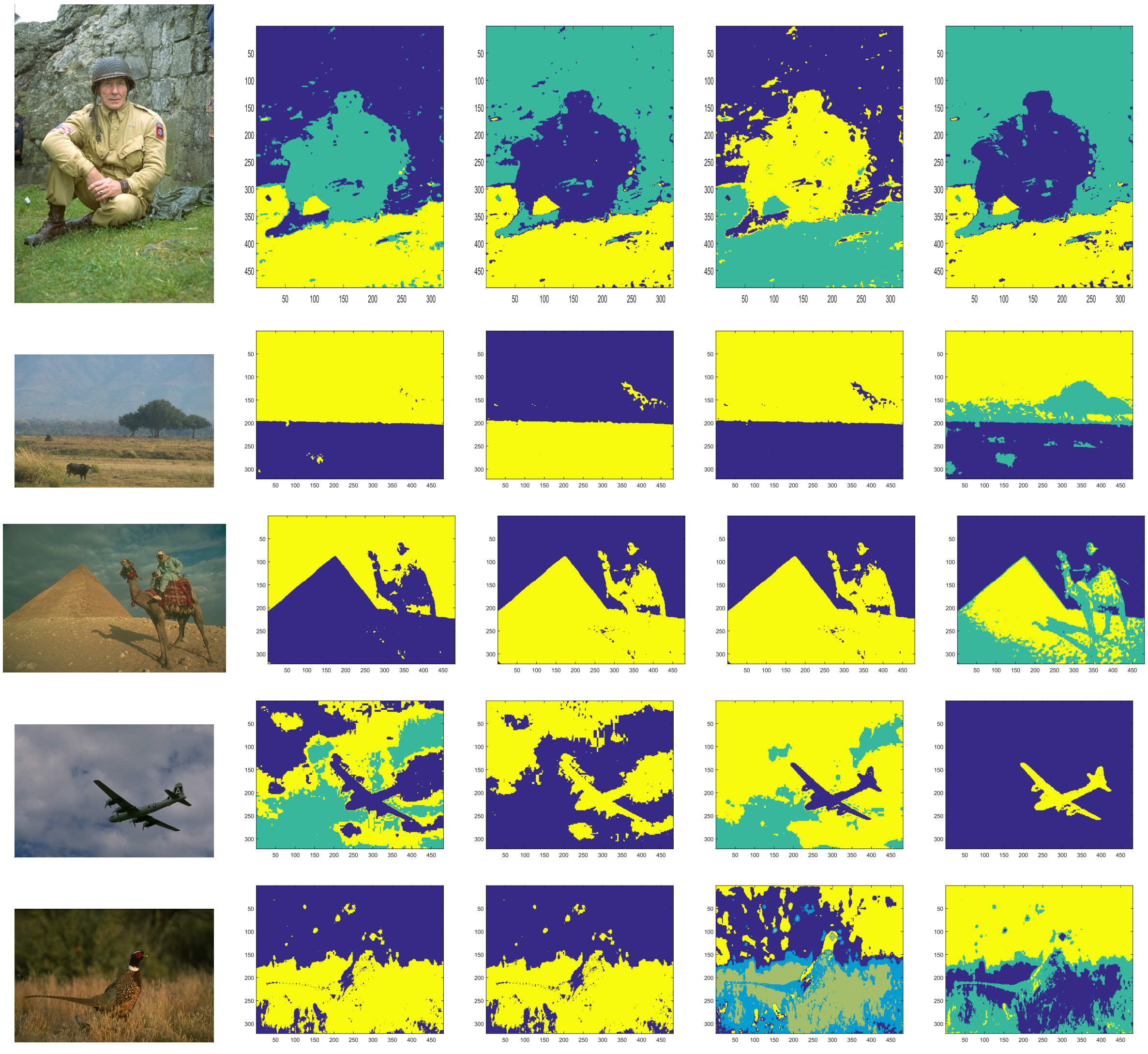


Figure 16 – Color segmentation results of different images (from top to bottom) with (a) Original images from Berkeley segmentation database (b) FCM method (c) FCM-ER (d) FCCI (e) Proposed method segmentations.



(a) Original (b) FCM (c) FCM-ER (d) FCCI (e) AFCM-ER

Figure 17 – Color segmentation results of different images (from top to bottom) with (a) Original images from Berkeley segmentation dataset (b) FCM method (c) FCM-ER (d) FCCI (e) Proposed method segmentations.

4.5 Chapter summary

The proposed method utility was evaluated through several experiments with synthetic and real datasets. The obtained results showed the superiority of the proposed algorithm when some variables are more relevant than others for the determination of the clusters. The algorithm was also tested for color image segmentation on the Berkeley segmentation dataset. The algorithm showed that obtained results correspond to human perception and maintains a good trade-off between the two segmentation evaluation measures and have a good match with human ground truth segmentation.

5 CONCLUSIONS

In this chapter conclusions of this fuzzy clustering algorithm for soft subspace clustering based on adaptive Euclidean distances and entropy regularization research are presented. This work was presented and published in the proceedings of 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). A list of contributions and limitations of this work are presented, as well as future work directions.

5.1 Contributions

It was proposed a new FCM-type algorithm for soft subspace clustering based on adaptive Euclidean distances, and the entropy of the objects as the regularization term in the objective function. An study of the state of the art of fuzzy clustering methods based on automatic variables selection and entropy regularization resulted as an identification of more suitable approach for partitioning objects or data samples into groups.

In comparison with conventional methods, the proposed approach has a better adaptive distances, which change with each algorithm iterations and are different from one cluster to another (local adaptive distances). This type of dissimilarity measure is adequate to learn the variables weights during the clustering process, leading to an improvement in the performance of the algorithms.

The algorithm starts from an initial fuzzy partition, then it alternates over three steps (i.e. representation, weighting, and allocation) until it converges as the adequacy criterion reaches a stationary value. At this point a new objective function is formulated and update rules are derived. The derivation of the expressions of the relevance weights was made considering that the product of the weight of the variables in each group should be equal to one.

At the end was obtained for proposed algorithm an expression for the best prototypes of the clusters, the best vectors of variable weights, the best rule of allocation of the objects to the clusters and the best diffuse partition matrix. Convergence properties of the proposed method were also demonstrated. The proposed algorithm gives as solution: the best centroid of each cluster, the best relevance weight of each variable, as well the best partition, according to the clustering criterion.

The parameter selection is a complicated process in clustering algorithms that result in significant performance improvement or deterioration. In most bibliography methods, parameters adjustment is made using objects labels, that is impracticable in real clustering applications. In this work was adapted the proposal presented by Schwämmle *et al.* (SCHWÄMMLE; JENSEN, 2010) to determine the parameter values in an unsupervised way, a novelty in this field.

The proposed method utility was demonstrated through various experiments with synthetic and real datasets (available at the *UCI* machine learning repository) and color image segmentation.

5.2 Summary of Results

Experiments on synthetic datasets shown that AFCM-ER algorithm presented the best average performance rank (according to the objective function) respect to the fuzzy partitions in almost all datasets compared with the other approaches. The worst performance was presented by the FCM algorithm. In terms of the means and standard deviations the proposed method manifested robustness, obtaining significantly better results over other compared works.

In the other hand for hard partitions, the AFCM-ER algorithm also performed better than the other approaches for most of the datasets (according to the objective function). The method showed that is good for soft subspace clustering problems. Even for different shape sets the performance of proposed algorithm stays constant because AFCM-ER algorithm emphasizes more on the importance of correct attributes for the clusters, obtaining higher clustering accuracy. The average clustering results, in terms of the means and standard deviations, indicate robustness in the performance in hard partitions. The worst result was obtained by FCCI algorithm.

Result in real datasets show that proposed and FCCI algorithms present the best average performance rank (according to the objective function) respect to the fuzzy partitions, obtaining good quality of clustering in real datasets. Moreover, FCM and FCM-ER algorithms achieved the third and fourth best average rankings respectively. The results expressed in terms of the means and standard deviations, show that the proposed algorithm usually demonstrates better performance than the other approaches in almost all considered indices, proving more robustness despite not having presented the best overall average performance ranking according to the objective function. The worst performance was presented by FCM-ER algorithm.

For the hard partitions, AFCM-ER performed better than the other algorithms for most of the datasets. Moreover, the FCM and FCCI algorithm achieved, respectively, the second and third best average rankings according to the average ranking. Finally, the proposed algorithm, in general, performed better with a higher frequency. In terms of the means and standard deviations the AFCM-ER algorithm significantly outperformed the other three algorithms in most datasets.

The algorithm was also experimented for color image segmentation on the Berkeley segmentation database (ARBELAEZ; FOWLKES; MARTIN, 2007). The obtained results by AFCM-ER algorithm showed that segmented images correspond to human perception. The segmentation maintains a good trade off between the two segmentation evaluation measures and have a good match with human ground truth segmentation. Respect to PR and F indexes, the AFCM-ER algorithm presented the best performance among all these algorithms. For PR index the worst result was obtained by FCM algorithm and FCM-ER algorithm for Liu's evaluation measure. Segmentation results by the proposed method form well defined and interpretable clusters even when the color difference between two regions is not too distinct.

In general, the proposed algorithm obtained the best clustering results for hard partitions compared with the literature proposals, indicating that automatic variable selection and entropy regularization is an effective way to enhance the performance of the clustering algorithms.

5.3 Limitations

The Euclidean distance is traditionally used to compare the objects and the prototypes in the FCM algorithms, but theoretical studies indicate that methods based on Euclidean distance are not indicated for samples with outlier and characteristics with outlier. The proposed method is a version of the FCM-type algorithm for soft subspace clustering based on adaptive Euclidean distances, and the entropy of the objects as the regularization term in the objective function. In conclusion, the AFCM-ER algorithm is less robust to the presence of outliers.

5.4 Future Work

The distance metric used in clustering plays a critical role in Machine Learning algorithms. Generally, fuzzy clustering algorithms use the Euclidean distance to reasonably describe relationships between objects. The Euclidean distance metric considers that all variables are equally important, in the sense that all have the same weight for group definition. Nevertheless, in most areas of knowledge, and especially if we are dealing with high-volume datasets, some variables may be irrelevant. Among those that are relevant, some may be of greater importance than others in building the groups. Also, the contribution of each variable to each group may be different, for example, each group may have a different set of important variables. In addition, the Euclidean distance metric is not indicated for samples with outlier and characteristics with outliers. An appropriate solution is to introduce a distance metric with good quality, robust to outliers, and to identify relevant variables. The provision of such distance metrics is highly problem-specific and determines the success or failure of a learning algorithm.

Without doubt, a good clustering is crucial for many real world applications. The semi-supervised clustering based on the partitions methods, from the use of class labels, must-link and can not-link constraints between objects (CHAPELLE; SCHOLKOPF; ZIEN, 2009) or degrees of a priori membership in some instances is used to assist unsupervised clustering. This results in a generally attractive approach, both in theory and practice, for requiring less human effort and greater accuracy. In semi-supervised clustering, pairs of objects that must belong to the same partition are indicated as a must-link, while objects belonging to different partitions are indicated as can not-link. Such constraints are often easy to extract from applications that have the availability of labeled historical data that could be used to guide the collating process when such labels are no longer available. An added value of these techniques is that not only is the quality of the resulting cluster improved, it provides a solution that is more consistent with the user's view of the domain, but also improves computational performance by removing low quality groups or only with a few elements.

Therefore it is necessary to develop a method that uses a robust distance metric to outliers, which incorporates automatic selection of variables in the objective function to take into account the importance of the variables and incorporate must-link and can not-link constraints to aid the clustering process without supervision, giving continuity to the research in fuzzy clustering algorithms with automatic selection of variables and entropy regularization materialized in the work published by (RODRÍGUEZ; CARVALHO, 2017).

REFERENCE

ACHLIOPTAS, D. Database-friendly random projections. In: *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, p. 274-281, 2001.

AGRAWAL, R.; GEHRKE, J.; GUNOPULOS, D.; RAGHAVAN, P. Automatic subspace clustering of high dimensional data for data mining applications. ACM, v. 27, n. 2, 1998.

AHMAD, A.; DEY, L. A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern Recognition Letters*, Elsevier, v. 32, n. 7, p. 1062-1069, 2011.

AMORIM, R. C. D.; MIRKIN, B. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, Elsevier, v. 45, n. 3, p. 1061-1075, 2012.

ARBELAEZ, P.; FOWLKES, C.; MARTIN, D. The Berkeley segmentation dataset and benchmark. *see https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/*, 2007.

ASUNCION, A.; NEWMAN, D. *UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA*, 1994.

BAEZA-YATES, R.; RIBEIRO-NETO, B. Modern information retrieval. *ACM press New York*, v. 463, 1999.

BAI, L.; LIANG, J.; DANG, C.; CAO, F. A novel attribute weighting algorithm for clustering high-dimensional categorical data. *Pattern Recognition*, Elsevier, v. 44, n. 12, p. 2843-2861, 2011.

BEZDEK, J. C. Mathematical models for systematics and taxonomy. In: *Proceedings of eighth international conference on numerical taxonomy*, v. 3, p. 143-166, 1975.

BEZDEK, J. C. Objective function clustering. In: *Pattern recognition with fuzzy objective function algorithms*. Springer, p. 43-93, 1981.

BEZDEK, J. C. Pattern recognition with fuzzy objective function algorithms. *Springer Science & Business Media*, 2013.

BINGHAM, E.; MANNILA, H. Random projection in dimensionality reduction: applications to image and text data. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, p. 245-250, 2001.

BISHOP, C. M. Neural networks for pattern recognition. In: *Oxford University Press*, 1995.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. *Artificial intelligence*, Elsevier, v. 97, n. 1, p. 245-271, 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.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. *Classification and regression trees*, CRC press, 1984.

CAMASTRA, F.; VERRI, A. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 27, n. 5, p. 801-805, 2005.

CAMPELLO, R. J. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, Elsevier, v. 28, n. 7, p. 833-841, 2007.

CAMPELLO, R. J.; HRUSCHKA, E. R. A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, Elsevier, v. 157, n. 21, p. 2858-2875, 2006.

CHACON, M. I.; AGUILAR, L.; DELGADO, A. Definition and applications of a fuzzy image processing scheme. In: *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, IEEE p. 102-107, 2002.

CHAN, E. Y.; CHING, W. K.; NG, M. K.; HUANG, J. Z. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, Elsevier, v. 37, n. 5, p. 943-952, 2004.

CHANG, H.; YEUNG, D. Y. Robust path-based spectral clustering. *Pattern Recognition*, Elsevier, v. 41, n. 1, p. 191-203, 2008.

CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542-542, 2009.

CHEN, L.; JIANG, Q.; WANG, S. A probability model for projective clustering on high dimensional data. In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, IEEE, p. 755-760, 2008.

CHEN, L.; JIANG, Q.; WANG, S. Model-based method for projective clustering. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 24, n. 7, p. 1291-1305, 2012.

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, X.; XU, X.; HUANG, J. Z.; YE, Y. Tw-k-means: Automated two-level variable weighting clustering algorithm for multiview data. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 25, n. 4, p. 932-944, 2013.

CHEN, X.; YE, Y.; XU, X.; HUANG, J. Z. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, Elsevier, v. 45, n. 1, p. 434-446, 2012.

CHEUNG, Y. M.; ZENG, H. A maximum weighted likelihood approach to simultaneous model selection and feature weighting in gaussian mixture. *Artificial Neural Networks–ICANN 2007*, Springer, p. 78-87, 2007.

COX, E. *Fuzzy modeling and genetic algorithms for data mining and exploration*. Elsevier, 2005.

DASH, M.; CHOI, K.; SCHEUERMANN, P.; LIU, H. Feature selection for clustering-a filter solution. In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, IEEE p. 115-122, 2002.

DASH, M.; LIU, H. Feature selection for clustering. In: *Pacific-Asia Conference on knowledge discovery and data mining*, SPRINGER, p. 110-121, 2000.

DASH, M.; LIU, H.; YAO, J. Dimensionality reduction of unsupervised data. In: *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*, IEEE, p. 532-539, 1997.

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.

DESARBO, W. S.; CARROLL, J. D.; CLARK, L. A.; GREEN, P. E. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *Psychometrika*, Springer, v. 49, n. 1, p. 57-78, 1984.

DEVANEY, M.; RAM, A. Efficient feature selection in conceptual clustering. In: *ICML*, v. 97, p. 92-97, 1997.

DIDAY, E. Classification automatique avec distances adaptatives. *RAIRO Informatique Computer Science*, v. 11, n. 4, p. 329-349, 1977.

DIDAY, E.; GOVAERT, G. Automatic classification with adaptive intervals. *RAIRO Inf Comput Sci*, v. 11, n. 4, p. 329-349, 1977.

DING, C.; HE, X.; ZHA, H.; SIMON, H. D. Adaptive dimension reduction for clustering high dimensional data. In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, IEEE, p. 147-154, 2002.

DOM, B. E. An information-theoretic external cluster-validity measure. In: *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, MORGAN KAUFMANN PUBLISHERS INC., p. 137-145, 2002.

DOMENICONI, C.; AL-RAZGAN, M. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM, v. 2, n. 4, p. 17, 2009.

DOMENICONI, C.; GUNOPULOS, D.; MA, S.; YAN, B.; AL-RAZGAN, M.; PAPADOPOULOS, D. Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery*, Springer, v. 14, n. 1, p. 63-97, 2007.

DOMENICONI, C.; PAPADOPOULOS, D.; GUNOPULOS, D.; MA, S. Subspace clustering of high dimensional data. In: *Proceedings of the 2004 SIAM international conference on data mining*, SIAM, p. 517-521, 2004.

DUDA, R. O.; HART, P. E. Pattern classification and scene analysis. Wiley, 1973.

DUNN, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Taylor & Francis, 1973.

DY, J. G.; BRODLEY, C. E. Feature subset selection and order identification for unsupervised learning. In: *ICML*, p. 247-254, 2000.

DY, J. G.; BRODLEY, C. E. Feature selection for unsupervised learning. *Journal of machine learning research*, v. 5, n. Aug, p. 845-889, 2004.

ESTER, M.; KRIEGEL, H. P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*, v. 96, n. 34, p. 226-231, 1996.

EVERITT, B.; LANDAU, S.; LEESE, M.; STAHL, D. Cluster analysis. *Arnold, London*, 2001.

FERN, X. Z.; BRODLEY, C. E. Random projection for high dimensional data clustering: A cluster ensemble approach. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, p. 186-193, 2003.

FERREIRA, M. R.; De Carvalho, Francisco De A. T. Kernel fuzzy C-means with automatic variable weighting. *Fuzzy Sets and Systems*, Elsevier, v. 237, p. 1-46, 2014.

FISHER, D. H. Improving inference through conceptual clustering. In: *AAAI*, v. 87, p. 461-465, 1987.

FRIEDMAN, J. H.; MEULMAN, J. J. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Wiley Online Library, v. 66, n. 4, p. 815-849, 2004.

FRIGUI, H.; NASRAOUI, O. Simultaneous clustering and dynamic keyword weighting for text documents. In: *Survey of text mining*, Springer, p. 45-72, 2004.

FRIGUI, H.; NASRAOUI, O. Unsupervised learning of prototypes and attribute weights. *Pattern recognition*, Elsevier, v. 37, n. 3, p. 567-581, 2004.

FRITZ, H.; GARCÍA-ESCUDERO, L. A.; MAYO-ISCAR, A. Robust constrained fuzzy clustering. *Information Sciences*, Elsevier, v. 245, p. 38-52, 2013.

FU, L.; MEDICO, E. Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC Bioinformatics*, BioMed Central, v. 8, n. 1, p. 3, 2007.

GAN, G.; NG, M. K. P. Subspace clustering with automatic feature grouping. *Pattern Recognition*, Elsevier, v. 48, n. 11, p. 3703-3713, 2015.

GAN, G.; WU, J. A convergence theorem for the fuzzy subspace clustering (fsc) algorithm. *Pattern Recognition*, Elsevier, v. 41, n. 6, p. 1939-1947, 2008.

GAN, G.; WU, J.; YANG, Z. A fuzzy subspace algorithm for clustering high dimensional data. In: *ADMA*, SPRINGER, p. 271-278, 2006.

GENNARI, J. H.; LANGLEY, P.; FISHER, D. Models of incremental concept formation. *Artificial intelligence*, Elsevier, v. 40, n. 1-3, p. 11-61, 1989.

GIONIS, A.; MANNILA, H.; TSAPARAS, P. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM, v. 1, n. 1, p. 4, 2007.

GORUNESCU, F. Data Mining: Concepts, models and techniques. *Springer Science & Business Media*, v. 12, 2011.

GULLO, F.; DOMENICONI, C.; TAGARELLI, A. Projective clustering ensembles. *Data Mining and Knowledge Discovery*, Springer, v. 26, n. 3, p. 452-511, 2013.

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.

HANSEN, P.; JAUMARD, B. Cluster analysis and mathematical programming. *Mathematical programming*, Springer, v. 79, n. 1-3, p. 191-215, 1997.

HARTIGAN, J. A.; HARTIGAN, J. Clustering algorithms. *Wiley New York*, v. 209, 1975.

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.

HINNEBURG, A.; KEIM, D. A. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. 1999.

HÖPPNER, F. Fuzzy cluster analysis: methods for classification, data analysis and image recognition. *John Wiley & Sons*, 1999.

HORE, P.; HALL, L. O.; GOLDGOF, D. B. Single pass fuzzy c means. In: *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, IEEE, p. 1-7, 2007.

HRUSCHKA, E. R.; CAMPELLO, R. J.; CASTRO, L. N. D. Evolving clusters in gene-expression data. *Information Sciences*, Elsevier, v. 176, n. 13, p. 1898-1927, 2006.

HRUSCHKA, E. R.; CASTRO, L. N. de; CAMPELLO, R. J. Evolutionary algorithms for clustering gene-expression data. In: *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, IEEE, p. 403-406, 2004.

HUANG, H. C.; CHUANG, Y. Y.; CHEN, C. S. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 20, n. 1, p. 120-134, 2012.

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. A fuzzy variant of the rand index for comparing clustering structures. In: *Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, IFSA-EUSFLAT 2009*, IFSA-EUSFLAT, 2009.

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.

JAYNES, E. T. Information theory and statistical mechanics. *Physical review*, APS, v. 106, n. 4, p. 620, 1957.

JIANG, Y.; CHUNG, F. L.; WANG, S.; DENG, Z.; WANG, J.; QIAN, P. Collaborative fuzzy clustering from multiple weighted views. *IEEE transactions on cybernetics*, IEEE, v. 45, n. 4, p. 688-701, 2015.

JING, L.; NG, M. K.; HUANG, J. Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 19, n. 8, 2007.

JING, L.; NG, M. K.; XU, J.; HUANG, J. Z. Subspace clustering of text documents with feature weighting k-means algorithm. In: *PAKDD*, SPRINGER, p. 802-812, 2005.

KARAYIANNIS, N. B.; PAI, P. I. Fuzzy vector quantization algorithms and their application in image compression. *IEEE Transactions on Image Processing*, IEEE, v. 4, n. 9, p. 1193-1201, 1995.

KAUFMAN, L.; ROUSSEEUW, P. Clustering by means of medoids. *North-Holland*, 1987.

KAUFMAN, L.; ROUSSEEUW, P. J. Finding groups in data: an introduction to cluster analysis. *John Wiley & Sons*, v. 344, 2009.

KELLER, A.; KLAWONN, F. Fuzzy clustering with weighting of data variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, World Scientific, v. 8, n. 06, p. 735-746, 2000.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 273-324, 1997.

KOLEN, J. F.; HUTCHESON, T. Reducing the time complexity of the fuzzy c-means algorithm. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 10, n. 2, p. 263-267, 2002.

KONONENKO, I.; KUKAR, M. Machine learning and data mining: introduction to principles and algorithms. *Horwood Publishing*, 2007.

KRIEGEL, H. P.; KRÖGER, P.; ZIMEK, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM, v. 3, n. 1, p. 1, 2009.

LI, J.; GAO, X.; JIAO, L. A new feature weighted fuzzy clustering algorithm. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer, p. 412-420, 2005.

LI, R. P.; SADAAKI, M. A maximum-entropy approach to fuzzy clustering. In: *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, IEEE, p. 2227-2232, 1995.

LI, T.; CHEN, Y. An improved k-means algorithm for clustering using entropy weighting measures. In: *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, IEEE, p. 149-153, 2008.

LIPOVETSKY, S. Additive and multiplicative mixed normal distributions and finding cluster centers. *International Journal of Machine Learning and Cybernetics*, Springer, v. 4, n. 1, p. 1-11, 2013.

LIU, H.; MOTODA, H. Feature selection for knowledge discovery and data mining. *Springer Science & Business Media*, v. 454, 2012.

LIU, J.; YANG, Y. H. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 16, n. 7, p. 689-700, 1994.

LIU, Y.; WU, S.; LIU, Z.; CHAO, H. A fuzzy co-clustering algorithm for biomedical data. *PloS one*, Public Library of Science, v. 12, n. 4, p. e0176536, 2017.

LU, Y.; MA, T.; YIN, C.; XIE, X.; TIAN, W.; ZHONG, S. Implementation of the fuzzy C-means clustering algorithm in meteorological data. *International Journal of Database Theory and Application*, v. 6, n. 6, p. 1-18, 2013.

LU, Y.; WANG, S.; LI, S.; ZHOU, C. Particle swarm optimizer for variable weighting in clustering high-dimensional data. *Machine learning*, Springer, v. 82, n. 1, p. 43-70, 2011.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, OAKLAND, CA, USA, v. 1, n. 14, p. 281-297, 1967.

MAKARENKOV, V.; LEGENDRE, P. Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. *Journal of Classification*, Springer, v. 18, n. 2, p. 245-271, 2001.

MINING, W. I. D. Data mining: Concepts and techniques. *Morgan Kaufinann*, 2006.

MITRA, P.; MURTHY, C.; PAL, S. K. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 3, p. 301-312, 2002.

MITRA, S.; ACHARYA, T. *Data mining: multimedia, soft computing, and bioinformatics*. John Wiley & Sons, 2005.

MIYAGISHI, K.; YASUTOMI, Y.; ICHIHASHI, H.; HONDA, K. Fuzzy clustering with regularization by kl information. In: *16th Fuzzy System Symposium*, p. 549-550, 2000.

MODHA, D. S.; SPANGLER, W. S. Feature weighting in k-means clustering. *Machine learning*, Springer, v. 52, n. 3, p. 217-237, 2003.

MURTAGH, F. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, The British Computer Society, v. 26, n. 4, p. 354-359, 1983.

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

PARVIN, H.; MINAEI-BIDGOLI, B. A clustering ensemble framework based on elite selection of weighted clusters. *Advances in Data Analysis and Classification*, Springer, v. 7, n. 2, p. 181-208, 2013.

PENA, J. M.; LOZANO, J. A.; LARRAÑAGA, P.; INZA, I. Dimensionality reduction in unsupervised learning of conditional gaussian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 23, n. 6, p. 590-603, 2001.

PENG, L.; ZHANG, J. An entropy weighting mixture model for subspace clustering of high-dimensional data. *Pattern Recognition Letters*, Elsevier, v. 32, n. 8, p. 1154-1161, 2011.

RANA, S.; JASOLA, S.; KUMAR, R. A boundary restricted adaptive particle swarm optimization for data clustering. *International journal of machine learning and cybernetics*, Springer, v. 4, n. 4, p. 391-400, 2013.

RAND, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, Taylor & Francis Group, v. 66, n. 336, p. 846-850, 1971.

ROBERTS, J. From know-how to show-how? questioning the role of information and communication technologies in knowledge transfer. *Technology Analysis & Strategic Management*, Taylor & Francis, v. 12, n. 4, p. 429-443, 2000.

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

ROSE, K.; GUREWITZ, E.; FOX, G. C. Statistical mechanics and phase transitions in clustering. *Physical review letters*, APS, v. 65, n. 8, p. 945, 1990.

SADAAKI, M.; MASAO, M. Fuzzy c-means as a regularization and maximum entropy approach. In: *IFSA'97 Prague : proceedings of the seventh International Fuzzy Systems Association World Congress*, p. 86-92, 1997.

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

SHANNON, C. E. A mathematical theory of communication, part i, part ii. *Bell Syst. Tech. J.*, v. 27, p. 623-656, 1948.

SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, v. 5, n. 1, p. 3-55, 2001.

SHEN, H.; YANG, J.; WANG, S.; LIU, X. Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, Springer, v. 10, n. 11, p. 1061-1073, 2006.

SIM, K.; GOPALKRISHNAN, V.; ZIMEK, A.; CONG, G. A survey on enhanced subspace clustering. *Data mining and knowledge discovery*, Springer, v. 26, n. 2, p. 332-397, 2013.

SOETE, G. D. Optimal variable weighting for ultrametric and additive tree clustering. *Quality and Quantity*, Springer, v. 20, n. 2-3, p. 169-180, 1986.

SOETE, G. D.; CARROLL, J. D. K-means clustering in a low-dimensional euclidean space. In: *New approaches in classification and data analysis*, Springer, p. 212-219, 1994.

SREENIVASARAO, V.; VIDYAVATHI, D. S. Comparative analysis of fuzzy C-mean and modified fuzzy possibilistic c-mean algorithms in data mining. *IJCST*, v. 1, n. 1, p. 104-106, 2010.

THOMAS, B.; NASHIPUDIMATH, M. Comparative analysis of fuzzy clustering algorithms in data mining. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, v. 1, n. 7, p. pp-221, 2012.

TIMMERMAN, M. E.; CEULEMANS, E.; KIERS, H. A.; VICHI, M. Factorial and reduced k-means reconsidered. *Computational Statistics & Data Analysis*, Elsevier, v. 54, n. 7, p. 1858-1871, 2010.

TSAI, C.; CHIU, C. Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. *Computational Statistics and Data Analysis*, v. 52, p. 4658-4672, 2008.

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.

UNNIKRISHNAN, R.; HEBERT, M. Measures of similarity. In: *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, IEEE, v. 1, p. 394-394, 2005.

WANG, J.; CHUNG, F. l.; WANG, S.; DENG, Z. Double indices-induced fcm clustering and its integration with fuzzy subspace clustering. *Pattern analysis and applications*, Springer, v. 17, n. 3, p. 549-566, 2014.

WANG, J.; DENG, Z.; JIANG, Y.; QIAN, P.; WANG, S. Multiple-kernel based soft subspace fuzzy clustering. In: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, IEEE, p. 186-193, 2014.

WANG, P. *Pattern-recognition with fuzzy objective function algorithms-Bezdek*, JC, SIAM PUBLICATIONS 3600 UNIV CITY SCIENCE CENTER, PHILADELPHIA, p. 19104-2688, 1983.

WEBB, A. R. Statistical Pattern Recognition. *John Wiley & Sons*, 2003.

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, 2012.

XIE, X. L.; BENI, G. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, v. 13, n. 8, p. 841-847, 1991.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, IEEE, v. 16, n. 3, p. 645-678, 2005.

YEUNG, K. Y.; HAYNOR, D. R.; RUZZO, W. L. Validating clustering for gene expression data. *Bioinformatics*, Oxford University Press, v. 17, n. 4, p. 309-318, 2001.

ZADEH, L. A. Fuzzy sets. *Information and control*, Elsevier, v. 8, n. 3, p. 338-353, 1965.

ZAHN, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, IEEE, v. 100, n. 1, p. 68-86, 1971.

ZAÏT, M.; MESSATFA, H. A comparative study of clustering methods. *Future Generation Computer Systems*, Elsevier, v. 13, n. 2-3, p. 149-159, 1997.

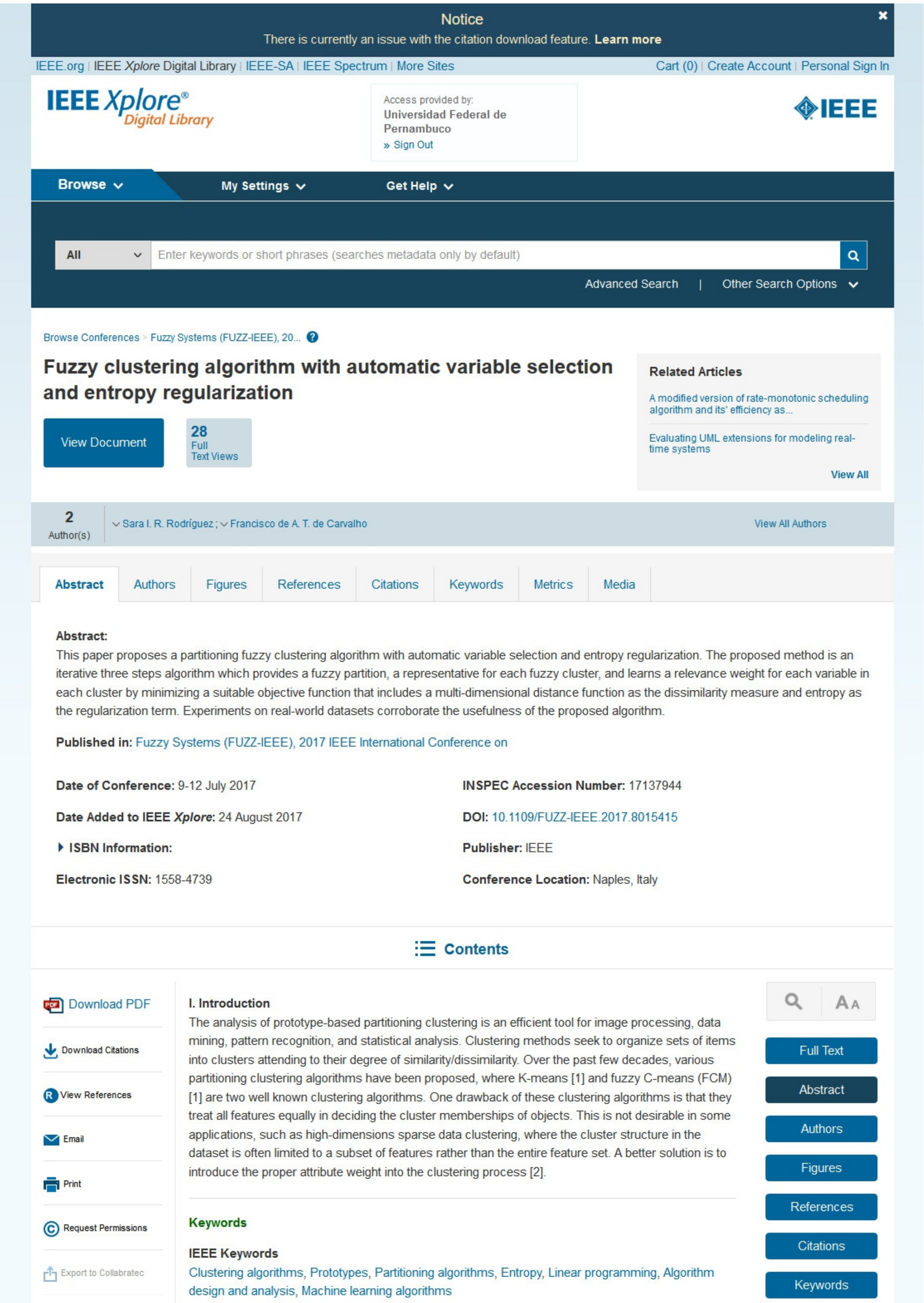
ZHOU, J.; CHEN, L.; CHEN, C. P.; ZHANG, Y.; LI, H. X. Fuzzy clustering with the entropy of attribute weights. *Neurocomputing*, Elsevier, v. 198, p. 125-134, 2016.

ZHU, L.; CAO, L.; YANG, J. Multiobjective evolutionary algorithm-based soft subspace clustering. In: IEEE. *Evolutionary Computation (CEC), 2012 IEEE Congress on*, IEEE, p. 1-8, 2012.

Appendix

APPENDIX A – PAPER FUZZ-IEEE 2017

This appendix contains the FUZZ-IEEE page where the article (RODRÍGUEZ; CARVALHO, 2017) was published and the first page of the title article: *Fuzzy clustering algorithm with automatic variable selection and entropy regularization* accepted and published in 2017 IEEE International Conference on Fuzzy Systems.



Notice

There is currently an issue with the citation download feature. **Learn more**

×
IEEE.org | IEEE *Xplore* Digital Library | IEEE-SA | IEEE Spectrum | More Sites
Cart (0) | Create Account | Personal Sign In
IEEE *Xplore*® Digital Library
IEEE
Browse
My Settings
Get Help
All
Advanced Search
Other Search Options

Access provided by:
Universidad Federal de Pernambuco
» Sign Out

Enter keywords or short phrases (searches metadata only by default)

Browse Conferences > Fuzzy Systems (FUZZ-IEEE), 20...

Fuzzy clustering algorithm with automatic variable selection and entropy regularization

28 Full Text Views

Related Articles

A modified version of rate-monotonic scheduling algorithm and its' efficiency as...

Evaluating UML extensions for modeling real-time systems

2 Author(s)

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

Abstract:
This paper proposes a partitioning fuzzy clustering algorithm with automatic variable selection and entropy regularization. The proposed method is an iterative three steps algorithm which provides a fuzzy partition, a representative for each fuzzy cluster, and learns a relevance weight for each variable in each cluster by minimizing a suitable objective function that includes a multi-dimensional distance function as the dissimilarity measure and entropy as the regularization term. Experiments on real-world datasets corroborate the usefulness of the proposed algorithm.

Published in: Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on

Date of Conference: 9-12 July 2017

INSPEC Accession Number: 17137944

Date Added to IEEE *Xplore*: 24 August 2017

DOI: 10.1109/FUZZ-IEEE.2017.8015415

ISBN Information:

Publisher: IEEE

Electronic ISSN: 1558-4739

Conference Location: Naples, Italy

Contents

I. Introduction

The analysis of prototype-based partitioning clustering is an efficient tool for image processing, data mining, pattern recognition, and statistical analysis. Clustering methods seek to organize sets of items into clusters attending to their degree of similarity/dissimilarity. Over the past few decades, various partitioning clustering algorithms have been proposed, where K-means [1] and fuzzy C-means (FCM) [1] are two well known clustering algorithms. One drawback of these clustering algorithms is that they treat all features equally in deciding the cluster memberships of objects. This is not desirable in some applications, such as high-dimensions sparse data clustering, where the cluster structure in the dataset is often limited to a subset of features rather than the entire feature set. A better solution is to introduce the proper attribute weight into the clustering process [2].

Keywords

IEEE Keywords

Clustering algorithms, Prototypes, Partitioning algorithms, Entropy, Linear programming, Algorithm design and analysis, Machine learning algorithms

Fuzzy clustering Algorithm with Automatic Variable Selection and Entropy Regularization

Sara I.R. Rodríguez
Centro de Informatica - CIn/UFPE
Av. Prof. Luiz Freire, s/n - Cidade Universitaria
CEP: 50740-540, Recife-PE, Brazil
Email: sirr@cin.ufpe.br

Francisco de A.T. de Carvalho
Centro de Informatica - CIn/UFPE
Av. Prof. Luiz Freire, s/n - Cidade Universitaria
CEP: 50740-540, Recife-PE, Brazil
Email: fatc@cin.ufpe.br

Abstract—This paper proposes a partitioning fuzzy clustering algorithm with automatic variable selection and entropy regularization. The proposed method is an iterative three steps algorithm which provides a fuzzy partition, a representative for each fuzzy cluster, and learns a relevance weight for each variable in each cluster by minimizing a suitable objective function that includes a multi-dimensional distance function as the dissimilarity measure and entropy as the regularization term. Experiments on real-world datasets corroborate the usefulness of the proposed algorithm.

I. Introduction

The analysis of prototype-based partitioning clustering is an efficient tool for image processing, data mining, pattern recognition, and statistical analysis. Clustering methods seek to organize sets of items into clusters attending to their degree of similarity/dissimilarity. Over the past few decades, various partitioning clustering algorithms have been proposed, where K-means [1] and fuzzy C-means (FCM) [1] are two well known clustering algorithms. One drawback of these clustering algorithms is that they treat all features equally in deciding the cluster memberships of objects. This is not desirable in some applications, such as high-dimensions sparse data clustering, where the cluster structure in the dataset is often limited to a subset of features rather than the entire feature set. A better solution is to introduce the proper attribute weight into the clustering process [2].

Partitioning methods can be divided into hard and fuzzy clustering. Hard clustering provides a hard partition in which each object of the dataset is assigned to one and only one cluster. Fuzzy clustering generates a fuzzy partition which assigns a degree of membership to each pattern in a given cluster. The degree of membership is important for discovering intricate relations which may arise between a given data object and all clusters [3]. The fuzzy clustering method is peculiarly effective when the boundaries between clusters of data are ambiguous.

Several attribute-weighted fuzzy clustering methods have been proposed in the past few decades. Ref. [4] introduced a basic attribute-weighted FCM algorithm by assigning one influence parameter to each single data dimension for each cluster, while Ref. [5] put forward an approach searching for the optimal prototype parameters and the optimal feature weights simultaneously. The algorithm proposed in Ref. [6] present an enhanced soft subspace clustering (ESSC) algorithm by employing both within-cluster and between-cluster information.

Reference [7] proposed a version of FCM algorithm with entropy regularization. Despite the usefulness of this algorithm (hereafter named FCM-ER) in comparison with the conventional FCM, it still assumes that the variables have the same importance for the clustering task. Later, Ref. [8] proposed a FCM with entropy regularization and automatic selection of variables.

This paper presents a new FCM-type algorithm for soft subspace clustering based on adaptive Euclidean distances, and the entropy of the objects as the regularization term in the objective function. The proposed is an iterative three-steps relocation algorithm which determines a fuzzy partition, a vector of representatives (prototypes) for the fuzzy clusters, and learns a relevance weight of each variable in each fuzzy cluster by optimizing an adequacy criterion that measures the fitting between the fuzzy clusters and their representatives. The relevance weights change at each iteration of the algorithm, and differ from one fuzzy cluster to another. An advantage of such algorithm is that it requires the tuning of less parameters than does the Fuzzy Co-Clustering algorithm for Images (hereafter named FCCI) of Ref. [8].

The paper is organized as follows. Section II reviews two works closely related with the proposed approach. The fuzzy clustering algorithm with automatic variable selection and entropy regularization is presented in Section III. Section IV provides several experiments with some UCI machine learning repository datasets that corroborate the usefulness of the proposed algorithm. Finally, conclusions are drawn in Section V.

II. Related work

Several maximum entropy clustering algorithms and their variants are available in the literature which aims to search for global regularity and obtain the smoothest reconstructions from the available data. In this section, we briefly describe two algorithms closely related to our approach.

Let $E = \{e_1, \ldots, e_N\}$ be a set of N objects. Each object e_i $(1 \le i \le N)$ is described by the vector $x_i = (x_{i1}, \ldots, x_{iP})$,

978-1-5090-6034-4/17/$31.00 c 2017 IEEE