Contents lists available at ScienceDirect

Neurocomputing

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

Multivariate Fuzzy C-Means algorithms with weighting

Bruno Almeida Pimentel, Renata M.C.R. de Souza*

Universidade Federal de Pernambuco (UFPE), Centro de Informática (CIn), Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária 50, 740-560 Recife - PE, Brazil

ARTICLE INFO

Article history:
Received 9 December 2014
Received in revised form
21 August 2015
Accepted 5 October 2015
Communicated by M. Sato-Ilic

Keywords:
Fuzzy C-Means
Multivariate membership
Weighted variables
Interpretation indices

ABSTRACT

With the growing interest in automatic understanding, processing and summarization of data, several application domains, such as pattern recognition, machine learning and computational biology, have been making use of clustering algorithms. In the fuzzy clustering approach, Fuzzy C-Means (FCM) is the best-known method. Although FCM has performed well in cluster detection, membership values for each element assigned to each of the clusters cannot indicate how well the individuals are clustered in relation to each variable. To deal with this problem, a multivariate version of fuzzy c-means algorithm has been proposed. This proposition does not consider that there is a different relevant weight associated with each variable and that it may also be different from one cluster to another. Here, we propose two multivariate fuzzy c-means algorithms with weighting. Weights aim to represent how important each different variable is for each cluster and to improve the clustering quality. Furthermore, we propose tools based on suitable dispersion measures for interpretation of the fuzzy partition and fuzzy clusters obtained by multivariate fuzzy c-means methods. These tools allow the measurement of the overall quality, homogeneity of clusters and the role of different variables in the cluster formation process. To evaluate the performance of the proposed algorithms against other methods established by the clustering literature, experiments are performed with synthetic and UCI repository data sets, showing the usefulness of the algorithms with weighting.

1. Introduction

Machine learning is an important branch of artificial intelligence whose main goal is to study and propose methods able to learn from data [1,2]. In many problems, machine learning may be applied where the process of extracting information from data is complex [3]. Based on the type of input available during the training phase, machine learning algorithms can be supervised or unsupervised [4,5]. In the former, algorithms are trained on labeled data, whereas in the latter they are trained on unlabeled data. For several problems, it is not feasible to obtain data labels [6,7], therefore approaches for unsupervised learning like clustering are more suitable to solve these problems. The main goal of clustering algorithms is to form groups in a set of objects such that the intra-group similarity is greater than the similarity between groups. Clustering methods aim to optimize a suitable mathematical objective function and usually converge to a locally optimal partition [8,9]. With the growing interest in automatic understanding, processing and summarizing data, several application domains such as pattern recognition, machine learning, data mining, computer vision and computational biology have used clustering algorithms [10–13].

According to the way clustering algorithms find groups, they may be classified into: hierarchical and partitional [11,14]. In the former category, algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters. If the criterion is merging, then the hierarchical algorithm is agglomerative, starting with each pattern in a singleton cluster and successively merging clusters based on similarities. If the criterion is splitting, then the hierarchical algorithm is divisive and begins with all patterns in a single cluster, after which splits are executed until singleton clusters are found. The distance between two clusters may be measured using a linkage criterion that can change the way hierarchical algorithms find clusters [15]. Linkage criteria use pairwise similarity between objects, which make hierarchical algorithms limited in practice [16]. On the other hand, in the latter category of clustering algorithms, partitional ones obtain a single partition of the data instead of a nested series of partitions. The chosen number of desired clusters may be a problem since there is no always a priori information about data [17]. There are several works about how to automatically find the number of clusters [18,19]. However, partitional methods have the advantage of large data sets, since the time complexity is less than that of hierarchical methods [11,12].

* Corresponding author. Tel.: +55 8121268430; fax: +55 8121268438.
E-mail addresses: bap@cin.ufpe.br (B.A. Pimentel), rmcrs@cin.ufpe.br (R.M.C.R. de Souza).

http://dx.doi.org/10.1016/j.neucom.2015.10.011
0925-2312/© 2015 Elsevier B.V. All rights reserved.

Partitional methods may be classified into two main categories: hard and soft [11]. In hard partitional methods, objects belong to exactly one cluster, where the degree of belonging of an object to a cluster is 1 if the object belongs to cluster or 0 otherwise [20]. One of the most popular hard partitional methods is K-Means [21]. The term K-Means was first used by [22] and is one of the simplest and most widely used clustering methods [21]. Several extensions of K-Means method have been proposed [23]. In soft partitional methods, each object has a degree of membership value in the interval [0,1] for each cluster. Soft methods may be fuzzy, probabilistic or possibilistic [20]. The Fuzzy C-Means (FCM) clustering method introduced by [24] is a well-known and powerful method in the fuzzy approach [25]. The main representative of possibilistic approach is the Possibilistic C-Means (PCM) clustering method proposed by [26]. An advantage of FCM is that it is more suitable for overlapping clusters than hard methods [27].

Regarding the fuzzy approach, many extensions of the FCM method have been proposed in the literature. Gustafson and Kessel [28] extended the FCM by employing an adaptive distance norm. Zhang and Chen [29] proposed a new version of fuzzy c-means and possibilistic c-means substituting the original Euclidian distance with a kernel-induced one producing more significant membership degrees. Pal et al. [30] presented a new model called fuzzy-possibilistic c-means (FPCM), which mixes properties of FCM and PCM methods. Pal et al. [20] proposed a hybridization of PCM and FCM (abbreviated by PFCM) that avoids various problems of PCM, FCM and FPCM. Keller and Klawonn [31] introduced an objective function that assigns one influence parameter to each single data variable for each cluster. de Carvalho et al. [32] presented fuzzy clustering methods based on adaptive quadratic distances that can either be the same for all clusters or different from one cluster to another. Ferreira and de Carvalho [33] introduced variable-wise kernel fuzzy c-means clustering methods that allow the use of adaptive distances to improve the clustering quality. Although these methods can produce better results than traditional FCM, they do not consider that different variables may produce different membership degrees. Therefore, Pimentel and Souza [34] proposed a multivariate fuzzy c-means (MFCM) method where membership degrees of both clusters and variables are computed. From the multivariate approach, three important contributions can be identified: (1) the ability to interpret the relevance of each object for a given group according to each variable; (2) the ability to obtain more information from data, which leads methods to achieve higher clustering quality; (3) a tool for multivariate data analysis.

Although the multivariate model is useful for problems in which the variances for different features are not similar, the importance of each variable must be measured in clustering. Small weights reduce or eliminate the effect of insignificant variables. In addition, weights can be used in variable selection for data mining applications where large and complex data are often involved. But these weights need to be determined suitably with goodness-of-fit criteria in cluster analysis, since improper weights may result in biased treatment of different variables. Here, we propose multivariate clustering methods that weight the variables by minimizing a criterion function based on prototypes. Therefore, the main contributions of this paper are

- To introduce multivariate fuzzy c-means methods weighting variables. Two ways to obtain the weights are considered, and they optimize the goodness-of-fit criterion between clusters and their representatives. The first method is based on [35], where variable weights are obtained by modeling membership degrees for each point. In the second method, weights are obtained by modeling dispersions using the concept of adaptive distance [32,36,37]: they change at each iteration of the algorithm and they are different from one variable to another and from one cluster to another.
- To propose various fuzzy partition and cluster interpretation tools for multivariate fuzzy-c means and its proposed weighted methods. Indices' interpretations are useful tools that allow a better understanding of the resulting clustering, and are open topic concerning multivariate fuzzy clustering methods.
- To present an experimental evaluation with several real and synthetic data sets in order to demonstrate the usefulness of the algorithms and the merit of the proposed partition and cluster interpretation tools. For synthetic data sets, two different groups of configurations were set up: in the first one, partitions contain classes with elliptical or spherical shape and similar or different sizes; in the second one, partitions contain a mixture of elliptical and spherical classes.

The remainder of this paper is organized as follows: Section 2 introduces a brief review of the multivariate fuzzy c-means method. Section 3 shows multivariate fuzzy c-means methods using weighted variables. In Section 4, we propose tools based on suitable dispersion measures for the interpretation of the fuzzy partitions and fuzzy clusters. From these indices, it is possible to evaluate the overall quality, homogeneity of clusters and the role of different variables in the process of cluster formation. Experiments with synthetic and UCI machine learning repository data sets are shown in Section 5 and a comparative study of the proposed algorithms with the Gustafson–Kessel method [28], the traditional FCM [24], versions of FCM with weighting [31,32] and the MFCM [34] is performed, in Section 5. Moreover, this section carries out an analysis of interpretation indices for a benchmark data set. Finally, concluding remarks are given in Section 6.

2. Multivariate Fuzzy C-Means method: a review

Although the Fuzzy C-Means method has shown good performance in detecting clusters, the membership values for each individual computed to each cluster cannot indicate how well the individuals are classified concerning single variables. Therefore, a multivariate approach for membership degree was proposed by [34]. This method assumes that variables are not correlated and the data is linearly separable as well as their weighted versions. In this section, a brief description of the Multivariate Fuzzy C-Means (here called MFCM) method is presented.

Let $\Omega = \{1, \ldots, k, \ldots, n\}$ be a set of n objects indexed by k. Each object k is represented by a quantitative variable vector $\mathbf{x}_k = (x_{1k}, \ldots, x_{jk}, \ldots, x_{pk})^T$ described by p variables indexed by j where $x_{jk} \in \Re$.

Let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_c\}$ be a set of c prototypes related to c clusters of the partition, where each prototype of a cluster C_i $(i = 1, \ldots, c)$ is also represented as a quantitative variable vector $\mathbf{y}_i = (y_{i1}, \ldots, y_{ij}, \ldots, y_{ip})^T$, where $y_{ij} \in \Re$.

Consider $\mathbf{U} = [\mathbf{u}_k]$ a matrix of multivariate membership matrices, where for each object k from Ω there is a $c \times p$ multivariate membership matrix $\mathbf{u}_k = [u_{ijk}]$ $(k = 1, \ldots, n)$ where u_{ijk} is the membership degree of the object k to the cluster C_i $(i = 1, \ldots, c)$ on the variable j $(j = 1, \ldots, p)$.

And let $\Delta = [\delta_{ik}]$ be a $c \times n$ standard membership matrix where δ_{ik} is a membership degree of the object k to the cluster C_i.

The MFCM method aims to minimize the objective function given as

$$J^1 = \sum_{i=1}^{c} \sum_{j=1}^{p} \sum_{k=1}^{n} (u_{ijk})^m d_{ijk}, \tag{1}$$

where d_{ijk} is the squared Euclidian distance that measures the dissimilarity between x_{jk} of object k and y_{ij} of prototype i of cluster

C_i for a given variable j. It is calculated as

$$d_{ijk} = \left(x_{jk} - y_{ij}\right)^2. \quad (2)$$

Regarding single variable, prototypes take into account the degree of belonging and quantitative values of each object from Ω:

$$y_{ij} = \frac{\sum_{k=1}^{n} (u_{ijk})^m x_{jk}}{\sum_{k=1}^{n} (u_{ijk})^m}. \quad (3)$$

And multivariate membership degree of a given object k to a cluster i concerning a variable j is calculated as follows:

$$u_{ijk} = \left[\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{d_{ijk}}{d_{abk}}\right)^{1/(m-1)}\right]^{-1}. \quad (4)$$

In the MFCM method, the matrix of memberships by cluster $\Delta = [\delta_{ik}]$ is obtained as follows:

$$\delta_{ik} = \sum_{j=1}^{p} u_{ijk}, \quad (5)$$

and it satisfies the following restrictions:

1. $\delta_{ik} \in [0,1]$ for all i and k;
2. $0 < \sum_{k=1}^{n} \delta_{ik} < n$ for all i and
3. $\sum_{i=1}^{c} \delta_{ik} = 1$ for all k.

The algorithm for Multivariate Fuzzy C-Means method is described as follows:

Algorithm 1. MFCM (Ω, c).

```
Require: Data set $\Omega$ and the number of clusters $c$;
Let $\varepsilon > 0$. Fix $c$, $2 \le c \le n$; fix $m$, $1 < m < \infty$; fix T; and fix $\varepsilon > 0$;
Initialize randomly $u_{ijk}(i=1,\dots,c;\ j=1,\dots,p$ and $k=1,\dots,n)$ of object $k$ belonging to cluster $C_i$ for variable $j$ such that $u_{ijk} \in [0,1]$, $0 < \sum_{k=1}^{n} u_{ijk} < n$ and $\sum_{i=1}^{c}\sum_{j=1}^{p} u_{ijk} = 1$, for all $k \in \Omega$.;
$t \leftarrow 0$;
$J^1(t) \leftarrow 0$;
$J^1(t+1) \leftarrow \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m d_{ijk}$;
While $|J^1(t) - J^1(t+1)| > \varepsilon$ and $t < T$ do
    Update matrix of prototype Y: Fixing the membership $u_{ijk}$ update the prototypes $y_{ij}$ using Eq. (3);
    Update matrix of multivariate membership U: Fixing the prototype $y_{ij}$ $(i=1,\dots,c;\ j=1,\dots,p$ and $k=1$ ,...., $n)$ update the membership $u_{ijk}$ using Eq. (4);
        $J^1(t) \leftarrow J^1(t+1)$;
    $J^1(t+1) \leftarrow \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m d_{ijk}$;
    $t \leftarrow t+1$;
end while
Compute matrix of memberships by cluster $\Delta$: Aggregate the multivariate memberships using Eq. (5).
return Matrices Y and U.
```

The time complexity of the algorithm can be understood according to its different steps. In the initialization, the parameters that will be used during the execution such as ε, m, T, weights and memberships are set. This step needs cpn operations for matrix **U**, so the time complexity for initialization step is $O(cpn)$. The algorithm performs iteratively two steps until the convergence. In the first one, prototypes are updated and the algorithm takes pn operations for each one of c prototypes, therefore the time complexity of this step is $O(cpn)$. In the second step, there are cpn membership values where each one takes cp operations, thus the time complexity of this step is $O(c^2p^2n)$. Therefore, the worst-case computational time complexity of the algorithm is $O(c^2p^2n)$.

In order to compute the space complexity of the algorithm, it can be understood according to its matrices. The matrix of prototype **Y** requires p space for each one of c prototypes, thus it is necessary $O(cp)$ space to store this matrix. For the matrix of membership **U**, there are n matrices of multivariate membership matrices where each one requires cp space, thus the space complexity of matrix **U** is $O(cpn)$. Therefore, the space complexity of this algorithm is $O(cpn)$.

After the convergence of algorithm, crisp cluster boundaries may be obtained by a hard partition $P = \{C_1, \dots, C_i, \dots, C_c\}$ where $\bigcup_{i=1}^{c} C_i = \Omega$ and $\bigcap_{i=1}^{c} C_i = \varnothing$. Each object k is allocated to cluster C_{i*} such that

$$i* = \underbrace{\arg\max}_{1 \le i \le c} \delta_{ik}. \quad (6)$$

3. Multivariate Fuzzy C-Means algorithms with weighting

In the following sections, weighted Multivariate Fuzzy C-Means methods are proposed. Two different ways to weight the MFCM are introduced. In the first one (called WMFCM-M), the importance of each multivariate membership degree is taken into account to compute the final object assignment. This method is based on a weighted multivariate fuzzy c-means for interval data of the Symbolic Data Analysis literature presented in [35]. The second one (called WMFCM-D) tries to identify the shape of clusters according to dispersions for each cluster and variable.

For the two methods presented ahead, consider $\mathbb{Y}^c = \underbrace{\mathbb{Y} \times \cdots \times \mathbb{Y}}_{c}$ and $\mathbb{Y} = \underbrace{\Re \times \cdots \times \Re}_{p} = \Re^p$, where $\mathbb{Y}$ is the representation space of prototypes such that $\mathbf{y}_i \in \mathbb{Y}$ and $\mathbf{Y} \in \mathbb{Y}^c$. Moreover, consider $\mathbb{U}^n = \underbrace{\mathbb{U} \times \cdots \times \mathbb{U}}_{n}$, $\mathbb{U} = \underbrace{\mathbb{V} \times \cdots \times \mathbb{V}}_{c}$ and $\mathbb{V} = \underbrace{[0,1] \times \cdots \times [0,1]}_{p}$. Thus, $\mathbb{U}$ is the representation space of multivariate membership matrix $\mathbf{u}_k$ such that $\mathbf{u}_k \in \mathbb{U}$ and $\mathbf{U} \in \mathbb{U}^n$.

3.1. Weight in the membership

In this method, each object, cluster and variable has a suitable weight. The goal is to identify what is the relevance of a given variable when computing the membership degree for an object regarding a cluster. This relevance depends on the localization of object in the partition, therefore each object has a particular interpretation, which may reduce an ambiguous assignment. As example, consider Fig. 1. It contains two classes where the prototypes are in the position (10, 10) and (40, 10), respectively. Using the MFCM method to classify the object (circle in the position (24, 11)), the final membership degree by cluster is very nearly the

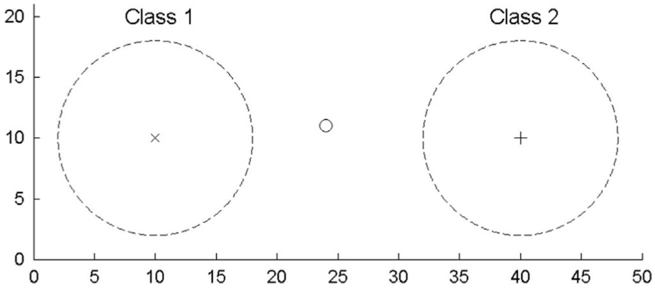

Fig. 1. Two classes and an object in an ambiguity zone.

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

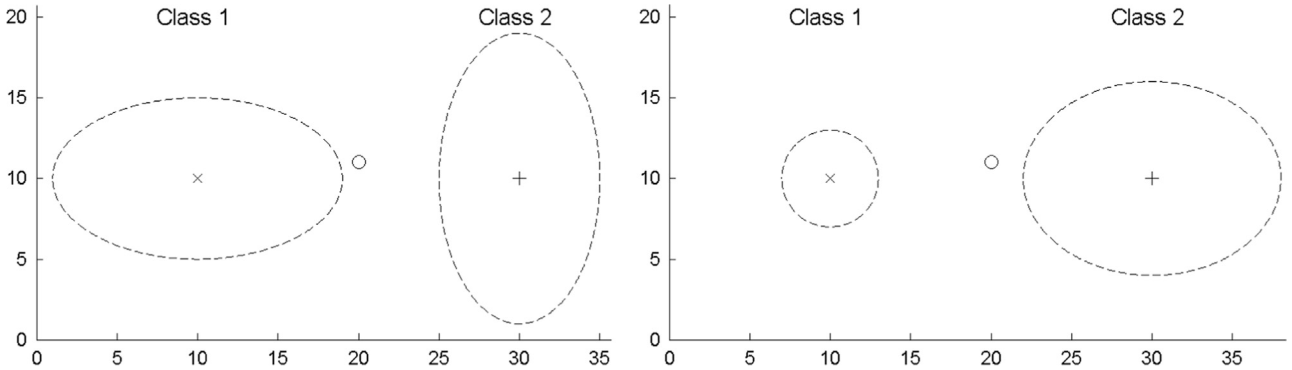

Fig. 2. Two pairs of classes with different shapes and sizes.

Table 1
Parameters for data sets 1 and 2.

Parameter	Data set 1			Data set 2		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
μ_1	5	15	18	0	30	12
μ_2	0	5	14	0	0	25
${\sigma_1}^2$	81	9	25	100	49	16
${\sigma_2}^2$	9	100	36	100	49	16
Cardinality	200	100	50	200	100	50

Table 2
Parameters for data sets 3 and 4.

Parameter	Data set 3			Data set 4		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
μ_1	0	15	15	0	15	−15
μ_2	0	5	−5	0	0	0
σ_1^2	100	100	100	16	16	16
σ_2^2	4	4	4	16	16	16
Cardinality	100	100	100	150	100	50

same (both close to 50%), even though the object is closer to Class 1. Therefore, weights should be used to reduce this ambiguity.

Let $\mathbf{\Gamma}=[\mathbf{\Gamma}_k]$ be a matrix of weight matrices, where for a given object k $(k=1,\ldots,n)$ $\mathbf{\Gamma}_k=[\gamma_{ijk}]$ is a $c\times p$ weight matrix where γ_{ijk} is the weight of the membership u_{ijk} regarding the cluster C_i $(i=1,\ldots,c)$ and variable j $(j=1,\ldots,p)$.

The goal of Weighted Multivariate Fuzzy C-Means (here abbreviated by WMFCM-M) is to look for a matrix of prototypes $\mathbf{Y}^*$, a matrix of multivariate memberships $\mathbf{U}^*$ and a matrix of weights $\mathbf{\Gamma}^*$ such that

$$J^2(\mathbf{Y}^*,\mathbf{U}^*,\mathbf{\Gamma}^*)=\min\{J^2(\mathbf{Y},\mathbf{U},\mathbf{\Gamma}):\mathbf{Y}\in\mathbb{Y}^c,\mathbf{U}\in\mathbb{U}^n,\mathbf{\Gamma}\in\mathbb{G}^n\},\tag{7}$$

where $\mathbb{G}^n=\underbrace{\mathbb{G}\times\cdots\times\mathbb{G}}_{n}$, $\mathbb{G}=\underbrace{\mathbb{H}\times\cdots\times\mathbb{H}}_{c}$ and $\mathbb{H}=\Re^*_+p$, where $\mathbb{G}$ is the representation space of weight matrix $\mathbf{\Gamma}_k$ such that $\mathbf{\Gamma}_k\in\mathbb{G}$ and $\mathbf{\Gamma}\in\mathbb{G}^n$.

The algorithm aims to minimize the objective function given as follows:

$$J^2(\mathbf{Y},\mathbf{U},\mathbf{\Gamma})=\sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m d_{ijk},\tag{8}$$

where d_{ijk} is the squared Euclidian distance that measures the dissimilarity between object x_{jk} and prototype y_{ij} for a given variable j as presented in Eq. (2).

Suitable prototypes are computed taking into account the weights, unlike the Multivariate Fuzzy C-Means presented in the previous section.

Proposition 1. *Fixing the membership degree u_{ijk} and the weight γ_{ij}, the prototype y_{ij} of cluster C_i for variable j that minimizes the criterion J^2 is updated using the following equation*:

$$y_{ij}=\frac{\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m x_{jk}}{\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m}.\tag{9}$$

Proof. The proof can be obtained in a similar way as described in [35] for the standard quantitative data case.□

The membership also uses the values of weights to represent the degree in which each given object belongs to each cluster. The greater the weight value, the greater the membership degree.

Proposition 2. *Fixing the membership weight γ_{ijk} and the prototype y_{ij}, the membership degree u_{ijk} under the restriction $\sum_{i=1}^{c}\sum_{j=1}^{p}u_{ijk}=1$ that minimizes the criterion J^2 is updated using the following expression*:

$$u_{ijk}=\left[\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{d_{ijk}}{d_{abk}}\right)^{1/(m-1)}\left(\frac{\gamma_{abk}}{\gamma_{ijk}}\right)^{m/(m-1)}\right]^{-1}.\tag{10}$$

Proof. The proof can be obtained in a similar way as described in [35] for the standard quantitative data case.□

The weight in the WMFCM-M method uses the membership degree and distances between a given object and each prototype in order to measure the relevance of a variable to final membership degree regarding a given cluster.

Proposition 3. *Fixing the membership degree u_{ijk} and the prototype y_{ij}, the weight γ_{ijk} that minimizes the criterion J^2 under the restriction*

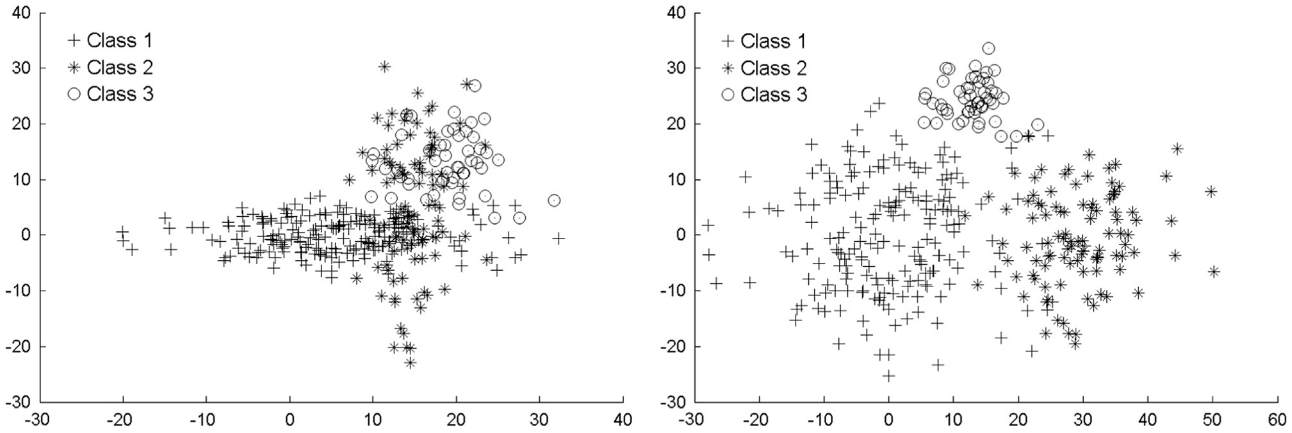

Fig. 3. Data sets 1 (left) and 2 (right).

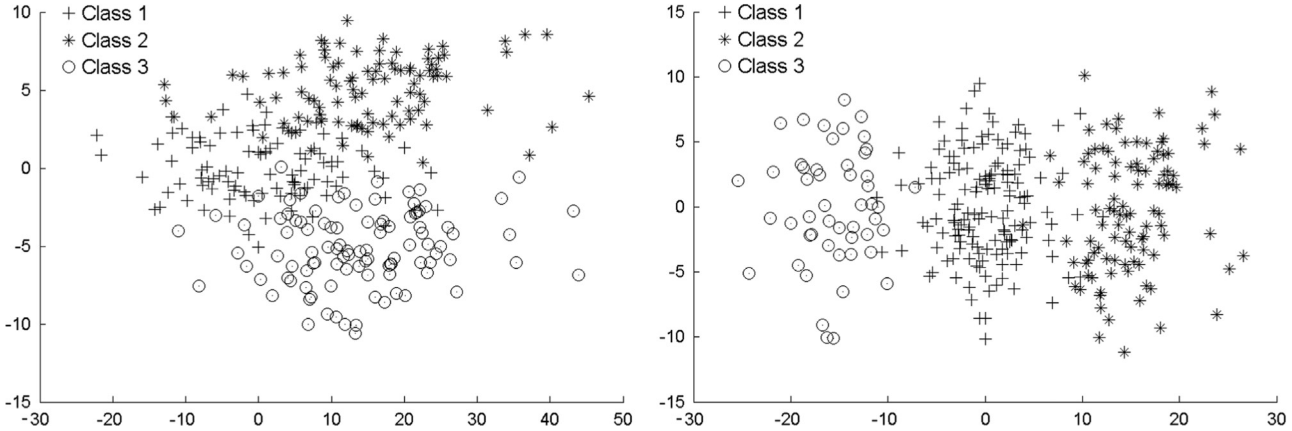

Fig. 4. Data sets 3 (left) and 4 (right).

Table 3
Average (larger values in bold) and standard deviation (in parentheses) of the FR index for clustering methods and data sets 1–4.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
1	0.47886 (0.01061)	0.44645 (0.01183)	0.46431 (0.00982)	0.46755 (0.01391)	0.53735 (0.01337)	**0.64188** (0.01335)	0.56830 (0.01488)
2	0.62227 (0.01423)	0.59497 (0.02975)	0.59820 0.00634	0.58164 (0.03336)	0.48719 (0.01713)	0.52652 (0.01752)	**0.63563** (0.02653)
3	0.53974 (0.01626)	0.39248 (0.01415)	0.43684 (0.01356)	0.53464 (0.01475)	0.55166 (0.02189)	**0.73930** (0.01242)	0.59868 (0.01728)
4	0.63744 (0.01633)	0.62211 (0.02467)	0.57367 (0.01967)	0.53820 (0.02686)	0.36639 (0.01864)	0.42979 (0.02767)	**0.69547** (0.10303)

$\sum_{i=1}^{c}\sum_{j=1}^{p}\gamma_{ijk}u_{ijk}=1$ *is updated using the following equation*:

$$\gamma_{ijk}=\frac{\left[(u_{ijk})^m d_{ijk}\right]^{1/(m+1)}}{\sum_{a=1}^{c}\sum_{b=1}^{p}\left[(u_{abk})^m d_{abk}\right]^{1/(m+1)}u_{abk}}. \tag{11}$$

Proof. The proof can be obtained in a similar way as described in [35] for standard quantitative data case.□

In the WMFCM-M method, the final membership degree matrix $\Delta=[\delta_{ik}]$ is such that δ_{ik} is the combination between multivariate weights and memberships and satisfies the three restrictions previously presented. It is given as

$$\delta_{ik}=\sum_{j=1}^{p}\gamma_{ijk}u_{ijk}. \tag{12}$$

3.1.1. *The algorithm*

The algorithm for Weighted Multivariate Fuzzy C-Means method is described as follows:

Algorithm 2. WMFCM-M (Ω, c).

```
Require: Data set $\Omega$ and the number of clusters $c$;
Let $\varepsilon > 0$. Fix $c$, $2 \le c \le n$; fix $m$, $1 < m < \infty$; fix T; and fix $\varepsilon > 0$;
Initialize randomly $u_{ijk}(i=1,\dots,c;\ j=1,\dots,p$ and $k=1,\dots,n)$ of
    object $k$ belonging to cluster $C_i$ for variable $j$ such that
    $u_{ijk}\in[0,1]$, $0<\sum_{k=1}^{n}u_{ijk}<n$ and $\sum_{i=1}^{c}\sum_{j=1}^{p}u_{ijk}=1$, for all
    $k\in\Omega$. Let $\gamma_{ijk}=1$, for all $i$, $j$ and $k$;
$t \leftarrow 0$;
$J^2(t) \leftarrow 0$;
```

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

Table 4
Statistical tests comparing WMFCM-M method for data sets 1–4.

Data set	Statistical test					
1	H_0: $\mu_1 \geq \mu_6$ H_1: $\mu_1 < \mu_6$ -95.1185 (Reject H_0)	H_0: $\mu_2 \geq \mu_6$ H_1: $\mu_2 < \mu_6$ -109.0130 (Reject H_0)	H_0: $\mu_3 \geq \mu_6$ H_1: $\mu_3 < \mu_6$ -106.6089 (Reject H_0)	H_0: $\mu_4 \geq \mu_6$ H_1: $\mu_4 < \mu_6$ -101.5645 (Reject H_0)	H_0: $\mu_5 \geq \mu_6$ H_1: $\mu_5 < \mu_6$ -55.0474 (Reject H_0)	H_0: $\mu_7 \geq \mu_6$ H_1: $\mu_7 < \mu_6$ -36.6222 (Reject H_0)
2	H_0: $\mu_1 \leq \mu_6$ H_1: $\mu_1 > \mu_6$ 42.2093 (Reject H_0)	H_0: $\mu_2 \leq \mu_6$ H_1: $\mu_2 > \mu_6$ 19.7265 (Reject H_0)	H_0: $\mu_3 \leq \mu_6$ H_1: $\mu_3 > \mu_6$ 38.2789 (Reject H_0)	H_0: $\mu_4 \leq \mu_6$ H_1: $\mu_4 > \mu_6$ 11.6470 (Reject H_0)	H_0: $\mu_5 \geq \mu_6$ H_1: $\mu_5 < \mu_6$ -15.9708 (Reject H_0)	H_0: $\mu_7 \leq \mu_6$ H_1: $\mu_7 > \mu_6$ 34.1469 (Reject H_0)
3	H_0: $\mu_1 \geq \mu_6$ H_1: $\mu_1 < \mu_6$ -97.0439 (Reject H_0)	H_0: $\mu_2 \geq \mu_6$ H_1: $\mu_2 < \mu_6$ -183.2848 (Reject H_0)	H_0: $\mu_3 \geq \mu_6$ H_1: $\mu_3 < \mu_6$ -163.6604 (Reject H_0)	H_0: $\mu_4 \geq \mu_6$ H_1: $\mu_4 < \mu_6$ -104.5144 (Reject H_0)	H_0: $\mu_5 \geq \mu_6$ H_1: $\mu_5 < \mu_6$ -74.1813 (Reject H_0)	H_0: $\mu_7 \geq \mu_6$ H_1: $\mu_7 < \mu_6$ -65.7484 (Reject H_0)
4	H_0: $\mu_1 \leq \mu_6$ H_1: $\mu_1 > \mu_6$ 64.3053 (Reject H_0)	H_0: $\mu_2 \leq \mu_6$ H_1: $\mu_2 > \mu_6$ 51.6192 (Reject H_0)	H_0: $\mu_3 \leq \mu_6$ H_1: $\mu_3 > \mu_6$ 58.0784 (Reject H_0)	H_0: $\mu_4 \leq \mu_6$ H_1: $\mu_4 > \mu_6$ 36.2337 (Reject H_0)	H_0: $\mu_5 \geq \mu_6$ H_1: $\mu_5 < \mu_6$ -18.9079 (Reject H_0)	H_0: $\mu_7 \leq \mu_6$ H_1: $\mu_7 > \mu_6$ 24.7793 (Reject H_0)

Table 5
Statistical tests comparing WMFCM-D method for data sets 1–4.

Data set	Statistical test					
1	H_0: $\mu_1 \geq \mu_7$ H_1: $\mu_1 < \mu_7$ -48.6950 (Reject H_0)	H_0: $\mu_2 \geq \mu_7$ H_1: $\mu_2 < \mu_7$ -63.7780 (Reject H_0)	H_0: $\mu_3 \geq \mu_7$ H_1: $\mu_3 < \mu_7$ -58.0364 (Reject H_0)	H_0: $\mu_4 \geq \mu_7$ H_1: $\mu_4 < \mu_7$ -60.1432 (Reject H_0)	H_0: $\mu_5 \geq \mu_7$ H_1: $\mu_5 < \mu_7$ -15.3941 (Reject H_0)	H_0: $\mu_6 \leq \mu_7$ H_1: $\mu_6 > \mu_7$ 36.6222 (Reject H_0)
2	H_0: $\mu_1 \geq \mu_7$ H_1: $\mu_1 < \mu_7$ -4.4155 (Reject H_0)	H_0: $\mu_2 \geq \mu_7$ H_1: $\mu_2 < \mu_7$ -10.1493 (Reject H_0)	H_0: $\mu_3 \geq \mu_7$ H_1: $\mu_3 < \mu_7$ -13.6534 (Reject H_0)	H_0: $\mu_4 \geq \mu_7$ H_1: $\mu_4 < \mu_7$ -10.2982 (Reject H_0)	H_0: $\mu_5 \geq \mu_7$ H_1: $\mu_5 < \mu_7$ -46.7693 (Reject H_0)	H_0: $\mu_6 \geq \mu_7$ H_1: $\mu_6 < \mu_7$ -34.1469 (Reject H_0)
3	H_0: $\mu_1 \geq \mu_7$ H_1: $\mu_1 < \mu_7$ -24.7160 (Reject H_0)	H_0: $\mu_2 \geq \mu_7$ H_1: $\mu_2 < \mu_7$ -91.8616 (Reject H_0)	H_0: $\mu_3 \geq \mu_7$ H_1: $\mu_3 < \mu_7$ -73.3107 (Reject H_0)	H_0: $\mu_4 \geq \mu_7$ H_1: $\mu_4 < \mu_7$ -29.8188 (Reject H_0)	H_0: $\mu_5 \geq \mu_7$ H_1: $\mu_5 < \mu_7$ -16.7755 (Reject H_0)	H_0: $\mu_6 \leq \mu_7$ H_1: $\mu_6 > \mu_7$ 65.7484 (Reject H_0)
4	H_0: $\mu_1 \geq \mu_7$ H_1: $\mu_1 < \mu_7$ -5.5350 (Reject H_0)	H_0: $\mu_2 \geq \mu_7$ H_1: $\mu_2 < \mu_7$ -6.8898 (Reject H_0)	H_0: $\mu_3 \geq \mu_7$ H_1: $\mu_3 < \mu_7$ -7.3342 (Reject H_0)	H_0: $\mu_4 \geq \mu_7$ H_1: $\mu_4 < \mu_7$ -11.2403 (Reject H_0)	H_0: $\mu_5 \geq \mu_7$ H_1: $\mu_5 < \mu_7$ -31.2724 (Reject H_0)	H_0: $\mu_6 \geq \mu_7$ H_1: $\mu_6 < \mu_7$ -24.7793 (Reject H_0)

Table 6
Average and standard deviation (in parentheses) of the number of iterations for clustering methods and data sets 1–4.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
1	53.90000 (26.28479)	33.17362 (10.42433)	26.15000 (7.81841)	35.29473 (7.25718)	82.33333 (12.55211)	5.28413 (0.13294)	8.33333 (6.79869)
2	37.20000 (11.11575)	15.66667 (2.49444)	29.35000 (4.65054)	32.33333 (1.69967)	72.33333 (29.51083)	3.33333 (0.47140)	19.33333 (1.24722)
3	34.10000 (3.91024)	25.66667 (2.62467)	24.15000 (4.98272)	43.33333 (3.29983)	65.33333 (4.02768)	5.66667 (0.47140)	11.27463 (8.48528)
4	18.70000 (2.53180)	14.33333 (4.18994)	16.65000 (1.15217)	18.66667 (1.24722)	78.33333 (15.36952)	3.18439 (0.37138)	13.66667 (0.47140)

Table 7
Average and standard deviation (in parentheses) of convergence time (in milliseconds) for clustering methods and data sets 1–4.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
1	579.60000 (278.95383)	105.33333 (30.66304)	321.25000 (144.12456)	190.47361 (39.14929)	582.66667 (67.96731)	48.36533 (1.88562)	78.37264 (57.86767)
2	462.50000 (182.50877)	104.33333 (18.37269)	355.95000 (117.38589)	472.33333 (159.34310)	1135.27462 (827.31534)	105.58372 (30.34249)	176.66667 (9.56847)
3	379.90000 (96.18986)	110.37264 (6.68331)	306.25000 (163.73389)	532.37427 (75.27284)	769.66667 (316.36933)	159.37264 (40.83571)	238.33333 (247.38813)
4	264.30000 (77.06887)	48.33333 (25.48638)	199.55000 (69.17259)	176.66667 (36.73630)	637.33333 (196.99464)	58.47361 (22.75961)	102.33333 (5.90668)

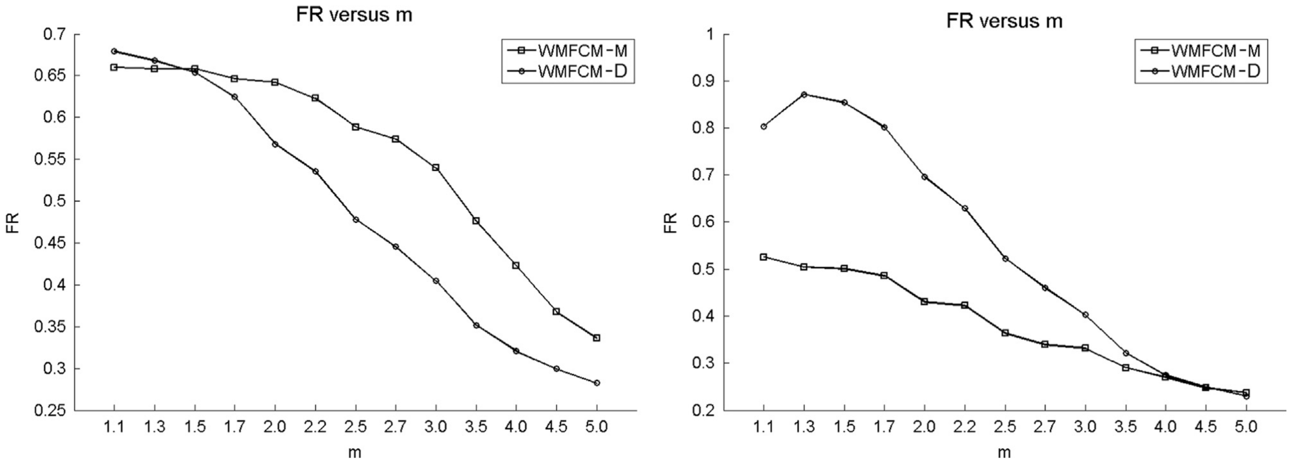

Fig. 5. Average of FR index according to parameter m for data sets 1 (left) and 4 (right) according to WMFCM-M and WMFCM-D methods.

Table 8
Parameters for data sets 5 and 6.

Parameter	Data set 5				Data set 6			
	Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4
μ_1	5	15	10	3	5	15	12	7
μ_2	0	5	−7	15	0	5	−12	17
σ_1^2	81	9	49	25	81	9	16	25
σ_2^2	9	100	16	25	9	100	16	25
Cardinality	50	50	50	50	50	50	50	50

```
$J^2(t+1) \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{p} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m d_{ijk}$;
while $|J^2(t) - J^2(t+1)| > \varepsilon$ and $t < T$ do
    Update matrix of prototype $\mathbf{Y}$: fixing the membership $u_{ijk}$ and weight $\gamma_{ijk}$ $(i=1,\ldots,c;\ j=1,\ldots,p$ and $k=1,\ldots,n)$ update the prototypes $y_{ij}$ using Eq. (9);
    Update matrix of multivariate membership $\mathbf{U}$: fixing the weight $\gamma_{ijk}$ and the prototype $y_{ij}$ $(i=1,\ldots,c;\ j=1,\ldots,p$ and $k=1,\ldots,n)$ update the membership $u_{ijk}$ using Eq. (10);
    Update matrix of weight $\mathbf{\Gamma}$: fixing the membership $u_{ijk}$ and the prototype $y_{ij}$ $(i=1,\ldots,c;\ j=1,\ldots,p$ and $k=1$ ,..., $n)$ update the weight $\gamma_{ijk}$ using Eq. (11);
    $J^2(t) \leftarrow J^2(t+1)$;
    $J^2(t+1) \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{p} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m d_{ijk}$;
    $t \leftarrow t+1$;
end while
Compute matrix of memberships by cluster $\Delta$: aggregate the multivariate memberships using Eq. (12).
return Matrices $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{\Gamma}$.
```

Table 9
Parameters for data set 7.

Parameter	Data set 7			
	Class 1	Class 2	Class 3	Class 4
μ_1	0	18	−18	0
μ_2	0	0	0	−12
σ_1^2	12	20	16	81
σ_2^2	12	20	16	20
Cardinality	50	50	50	50

The algorithm can be understood according to its different steps as described below. In the first step, the algorithm sets the parameters ε, m and T. In this step, the weights and memberships are also initialized and it is necessary cpn operations for the matrices $\mathbf{U}$ and $\mathbf{\Gamma}$, thus the time complexity for initialization step is $O(cpn)$. Until the convergence, the algorithm performs iteratively three main steps. In the first one, to update the prototypes, the algorithm takes pn operations for each one of c prototypes, thus the time complexity of this step is $O(cpn)$. In the second step, memberships are updated so that there are cpn membership values where each one takes cp calculations, then the time complexity of this step is $O(c^2p^2n)$. In the third step, weights for a given cluster, object and variable are calculated. It is necessary cp operations for each weight and there are cpn weights, thus this step takes $O(c^2p^2n)$. To compute the objective function value used in the stopping criterion, the processing time is $O(cpn)$. Therefore, the final time complexity of the algorithm is $O(c^2p^2n)$.

Regarding the space complexity, the matrix of prototype $\mathbf{Y}$ requires p space for each one of c prototypes, thus it is necessary $O(cp)$ space to store this matrix. The matrix of weight $\mathbf{\Gamma}$ is formed by n matrices of multivariate weights where each one requires cp space, thus the final space complexity to store $\mathbf{\Gamma}$ is $O(cpn)$. A similar reasoning can be made for the matrix of membership $\mathbf{U}$. Therefore, the space complexity of this algorithm is $O(cpn)$.

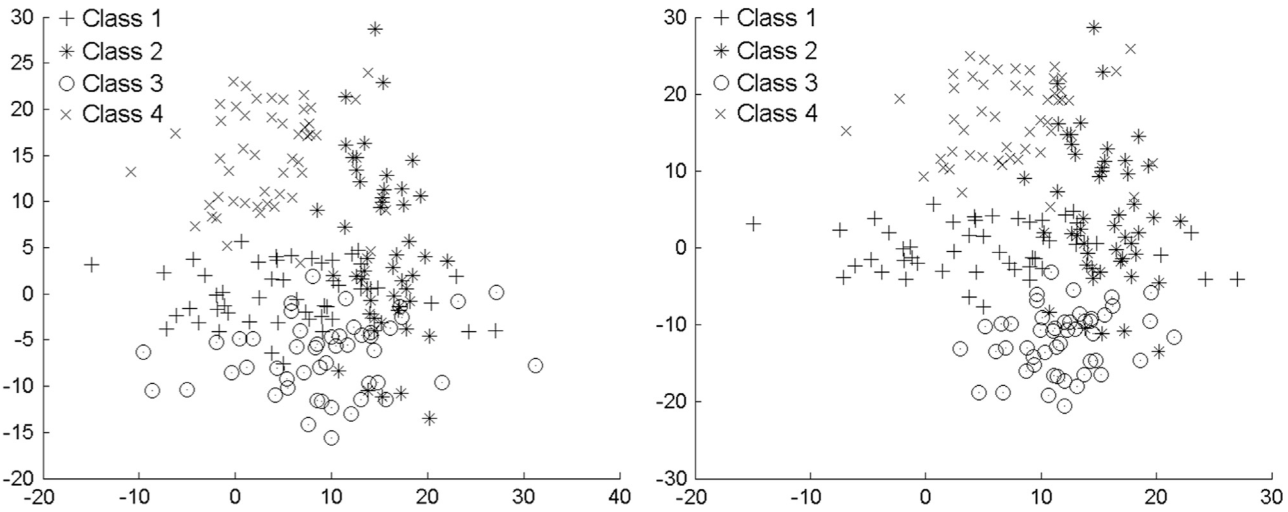

Fig. 6. Data sets 5 (left) and 6 (right).

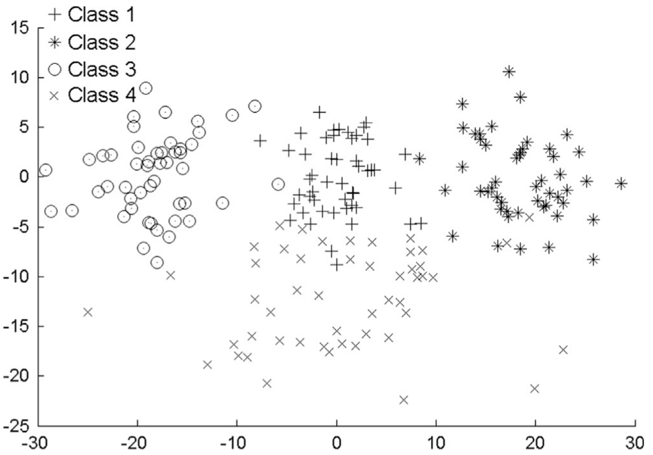

Fig. 7. Data set 7.

Table 10
Average and standard deviation (in parentheses) of the FR index for clustering methods and data sets 5–7.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
5	0.55685 (0.02102)	0.48399 (0.01815)	0.51057 (0.01048)	0.49736 (0.01965)	0.55761 (0.00926)	0.69660 (0.01869)	0.58713 (0.01922)
6	0.59386 (0.01657)	0.54137 (0.02410)	0.58072 (0.01277)	0.54326 (0.01967)	0.57988 (0.01241)	0.67661 (0.02068)	0.65027 (0.02161)
7	0.65071 (0.01340)	0.60568 (0.03666)	0.61288 (0.01405)	0.60086 (0.05360)	0.54247 (0.02137)	0.62529 (0.04480)	0.67322 (0.01454)

Table 11
Summarized properties of the UCI data sets.

Data set	#Classes	#Features	#Instances
Abalone	3	8	4177
Ecoli	6	8	336
Glass identification	7	10	214
Haberman's survival	2	3	306
Pima Indians diabetes	2	8	768
Statlog vehicle	4	18	946
Vertebral column	2	6	310
Wine	3	13	178
Wine quality	11	12	4898

Table 12
FR indexes (larger values in bold) for clustering methods and UCI data sets.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
Abalone	0.135	0.134	0.133	0.134	0.186	0.178	**0.199**
Ecoli	0.602	0.598	0.595	0.596	0.630	0.633	**0.637**
Glass identification	0.527	0.524	0.527	0.526	0.567	0.565	**0.576**
Haberman's survival	0.133	0.089	0.133	0.130	0.140	0.141	**0.167**
Pima Indians diabetes	0.079	0.061	0.078	0.079	0.072	0.080	**0.093**
Statlog vehicle	0.289	0.285	0.287	0.287	0.345	0.344	**0.374**
Vertebral column	0.117	0.118	0.117	0.112	0.100	0.113	**0.119**
Wine	0.149	0.147	0.148	0.148	0.207	0.202	**0.225**
Wine quality	0.582	0.579	0.580	0.580	0.580	**0.603**	0.602

Table 13
Convergence time (in milliseconds) for clustering methods and UCI data sets.

Data set	GK	FCM	FCM-C^1	FCM-C^2	MFCM	WMFCM-M	WMFCM-D
Abalone	1401	730	1312	2534	5135	4213	8912
Ecoli	452	235	507	851	1661	1413	2954
Glass identification	47,434	270	670	790	1816	1300	1432
Haberman's survival	16	21	57	58	63	53	90
Pima Indians diabetes	1767	211	1343	1383	1880	613	1293
Statlog vehicle	1783	157	1177	1094	2410	782	3995
Vertebral column	53	33	143	141	228	197	352
Wine	173	138	922	1021	2440	2036	3136
Wine quality	17,430	4174	16,315	18,874	50,085	13,291	84,917

3.1.2. Convergence of the algorithm

According to [38], two series may be considered regarding this type of algorithm: $\nu_t = (\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Gamma}^t) \in \mathbb{Y}^c \times \mathbb{U}^n \times \mathbb{G}^n$ and $\omega_t = J^2(\nu_t) = J^2(\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Gamma}^t)$, $t = 0, 1, 2, \ldots, T$. The algorithm starts from initial series $\nu_0 = (\mathbf{Y}^0, \mathbf{U}^0, \mathbf{\Gamma}^0)$ and computes the different terms of the series ν_t until it converges when the criterion J^2 achieves a stationary value.

Proposition 4. *The series $\omega_t = J^2(\nu_t)$ decreases at each iteration and converges.*

Proof. The proof can be obtained in a similar way as described in [35].□

Proposition 5. *The series $\nu_t = (\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Gamma}^t)$ converges.*

Proof. The proof can be obtained in a similar way as described in [35].□

Table 14
Overall heterogeneity indices for multivariate methods.

R^1	0.7131
R^2	0.8090
R^3	0.8029

3.2. Weight in the distance

This method introduces a weighted distance in order to take into account the variability of each variable and cluster, therefore weights that change at each iteration of algorithm are computed. The advantage of adaptive distances is that the clustering algorithm is able to find clusters of different shapes and sizes. Fig. 2 shows two partitions containing two classes where prototypes are in the position $(10, 10)$ and $(30, 10)$, respectively. Also, consider an object in the position $(20, 11)$. Note that the object should be allocated to Class 1 in the left-side partition and Class 2 in the right-side one. However the object will have the same multivariate matrix if the MFCM method is used. Thus, suitable weights that represent the shape of clusters should be considered in order to improve the cluster quality.

Let $\mathbf{\Lambda} = [\boldsymbol{\lambda}_i]$ be a matrix of weight vectors, where $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ij}, \ldots, \lambda_{ip})$ and $\lambda_{ij} \in \Re^*_+$ is the weight regarding the cluster $C_i (i = 1, \ldots, c)$ and variable j $(j = 1, \ldots, p)$.

The goal of Multivariate Fuzzy C-Means weighting the distance (here abbreviated by WMFCM-D) is to look for a matrix of prototypes $\mathbf{Y}^*$, a matrix of multivariate memberships $\mathbf{U}^*$ and a matrix of weights $\mathbf{\Lambda}^*$ such that

$$J^3(\mathbf{Y}^*, \mathbf{U}^*, \mathbf{\Lambda}^*) = \min\{J^3(\mathbf{Y}, \mathbf{U}, \mathbf{\Lambda}) : \mathbf{Y} \in \mathbb{Y}^c, \mathbf{U} \in \mathbb{U}^n, \mathbf{\Lambda} \in \mathbb{L}^c\}, \tag{13}$$

where $\mathbb{L}^c = \underbrace{\mathbb{L} \times \cdots \times \mathbb{L}}_{c}$, $\mathbb{L} = \underbrace{\Re^*_+ \times \cdots \times \Re^*_+}_{p} = \Re^*_+ p$, and $\mathbb{L}$ is the representation space of weight vector $\boldsymbol{\lambda}_i$ such that $\boldsymbol{\lambda}_i \in \mathbb{L}$ and $\mathbf{\Lambda} \in \mathbb{L}^c$.

The objective function $J^3(\mathbf{Y}, \mathbf{U}, \mathbf{\Lambda})$ of WMFCM-D is given as

$$J^3(\mathbf{Y}, \mathbf{U}, \mathbf{\Lambda}) = \sum_{i=1}^{c} \sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m d_{ijk}, \tag{14}$$

where d_{ijk} is the squared Euclidian distance between object x_{jk} and prototype y_{ij} for a given variable j calculated by Eq. (2).

Unlike traditional FCM, prototypes in proposed method are computed taking into account membership degree regarding each variable. Therefore, prototypes may better represent its respective cluster.

Proposition 6. *Fixing the membership degree u_{ijk} and the weight λ_{ij}, the prototype y_{ij} of cluster C_i for variable j that minimizes the criterion J^3 is updated using the following equation:*

$$y_{ij} = \frac{\sum_{k=1}^{n} (u_{ijk})^m x_{jk}}{\sum_{k=1}^{n} (u_{ijk})^m}. \tag{15}$$

Proof. The proof is given in Appendix A.

Multivariate memberships allow us to extract more information from data. That is, it is possible to compute the degree of belonging of a given object to a cluster according to each variable.

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

Table 15
Overall heterogeneity and between-cluster index according to each variable and method.

Method	Measure	Variable							
		1	2	3	4	5	6	7	8
MFCM	COR^1	0.2946	0.9601	0.9292	0.3167	0.1631	0.7956	0.8047	0.9156
	CTR^1	0.0000	0.0009	0.0001	0.1043	0.0015	0.0002	0.0004	0.8926
WMFCM-M	COR^2	0.6580	0.6293	0.5554	0.6820	0.0086	0.0667	0.0945	0.1439
	CTR^2	0.2741	0.2517	0.1353	0.2998	0.0012	0.0088	0.0121	0.0171
WMFCM-D	COR^3	0.2719	0.9618	0.9312	0.3139	0.1556	0.8083	0.8133	0.9071
	CTR^3	0.0063	0.3696	0.2479	0.0117	0.0054	0.0666	0.0713	0.2211

Table 16
Relative contribution by cluster of measures based on sum of square for each method.

Cluster	MFCM			WMFCM-M			WMFCM-D		
	$T^1(i)$	$B^1(i)$	$J^1(i)$	$T^2(i)$	$B^2(i)$	$J^2(i)$	$T^3(i)$	$B^3(i)$	$J^3(i)$
1	0.3278	0.4290	0.3204	0.3223	0.4180	0.3126	0.4643	0.4520	0.3333
2	0.3424	0.1841	0.3754	0.3476	0.1930	0.3986	0.3506	0.4007	0.3272
3	0.3298	0.3869	0.3042	0.3301	0.3890	0.2888	0.1851	0.1474	0.3396

Table 17
Interpretation indices by cluster and variable for MFCM method.

Variable	Cluster								
	1			2			3		
	COR	CTR	CE	COR	CTR	CE	COR	CTR	CE
1	0.0438	0.0000	0.0000	0.2351	0.0001	0.0000	0.0157	0.0000	0.0000
2	0.3862	0.0008	0.0004	0.2141	0.0011	0.0002	0.3598	0.0009	0.0003
3	0.3888	0.0001	0.0000	0.1888	0.0001	0.0000	0.3517	0.0001	0.0000
4	0.1763	0.1354	0.0581	0.0039	0.0069	0.0013	0.1366	0.1163	0.0450
5	0.0658	0.0014	0.0006	0.0084	0.0004	0.0001	0.0889	0.0021	0.0008
6	0.3924	0.0002	0.0001	0.0638	0.0001	0.0000	0.3395	0.0002	0.0001
7	0.3926	0.0004	0.0002	0.0735	0.0002	0.0000	0.3387	0.0004	0.0001
8	0.3792	0.8617	0.3697	0.1872	0.9911	0.1825	0.3492	0.8800	0.3405

This type of information increases the clustering quality, specially when features have dissimilar dispersions.

Proposition 7. *Fixing the weight λ_{ij} and the prototype y_{ij}, the membership degree u_{ijk} under the restriction $\sum_{i=1}^{c}\sum_{j=1}^{p} u_{ijk} = 1$ that minimizes the criterion J^3 is updated using the following expression*:

$$u_{ijk} = \left[\sum_{a=1}^{c} \sum_{b=1}^{p} \left(\frac{\lambda_{ij} d_{ijk}}{\lambda_{ab} d_{abk}} \right)^{1/(m-1)} \right]^{-1}. \qquad (16)$$

Proof. The proof is given in Appendix B.

The membership degree u_{ijk} of a given object $\mathbf{x}_k$ belonging to cluster C_i for the variable j satisfies the following restrictions:

1. $u_{ijk} \in [0,1]$ for all i, j and k;
2. $0 < \sum_{j=1}^{p} \sum_{k=1}^{n} u_{ijk} < n$ for all i and
3. $\sum_{i=1}^{c} \sum_{j=1}^{p} u_{ijk} = 1$ for all k.

The propose of weight is to compute the dispersion of features from data and increase the clustering quality. Multivariate memberships give more information to the clustering method, however the shape of clusters is not taken into account. Therefore, weights are computed to consider that clusters may be either ellipses or spheres. Thus, weights are measured at each algorithm iteration minimizing the criterion J^3.

There are two approaches for the derivation of weights according to constraints: (1) the sum of the weights of the variables must be equal to one [33,39,40] and (2) assuming that the product of the weights of the variables must be equal to one [33,32]. The last one is based on the Gustafson and Kessel algorithm [28]. It is known that the use of the sum of weights as constraint is more suitable in a specific case: elongated clusters [31,41], whereas there is no empirical study that shows restrictions for methods that use the product of the weights as constraint. Therefore, here, we adopted the constraint based on the product of weights.

Proposition 8. *Fixing the membership degree u_{ijk} and the prototype y_{ij}, the weight λ_{ij} that minimizes the criterion J^3 under the restriction $\lambda_{ij} > 0$ and $\prod_{j=1}^{p} \lambda_{ij} = 1$ is updated using the following equation*:

$$\lambda_{ij} = \frac{\left\{ \prod_{h=1}^{p} \left[\sum_{k=1}^{n} (u_{ihk})^m d_{ihk} \right] \right\}^{1/p}}{\sum_{k=1}^{n} (u_{ijk})^m d_{ijk}}. \qquad (17)$$

Proof. The proof is given in Appendix C.

After the convergence of the algorithm, the membership degree δ_{ik} of the object k to the cluster C_i can be similarly obtained as presented in Eq. (5) and it follows the three restrictions previously presented.

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

Table 18
Interpretation indices by cluster and variable for WMFCM-M method.

Variable	Cluster								
	1			2			3		
	COR	CTR	CE	COR	CTR	CE	COR	CTR	CE
1	0.3201	0.2951	0.1334	0.2951	0.3068	0.1229	0.0427	0.1208	0.0178
2	0.5470	0.4840	0.2188	0.0811	0.0809	0.0324	0.0013	0.0035	0.0005
3	0.0526	0.0283	0.0128	0.3525	0.2143	0.0859	0.1503	0.2484	0.0366
4	0.1636	0.1591	0.0719	0.3286	0.3605	0.1444	0.1898	0.5661	0.0834
5	0.0027	0.0008	0.0004	0.0032	0.0011	0.0004	0.0028	0.0026	0.0004
6	0.0212	0.0062	0.0028	0.0200	0.0066	0.0026	0.0255	0.0228	0.0034
7	0.0315	0.0089	0.0040	0.0315	0.0101	0.0040	0.0315	0.0274	0.0040
8	0.0666	0.0175	0.0079	0.0666	0.0198	0.0079	0.0106	0.0085	0.0013

Table 19
Interpretation indices by cluster and variable for WMFCM-D method.

Variable	Cluster								
	1			2			3		
	COR	CTR	CE	COR	CTR	CE	COR	CTR	CE
1	0.0418	0.0023	0.0010	0.2149	0.0257	0.0050	0.0153	0.0009	0.0004
2	0.3848	0.3538	0.1479	0.2042	0.4067	0.0785	0.3728	0.3682	0.1432
3	0.3845	0.2449	0.1024	0.1892	0.2610	0.0504	0.3574	0.2446	0.0952
4	0.1770	0.0158	0.0066	0.0042	0.0008	0.0002	0.1327	0.0128	0.0050
5	0.0651	0.0054	0.0023	0.0109	0.0020	0.0004	0.0796	0.0071	0.0028
6	0.3949	0.0779	0.0325	0.0639	0.0273	0.0053	0.3494	0.0740	0.0288
7	0.3925	0.0823	0.0344	0.0735	0.0334	0.0064	0.3473	0.0783	0.0305
8	0.3730	0.2175	0.0909	0.1924	0.2431	0.0469	0.3416	0.2141	0.0833

Table 20
Standard deviation of variables by class for Abalone data set.

Variable	Class 1	Class 2	Class 3
Length	0.0862	0.1089	0.1027
Diameter	0.0710	0.0881	0.0844
Height	0.0400	0.0320	0.0348
Whole weight	5.6881	1.6039	5.5324
Shucked weight	0.8416	0.1284	1.2405
Viscera weight	0.0976	0.0625	0.1049
Shell weight	0.1256	0.0849	0.1308
Number of rings	3.1043	2.5116	3.0263

Table 21
Standard deviation of variables by class for Ecoli data set.

Variable	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
mcg	0.1238	0.1945	0.0925	0.1086	0.0694	0.0444	0.0903
gvh	0.0896	0.0883	0.0712	0.0921	0.1200	0.0460	0.1291
lip	0.0000	0.0593	0.3002	0.0879	0.1163	0.0000	0.0000
chg	0.0000	0.0000	0.2500	0.0000	0.0000	0.0000	0.0000
aac	0.0879	0.1141	0.0645	0.1148	0.0939	0.1119	0.0856
alm1	0.0992	0.1032	0.1733	0.0843	0.0858	0.0207	0.1010
alm2	0.0961	0.1667	0.2844	0.0955	0.0839	0.1390	0.1183

3.2.1. *The algorithm*

The algorithm for Multivariate Fuzzy C-Means method weighting the distance is described as follows:

Algorithm 3. WMFCM-D (Ω, c).

```
Require:   Data set $\Omega$ and the number of clusters $c$;
Let $\varepsilon > 0$. Fix $c$, $2 \le c \le n$; fix $m$, $1 < m < \infty$; fix T; and fix $\varepsilon > 0$;
Initialize randomly $u_{ijk} (i = 1, \ldots, c;\ j = 1, \ldots, p$ and $k = 1\ ,\ldots,\ n)$ of
    object $k$ belonging to cluster $C_i$ for variable $j$ such that
    $u_{ijk} \in [0, 1]$, $0 < \sum_{k=1}^{n} u_{ijk} < n$ and $\sum_{i=1}^{c} \sum_{j=1}^{p} u_{ijk} = 1$, for all
    $k \in \Omega$. Let $\lambda_{ij} = 1$, for all $i$ and $j$;
$t \leftarrow 0$;
$J^3(t) \leftarrow 0$;
$J^3(t+1) \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m d_{ijk}$;
while $|J^3(t) - J^3(t+1)| > \varepsilon$ and $t < T$ do
    Update matrix of prototype Y: fixing the membership
      $u_{ijk}$ and weight $\lambda_{ij}$ $(i = 1, \ldots, c;\ j = 1, \ldots, p$ and $k = 1, \ldots, n)$
      update the prototypes $y_{ij}$ using Eq. (15);
    Update matrix of multivariate membership U: fixing
      the weight $\lambda_{ij}$ and the prototype $y_{ij}$ $(i = 1, \ldots, c;\ j = 1, \ldots, p$ and
      $k = 1\ ,\ldots,\ n)$ update the membership $u_{ijk}$ using Eq. (16);
    Update matrix of weight $\mathbf{\Lambda}$: fixing the membership $u_{ijk}$
      and the prototype $y_{ij}$ $(i = 1, \ldots, c;\ j = 1, \ldots, p$ and $k = 1\ ,\ldots,\ n)$
      update the weight $\lambda_{ij}$ using Eq. (17);
    $J^3(t) \leftarrow J^3(t+1)$;
    $J^3(t+1) \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m d_{ijk}$;
    $t \leftarrow t+1$;
end while
Compute matrix of memberships by cluster $\Delta$: Aggregate the multivariate memberships using Eq. (5).
return Matrices Y, U and $\mathbf{\Lambda}$.
```

In order to calculate the time complexity of the algorithm, it can be understood according to its different steps as described below. Initially, the parameters that will be used during the execution such as ε, m, T, weights and memberships are set. This

Table 22
Standard deviation of variables by class for glass identification data set.

Variable	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Refractive index	0.2268	0.3802	0.1916	0.3345	0.3116	0.2545
Sodium	0.4993	0.6642	0.5069	0.7770	1.0840	0.6864
Magnesium	0.2470	1.2157	0.1628	0.9991	1.0971	1.1177
Aluminum	0.2732	0.3183	0.3475	0.6939	0.5719	0.4427
Silicon	0.5695	0.7246	0.5123	1.2823	1.0795	0.9402
Potassium	0.2149	0.2137	0.2299	2.1387	0.0000	0.6685
Calcium	0.5748	1.9216	0.3801	2.1838	1.4499	0.9735
Barium	0.0838	0.3623	0.0364	0.6083	0.0000	0.6653
Iron	0.0891	0.1064	0.1079	0.1556	0.0000	0.0298

Table 23
Standard deviation of variables by class for Haberman's survival data set.

Variable	Class 1	Class 2
Age	11.0122	10.1671
Year	3.2229	3.3421
Number of nodes	5.8703	9.1857

Table 24
Standard deviation of variables by class for Pima Indians diabetes data set.

Variable	Class 1	Class 2
Number of times pregnan	3.0172	3.7412
Plasma glucose concentration	26.1412	31.9396
Diastolic blood pressure	26.1412	21.4918
Triceps skin fold thickness	14.8899	17.6797
2-Hour serum insulin	98.8653	138.6891
Body mass index	7.6899	7.2630
Diabetes pedigree function	0.2991	0.3724
Age	11.6677	10.9683

Table 25
Standard deviation of variables by class for Statlog vehicle data set.

Variable	Class 1	Class 2	Class 3	Class 4
Compactness	8.4359	9.4338	10.3694	3.8750
Circularity	5.1418	8.2558	7.3628	4.5274
Distance circularity	12.9375	17.6456	17.8308	11.9720
Radius ratio	30.1255	39.6570	33.7412	36.1658
Pr. axis aspect ratio	11.0849	5.8219	4.7002	13.7454
Max. length aspect ratio	8.9955	1.8209	1.9084	7.8904
Scatter ratio	36.2105	34.4818	32.7261	14.3834
Elongatedness	6.7937	8.5600	7.6933	4.8961
Pr. axis rectangularity	3.0584	2.5849	2.4836	1.0659
Max. length rectangularity	9.9169	19.8505	17.7096	11.7987
Scaled variance 1	36.2107	33.6397	29.6951	24.8049
Scaled variance 2	211.9640	180.1988	169.8642	58.0820
Scaled radius of gyration	30.6875	41.1769	39.4500	26.6110
Skewness about 1	11.9765	5.6052	6.2315	10.3686
Skewness about 2	3.1846	4.3516	5.1565	3.8973
Kurtosis about 1	6.2369	10.4398	7.6475	5.4185
Kurtosis about 2	7.7133	5.9848	5.8424	6.0299
Hollows ratio	8.0065	6.2078	7.0820	7.0722

Table 26
Standard deviation of variables by class for vertebral column data set.

Variable	Class 1	Class 2
Pelvic incidence	17.6618	12.3679
Pelvic tilt	10.5157	6.7787
Lumbar lordosis angle	19.6690	12.3616
Sacral slope	14.5151	9.6238
Pelvic radius	14.0910	9.0138
Degree spondylolisthesis	40.6967	6.3070

Table 27
Standard deviation of variables by class for wine data set.

Variable	Class 1	Class 2	Class 3
Alcohol	0.4621	0.5380	0.5302
Malic acid	0.6885	1.0156	1.0879
Ash	0.2272	0.3155	0.1847
Alcalinity of ash	2.5463	3.3498	2.2582
Magnesium	10.4989	16.7535	10.8905
Total phenols	0.3390	0.5454	0.3570
Flavanoids	0.3975	0.7057	0.2935
Nonflavanoid phenols	0.0700	0.1240	0.1241
Proanthocyanins	0.4121	0.6021	0.4088
Color intensity	1.2386	0.9249	2.3109
Hue	0.1165	0.2029	0.1144
OD280/OD315	0.3571	0.4966	0.2721
Proline	221.5208	157.2112	115.0970

Table 28
Standard deviation of variables by class for wine quality data set.

Variable	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Fixed acidity	1.7709	1.6266	1.5640	1.8010	1.9968	2.1743
Volatile acidity	0.3313	0.2201	0.1648	0.1609	0.1454	0.1453
Citric acid	0.2507	0.2010	0.1800	0.1956	0.1950	0.2035
Residual sugar	1.4016	1.7894	1.3598	1.4405	1.3686	1.2615
Chlorides	0.0662	0.0762	0.0537	0.0395	0.0294	0.0114
Free sulfur dioxide	9.7639	9.0259	10.9554	9.9357	10.1537	10.8674
Total sulfur dioxide	16.8289	27.5834	36.9931	25.0198	33.1185	24.9115
Density	2.8495	13.6026	11.4047	6.9807	9.9978	2.0677
pH	0.1441	0.1814	0.1506	0.1539	0.1498	0.1950
Sulphates	0.1220	0.2394	0.1711	0.1586	0.1358	0.1210
Alcohol	0.8180	0.9348	0.7365	1.0510	0.9699	1.3853

step needs cpn operations for matrix $\mathbf{U}$ and cp operations for matrix $\mathbf{\Lambda}$, thus the time complexity for initialization step is $O(cpn)$. The algorithm performs iteratively three main steps until the convergence. In the first one, prototypes are updated and the algorithm takes pn operations for each one of c prototypes, therefore the time complexity of this step is $O(cpn)$. In the second step, there are cpn membership values where each one takes cp operations, thus the time complexity of this step is $O(c^2p^2n)$. In the third one, c weight vectors are computed where each contains p weights that need pn operations, thus the time complexity of this step is $O(cp^2n)$. Therefore, the worst-case computational time complexity of the algorithm is $O(c^2p^2n)$.

Regarding the space complexity, the matrix of prototype $\mathbf{Y}$ requires p space for each one of c prototypes, thus it is necessary for $O(cp)$ space to store this matrix. For the matrix of membership $\mathbf{U}$, there are n matrices of multivariate membership matrices where each one requires cp space, thus the space complexity of matrix $\mathbf{U}$ is $O(cpn)$. The matrix of weight $\mathbf{\Lambda}$ is formed by c vectors of weights where each one requires p space, thus the final space complexity to store $\mathbf{\Lambda}$ is $O(cp)$. Therefore, the space complexity of this algorithm is $O(cpn)$.

3.2.2. Convergence of the algorithm

Following [38], properties of convergence of this kind of algorithm may be studied from two series: $\nu_t = (\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Lambda}^t) \in \mathbb{Y}^c \times \mathbb{U}^n \times \mathbb{L}^c$ and $\omega_t = J^3(\nu_t) = J^3(\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Lambda}^t)$, $t = 0, 1, 2, \ldots, T$. The algorithm starts from initial series $\nu_0 = (\mathbf{Y}^0, \mathbf{U}^0, \mathbf{\Lambda}^0)$ and computes the different terms of the series ν_t until the convergence when the criterion J^3 achieves a stationary value.

Proposition 9. *The series* $\omega_t = J^3(\nu_t)$ *decreases at each iteration and converges.*

Proof. The proof is given in Appendix D.

Proposition 10. *The series* $\nu_t = (\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Lambda}^t)$ *converges.*

Proof. The proof is given in Appendix E.

4. Partition and cluster interpretation

Indices interpretations are useful tools that allow a better understanding of the clustering results. For quantitative data partitioned by the standard hard c-means clustering algorithm, Celeux et al. [42] have introduced a family of indices for cluster interpretation that are based on the Sum of Squares (SSQ). In this section, we adapt these indices to the case of quantitative data partitioned by multivariate fuzzy c-means clustering algorithms presented in this paper (MFCM, WMFCM-M and WMFCM-D).

For the next equations, the index $l(l = 1, 2, 3)$ corresponds to a method, where $l=1$ is associated to the MFCM method, $l=2$ to WMFCM-M and $l=3$ to WMFCM-D.

4.1. *Measures based on the sum of squares*

In this section, we define the overall SSQ and SSQ within and between clusters for multivariate fuzzy c-means methods MFCM, WMFCM-M and WMFCM-D. Sum of squares may be decomposed into sum of squares by cluster and by variable. Moreover, these decompositions are basis for defining interpretation indices.

The overall heterogeneity of all n patterns is measured by the overall dispersion, which is computed according to the distance function given by Eq. (2). It is as follows:

$$T^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - z_j^1)^2$$

$$T^2 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - z_j^2)^2$$

$$T^3 = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - z_j^3)^2 \qquad (18)$$

For all three methods, T measures the overall heterogeneity of all patterns, that is, it measures how dispersed the patterns are with respect to the overall centroid $\mathbf{z}^l$.

Proposition 11. *The overall centroid vector* $\mathbf{z}^l = (z_1^l, \ldots, z_j^l, \ldots, z_p^l)$ *for all n data points, which minimizes the overall dispersion* T^l, *has its components* z_j^l *updated by*

$$z_j^1 = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ijk})^m x_{jk}}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ijk})^m}. \qquad (19)$$

$$z_j^2 = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m x_{jk}}{\sum_{i=1}^{c}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m}. \qquad (20)$$

$$z_j^3 = \frac{\sum_{i=1}^{c}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m x_{jk}}{\sum_{i=1}^{c}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m}. \qquad (21)$$

Proof. The proof can be obtained in a similar way as presented in Appendix A.

Since T^l is addictive, it may be decomposed into overall dispersion T_i^l for cluster C_i given by $T^l = \sum_{i=1}^{c} T_i^l (l = 1, 2, 3)$ with

$$T_i^1 = = \sum_{j=1}^{p}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^1)^2$$

$$T_i^2 = \sum_{j=1}^{p}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - z_j^2)^2$$

$$T_i^3 = \sum_{j=1}^{p}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^3)^2 \qquad (22)$$

A similar reasoning can be applied to the case where the overall dispersion is calculated according to variable j:

$$T_j^1 = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^1)^2$$

$$T_j^2 = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - z_j^2)^2$$

$$T_j^3 = \sum_{i=1}^{c}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^3)^2 \qquad (23)$$

Regarding cluster C_i and variable j, the overall dispersion for MFCM, WMFCM-M and WMFCM-D methods such that $T^l = \sum_{i=1}^{c} T_i^l = \sum_{j=1}^{p}(\sum_{i=1}^{c} T_{ij}^l)$ is given as

$$T_{ij}^1 = \sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^1)^2$$

$$T_{ij}^2 = \sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - z_j^2)^2$$

$$T_{ij}^3 = \lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - z_j^3)^2 \qquad (24)$$

The overall heterogeneity within-clusters fuzzy sum of squares measures the sum of dissimilarities between each object and prototypes. The more objects are similar to prototypes, the more clusters are concise. Many clustering methods aim reduce this measure. It is given as

$$J^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - y_{ij})^2$$

$$J^2 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - y_{ij})^2$$

$$J^3 = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - y_{ij})^2 \qquad (25)$$

The heterogeneity within cluster C_i is calculated as follows:

$$J_i^1 = \sum_{j=1}^{p}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - y_{ij})^2$$

$$J_i^2 = \sum_{j=1}^{p}\sum_{k=1}^{n}\left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - y_{ij})^2$$

$$J_i^3 = \sum_{j=1}^{p}\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - y_{ij})^2 \qquad (26)$$

Regarding variable j, the overall heterogeneity within-clusters is given as

$$J_j^1 = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ijk})^m (x_{jk} - y_{ij})^2$$

$$J_j^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - y_{ij})^2$$

$$J_j^3 = \sum_{i=1}^{c} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - y_{ij}^2 \tag{27}$$

In both cases, the heterogeneity within cluster C_i according to variable j where $J^l = \sum_{i=1}^{c} J_i^l = \sum_{j=1}^{p} (\sum_{i=1}^{c} J_{ij}^l)$ is as follows:

$$J_{ij}^1 = \sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - y_{ij})^2$$

$$J_{ij}^2 = \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (x_{jk} - y_{ij}))^2$$

$$J_{ij}^3 = \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - y_{ij})^2 \tag{28}$$

The between-cluster fuzzy sum of squares B^l measures the dispersion of the cluster prototypes and, consequently, the distinctness of all clusters. For this, the distance between prototypes and the overall centroid $\mathbf{z}^l$ is measured. According to the definition of clustering, the goal of clustering methods is to maximize this measure in order to find clusters such that objects in different groups are more dissimilar than in the same group. This measure is given as

$$B^1 = \sum_{i=1}^{c} \sum_{j=1}^{p} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^1)^2$$

$$B^2 = \sum_{i=1}^{c} \sum_{j=1}^{p} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (y_{ij} - z_j^2)^2$$

$$B^3 = \sum_{i=1}^{c} \sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^3)^2 \tag{29}$$

In relation to cluster C_i, the between-cluster fuzzy sum of squares is decomposed into the following measures:

$$B_i^1 = \sum_{j=1}^{p} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^1)^2$$

$$B_i^2 = \sum_{j=1}^{p} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (y_{ij} - z_j^2)^2$$

$$B_i^3 = \sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^3)^2 \tag{30}$$

Similarly, the dispersion of the cluster prototypes regarding the variable j can be calculated as

$$B_j^1 = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^1)^2$$

$$B_j^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (y_{ij} - z_j^2)^2$$

$$B_j^3 = \sum_{i=1}^{c} \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^3)^2 \tag{31}$$

Finally, the dispersion of the cluster prototypes regarding cluster C_i and variable j such that $B^l = \sum_{i=1}^{c} B_i^l = \sum_{j=1}^{p} (\sum_{i=1}^{c} B_{ij}^l)$ can be calculated as

$$B_{ij}^1 = \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^1)^2$$

$$B_{ij}^2 = \sum_{k=1}^{n} \left(\frac{u_{ijk}}{\gamma_{ijk}}\right)^m (y_{ij} - z_j^2)^2$$

$$B_{ij}^3 = \lambda_{ij} \sum_{k=1}^{n} (u_{ijk})^m (y_{ij} - z_j^3)^2 \tag{32}$$

Proposition 12. *The following relations hold for all* $i(i = 1, \ldots, c)$, $j(j = 1, \ldots, p)$ *and* $l = 1, 2, 3$:

$$T^l = J^l + B^l, T_i^l = J_i^l + B_i^l, T_j^l = J_j^l + B_j^l \text{ and } T_{ij}^l = J_{ij}^l + B_{ij}^l. \tag{33}$$

Proof. The proof is given in Appendix F.

4.2. *Interpretation indices*

Interpreting the overall quality of a partition after having applied a fuzzy clustering algorithm is an important problem in clustering analysis. In this section, we present a suitable adaptation of the indices presented in [42] to multivariate fuzzy c-means clustering algorithms.

4.2.1. *Partition interpretation indices*

The overall heterogeneity index R^l $(l = 1, 2, 3)$ is measured by the ratio among the between-cluster measure B^l and the overall dispersion T^l. It is given as

$$R^l = \frac{B^l}{T^l} \tag{34}$$

This index takes its values between 0 and 1. It is equal to 0 when the fuzzy partition is a single cluster or when fuzzy prototypes are equal to overall centroid. Whereas, it is equal to 1 when all fuzzy clusters have just a single pattern. A value of R^l closer to 0 means a poor representation of the elements of a fuzzy cluster C_i by its prototype. In this case, the clustering method could not find well-separated groups. Whereas, a value of R^l closer to 1 means more homogeneous clusters and a better representation of the elements of a fuzzy cluster C_i by its prototype.

Concerning single variable, the overall heterogeneity index that measures the proportion of overall dispersion and between-cluster index regarding variable j is defined as

$$COR^l(j) = \frac{B_j^l}{T_j^l} \tag{35}$$

This index takes also its values between 0 and 1, where a value of $COR^l(j)$ closer to 0 means that, according to variable j, prototypes are very similar to the overall centroid and denotes a partition of poor quality concerning the variable j. A value of $COR^l(j)$ closer to 1 denotes better quality of the partition concerning variable j.

The relative contribution of the variable j to the between-cluster measure B^l is given by

$$CTR^l(j) = \frac{B_j^l}{B^l} \tag{36}$$

A high value of $CTR^l(j)$ (closer to 1) indicates that the variable j provides an important contribution to the separation of the representatives of clusters. Whereas, values closer to 0 means that clusters are close to each other according to variable j.

4.2.2. *Cluster interpretation indices*

The analysis of the quality of an individual cluster is also an important problem after having applied a fuzzy clustering algorithm.

Regarding measures based on sum of square T^l, J^l and B^l and its contribution to cluster C_i (T_i^l, J_i^l and B_i^l), it is possible to calculate the relative contribution, respectively, as

$$T^l(i) = \frac{T_i^l}{T^l}$$

$$J^l(i) = \frac{J_i^l}{J^l}$$

$$B^l(i) = \frac{B_i^l}{B^l} \tag{37}$$

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

They take values between 0 and 1. High value of $T^l(i)$ indicates that the cluster C_i has a high contribution to overall heterogeneity. High value of $J^l(i)$ indicates that the cluster C_i is relatively heterogeneous in comparison with other clusters. Values closer to 1 of $B^l(i)$ indicate that the cluster C_i is relatively distant from the global center in comparison with other clusters.

Concerning single variable, the heterogeneity may be different for a cluster to another. Therefore, the previous overall heterogeneity index may be decomposed into the following expression:

$$COR^l(i,j) = \frac{B^l_{ij}}{T^l_j} \tag{38}$$

A high value of $COR^l(i,j)$ indicates that the variable j has a relatively homogeneous behavior within cluster C_i.

The relative contribution of variable j to the heterogeneity in the cluster C_i is given as

$$CTR^l(i,j) = \frac{B^l_{ij}}{B^l_i} \tag{39}$$

A high value of $CTR^l(i,j)$ indicates that the variable j provides an important contribution to the between-cluster measure of cluster C_i.

Finally, the relative contribution of variable j and cluster C_i to the between-cluster measure is calculated as

$$CE^l(i,j) = \frac{B^l_{ij}}{B^l} \tag{40}$$

Values closer to 1 of $CE^l(i,j)$ means that the variable j and cluster C_i provide an important contribution to the between-cluster measure. In other words, it is possible to verify which variable is contributing to the separation of cluster C_i among other clusters.

5. Experiments

In order to evaluate the performance of the proposed algorithms, a comparative study with five fuzzy clustering methods is carried out. The first method is the fuzzy clustering proposed by [28] (here called GK). The second method is the well-known FCM introduced by [24] which is able to identify spherical clusters since it uses the Euclidian distance. The third method was proposed by [31] (here called FCM-C^1) where there is a parameter which gives information about the influence of individual variables on the detected groups. The fourth method is a version of FCM using adaptive distances (here called FCM-C^2) proposed by [32]. This method assumes that clusters are not spherical. The fifth method is the multivariate fuzzy c-means (MFCM) proposed by [34]. The sixth and seventh methods are the versions of MFCM with weighting proposed in this paper.

The evaluation of clustering results furnished by each method is based on the computation of an external cluster validity index of fuzzy type proposed by [43] that follows the Rand index, proposed by [44] (here the fuzzy index is abbreviated by FR). The Rand index takes its values in the interval $[0,1]$ where the value 1 indicates a perfect agreement between partitions, values close to 0 correspond to insufficient recovery properties. FR according to fuzzy partitions P and Q is given as

$$FR(P,Q) = 1 - \frac{\sum_{k=1}^{n}\sum_{k'=1}^{n} |E_P(\mathbf{x}_k,\mathbf{x}_{k'}) - E_Q(\mathbf{x}_k,\mathbf{x}_{k'})|}{n(n-1)/2}, \tag{41}$$

where $E_P(\mathbf{x}_k,\mathbf{x}_{k'}) = 1 - \|\boldsymbol{\delta}_k - \boldsymbol{\delta}_{k'}\|$ and $\boldsymbol{\delta}_k = (\delta_{1k},\ldots,\delta_{ik},\ldots,\delta_{ck})$ is a vector of membership degrees by cluster of object $\mathbf{x}_k$ (similar definition $\boldsymbol{\delta}_{k'}$ with respect to object $\mathbf{x}_{k'}$ and $E_Q(\mathbf{x}_k,\mathbf{x}_{k'})$ to partition Q). In this paper, the metric $\|\cdot\|$ on $[0,1]^c$ that yields values in $[0,1]$ is given as $d(\boldsymbol{\delta}_k,\boldsymbol{\delta}_{k'}) = (1/c)\sum_{i=1}^{c}(\delta_{ik}-\delta_{ik'})^2$.

Syntectic data sets were created to evaluate the performance of methods in different configurations. Also, UCI machine learning repository data sets [45] were used to compare methods. For each UCI data set, partition and cluster indices were computed using clustering results of multivariate methods.

5.1. Synthetic data sets

In this study, four data sets in $\Re^2$ are considered. Each one contains three classes drawn according to a bi-variate normal distribution whose components are independent variables. The main difference between a configuration to another is the shape and volume of classes. The goal of use these data sets is to evaluate the methods through partitions with spherical or elliptical clusters. Tables 1 and 2 show parameters of classes for 1–4 point data sets.

According to parameters, each data set contains three classes based on the following descriptions:

- Data set 1: elliptical classes of different sizes.
- Data set 2: spherical classes of different sizes.
- Data set 3: elliptical classes of identical sizes.
- Data set 4: spherical classes of identical sizes.

Figs. 3 and 4 show the data sets.

5.1.1. Clustering quality results

In the framework of a Monte Carlo experiment, 100 replications of the previous process were carried out. For each replication, bivariate normal distributions following the parameters were randomly drawn 50 times and clustering methods were applied to data sets 1–4. The best result according to criterion function was selected and the FR index was calculated comparing 3-cluster partition obtained by each clustering method and the 3-class partition known a priori. After 100 replications, the mean and standard deviation of this index were calculated.

Table 3 shows values of the average and standard-deviation (in parentheses) of the FR index obtained by GK, FCM, FCM-C^1, FCM-C^2, MFCM, WMFCM-M and WMFCM-D with $m=2$ for data sets 1–4. The largest average value of FR index for each data set is in bold. Here, the FR index for multivariate methods (MFCM, WMFCM-M and WMFCM-D) is computed using memberships by cluster using the suitable aggregation function.

These results show clearly that the FR index for MFCM is greater than that for GK, FCM, FCM-C^1 and FCM-C^2 regarding data sets 1 and 3 with elliptical clusters. However, for data sets 1–4, WMFCM-M and WMFCM-D obtained better clustering quality than MFCM. WMFCM-M is superior to the others for data sets 1 (elliptical classes of different sizes) and 3 (elliptical classes of identical sizes). On the other hand, WMFCM-D is superior for data sets 2 (spherical classes of different sizes) and 4 (spherical classes of identical sizes).

In order to compare these methods, Student's t tests for independent samples with 5% of significance are performed. Tables 4 and 5 give the values of the test statistics following a Student's t distribution with 198 of freedom. In these tables, μ_1, μ_2, μ_3, μ_4, μ_5, μ_6 and μ_7 are, respectively, the average of the FR index for GK, FCM, FCM-C^1, FCM-C^2, MFCM, WMFCM-M and WMFCM-D.

Values in these tables support the hypothesis that MFCM was superior to GK, FCM, FCM-C^1 and FCM-C^2 for data sets 1 and 3 (with elliptical classes). Moreover, both WMFCM-M and WMFCM-D surpass MFCM for data sets 1–4. Concerning WMFCM-M, it outperforms all methods only for data sets 1 and 3. On the other hand, WMFCM-D outperforms all methods for data sets 2 and

4 and was better than the methods without multivariate membership (GK, FCM, FCM-C^{1}and FCM-C^2) for data sets 1 and 3.

In conclusion, as we know, the MFCM method is very suitable for configurations in which classes have variables with different dispersions. In fact, the use of the weighted versions of MFCM is necessary. WMFCM-D is the best option with spherical clusters and WMFCM-M with elliptical ones. Although MFCM is the worst option among all the methods for configurations of spherical clusters, its version weighting the variables based on their dispersion is essential to improve the clustering quality. Section 5.1.4 displays an evaluation of the methods with a mixture of elliptical and spherical clusters present in the data set.

5.1.2. Evaluating computational costs

In order to compare the performance of each method using the computational cost, a study was made concerning the number of iterations and the convergence time for each algorithm. Regarding time complexity, GK, FCM, FCM-C^1 and FCM-C^2 each take $O(c^2pn)$ operations. On the other hand, MFCM, WMFCM-M and WMFCM-D multivariate methods each take $O(c^2p^2n)$. Therefore, in terms of time complexity, methods based on multivariate memberships have a greater computational cost than methods with non-multivariate memberships.

The number of iterations and the time (in milliseconds) were noted for each algorithm until convergence. After 100 replications, the average and standard deviation of these measures were calculated. Tables 6 and 7 show values of average and standard deviation of the iteration number and time, respectively, regarding data sets 1–4. From results presented in these tables, we can conclude that the WMFCM-M method is the fastest one concerning both number of iterations and convergence time for multivariate methods. MFCM is the slowest method since it needs a greater number of iterations to converge. Regarding the WMFCM-D method, its computational cost can be considered intermediate: it is slower than WMFCM-M but faster than the MFCM method. In conclusion, multivariate fuzzy c-means methods WMFCM-M and WMFCM-D need a smaller number of iterations and less time to converge than MFCM for data sets 1–4.

5.1.3. Evaluating the parameter *m*

In order to evaluate the behavior of parameter *m* for the proposed methods, a Monte Carlo study was carried out using the following values for *m*: 1.1, 1.3, 1.5, 1.7, 2.0, 2.2, 2.5, 2.7, 3.0, 3.5, 4.0, 4.5 and 5.0. Fig. 5 shows averages of FR index for data sets 1 and 4.

From Fig. 5, it is possible to verify that, for values of *m* smaller than 2, WMFCM-M and WMFCM-D methods can obtain better results for both data sets. When *m* is approaching the value one, the method tends to be hard, and membership degrees tend to be 0 or 1. Thus the fuzzy matrix becomes more similar to the original hard partition. This is detected by the external index. On the other hand, for values of *m* larger than 2, the clustering quality of these methods is degraded and FR index decreases as the value of *m* increases. That is, when the fuzziness of the clustering method increases due to larger values of parameter *m*, the fuzzy matrix becomes more dissimilar than the hard partition. Furthermore, the clustering quality of the two methods for data set 4 tends to be similar for larger values of *m*.

5.1.4. Evaluating mixture of shapes

In Section 5.1.1 the study considers data sets with classes of same shape, that is, all classes are spherical or elliptical. Here, three data sets with 4 classes of equal cardinality were created. The goal of the study is to investigate the performance of methods with data sets containing classes of different shapes. Parameters of classes are presented in Tables 8 and 9. Figs. 6 and 7 show these data sets. From Table 10 we can obtain some important remarks:

1. The GK, FCM, FCM-C^1, FCM-C^2 and WMFCM-D methods improve their performance when the number of spherical clusters presents in these data sets increases. As expected, the MFCM has the worst performance when there are more spherical clusters than elliptical clusters (data set 7).
2. The FR index for WMFCM-M degrades when a number of spherical clusters present in the data set increases. However, this method is superior to the GK, FCM, FCM-C^1 FCM-C^2 and MFCM ones for data sets 5 and 6.
3. The use of weights is relevant to obtain better clustering quality since these methods surpass the methods without weighting for variables for all data sets (1–7). WMFCM-D should be preferred when there are more spherical clusters than elliptical ones.

5.2. UCI machine learning repository data sets

Fuzzy clustering methods were applied to 9 real data sets: Abalone, Ecoli, Glass Identification, Haberman's Survival, Pima Indian Diabetes, Statlog Vehicle, Vertebral Column, Wine and Wine Quality. These data sets may be found in the UCI machine learning repository [45]. Table 11 describes the number of classes, features and instances of each one of UCI data sets. In order to analyze the dispersion of the variables presented in these data sets, Appendix G displays tables with the standard deviation of each variable by cluster.

Each method is run until the convergence and the best result of the criterion function is selected among 200 repetitions. So, the FR index is calculated for each method comparing to the best partition obtained and the partition known a priori.

According to results shown in Table 12, it is possible to conclude that the WMFCM-D method was superior to the other ones for all real data sets of this study, except for Wine Quality. This is because there are homogeneous groups of variables regarding their dispersions by cluster in these data sets. Therefore, the use of multivariate memberships combined with weighted distances by cluster and variable (WMFCM-D) is necessary. We can conclude that multivariate memberships are necessary due to the intergroup heterogeneity of variables regarding the dispersion. On the other hand, weighted distances by cluster are necessary due to the intra-group homogeneity of variables. In addition, we can expect that if the number of these groups increases, the weighted multivariate memberships (WMFCM-M) should be preferred in relation to the multivariate memberships combined with weighted distances by cluster and variable (WMFCM-D).

Table 13 presents the convergence time (in milliseconds) for each clustering method and UCI data set.

5.3. Fuzzy partition and clusters interpretation: the Abalone data set

This section presents the usefulness of interpretation indices through the results obtained with application of the multivariate methods MFCM, WMFCM-M and WMFCM-D for the Abalone data set. Table 14 shows the overall heterogeneity index *R* for the three methods. These values are close to 1. Thus, for Abalone data set, the patterns belonging to clusters are well represented by the corresponding cluster prototypes.

A comparison between *COR*(*j*) and *CTR*(*j*) indices for each variable *j* of Abalone data set and for the MFCM, WMFCM-M and WMFCM-D methods is shown in Table 15. High values of *COR*(*j*) indicate a quality of the partition concerning variable *j*, while high values of *CTR*(*j*) indicate that the variable *j* provides an important contribution to the separation of the representatives of clusters. Concerning the *COR*(*j*) index for the MFCM and WMFCM-D methods, variables 4 (Whole weight) and 5 (Shucked weight) are not so useful to produce a high partition quality. On the other hand, for the WMFCM-M method, the variable 1 (Length), 2 (Diameter), 3 (Height) and 4 (Whole weight)

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

can produce good partition quality. Regarding *CTR*(*j*) index, variable 8 (Number of rings) has an important contribution to the separation of the representatives of clusters for MFCM method. On the other hand, whole weight and Height are the variables with the highest discriminating power for WMFCM-M and WMFCM-D, respectively.

Table 16 presents the $T^{(i)}, B^1(i)$ and $J^1(i)$ indices. High value of $T^1(i)$ indicates that the cluster C_i has a high contribution to overall heterogeneity. High value of $J^l(i)$ indicates that the cluster C_i is relatively heterogeneous in comparison with other clusters. Values closer to 1 of $B^l(i)$ indicate that the cluster C_i is relatively distant from the global center in comparison with other clusters. From these values, it is possible to conclude that cluster 2 is the closest to the global mean vector for MFCM and WMFCM-M, while for WMFCM-D method, is the cluster 3. For the three methods, the cluster 1 is the farthest away. For MFCM and WMFCM-M methods, the cluster 3 is the most homogeneous and, for WMFCM-D, the most homogeneous is the cluster 2.

Tables 17–19 show heterogeneity indices decomposed according to single variable and cluster for MFCM, WMFCM-M and WMFCM-D methods. A high value of $COR^l(i,j)$ indicates that the variable j has a relatively homogeneous behavior within cluster C_i. A high value of $CTR^{(i,j)}$ indicates that the variable j provides an important contribution to the between-cluster measure of cluster C_i. From Table 17, variables 7 (Shell weight), 1 (Length) and 2 (Diameter) have a relatively homogeneous behavior within clusters 1, 2 and 3, respectively. Variable 8 (Number of rings) is very important for separation of clusters.

According to Table 18, variables 2 (Diameter), 3 (Height) and 4 (Whole weight) are the highest contributors for the homogeneity within clusters 1, 2 and 3, respectively. Whereas, variable 2 (Diameter) contributes to separate cluster 1 and variable 4 (Whole weight) contributes to separate cluster 2 and 3.

Values in Table 19 indicate that variables 6 (Viscera weight), 1 (Length) and 3 (Height) present a relatively homogeneous behavior within clusters 1, 2 and 3, respectively. On the other hand, variable 2 (Diameter) is the most important one to allow the separation among clusters.

6. Conclusion

Two approaches for weighting a multivariate fuzzy c-means method were introduced. The first one uses weights in the membership, therefore objects may have different interpretations according to their position in the partition. The second one computes weights in the distance to quantify the dispersion of variables within clusters, allowing the algorithm to identify the shape of groups. Both types of weights may be different for each cluster and variable and change at each iteration of algorithm, minimizing the objective function. In addition, we proposed tools based on suitable dispersion measures for the interpretation of the fuzzy partition and fuzzy clusters obtained by multivariate fuzzy c-means methods.

In order to show the usefulness of the proposed methods in comparison with other fuzzy clustering methods, an experimental evaluation was carried out. Seven synthetic data sets varying the shape and dispersion structure for clusters were considered. The accuracy of the results furnished by clustering methods was assessed by the Fuzzy Rand Index. In addition, real data sets were also considered. In fact, the importance of weights for variables is highlighted from the results of clustering quality. An application of the multivariate fuzzy c-means methods to Abalone data set showed the merit of the fuzzy partition and cluster interpretation indices proposed in this paper.

Acknowledgment

The authors would like to thank CNPq and CAPES (Brazilian Agencies) for their financial support.

Appendix A. Proof of Proposition 6

Fixing the weight λ_{ij} and the membership u_{ijk}, the prototype y_{ij} that minimizes the criterion J^3 is updated using the following expression:

$$y_{ij}=\frac{\sum_{k=1}^{n}(u_{ijk})^m x_{jk}}{\sum_{k=1}^{n}(u_{ijk})^m}. \tag{A.1}$$

Proof. The criterion J^3 being additive, the problem becomes to minimize J^3_{ij} as follows. Let $d_{ijk}=(x_{jk}-y_{ij})^2$ be the distance between object $\mathbf{x}_k$ and prototype $\mathbf{y}_i$ for variable j:

$$J^3_{ij}=\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m(x_{jk}-y_{ij})^2. \tag{A.2}$$

J^3_{ij} becomes stationary when

$$\frac{\partial J^3_{ij}}{\partial y_{ij}}=-2\lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m(x_{jk}-y_{ij})=0 \tag{A.3}$$

That is

$$\sum_{k=1}^{n}(u_{ijk})^m x_{jk}-y_{ij}\sum_{k=1}^{n}(u_{ijk})^m=0 \tag{A.4}$$

From this, we can conclude

$$y_{ij}=\frac{\sum_{k=1}^{n}(u_{ijk})^m x_{jk}}{\sum_{k=1}^{n}(u_{ijk})^m}.\square \tag{A.5}$$

Appendix B. Proof of Proposition 7

Fixing the weight λ_{ij} and the prototype y_{ij}, the membership degree u_{ijk} that minimizes the criterion J^3 is updated using the following expression under the restriction $\sum_{i=1}^{c}\sum_{j=1}^{p}u_{ijk}=1$:

$$u_{ijk}=\left[\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{\lambda_{ij}d_{ijk}}{\lambda_{ab}d_{abk}}\right)^{1/(m-1)}\right]^{-1}. \tag{B.1}$$

Proof. The criterion J^3 being additive, the problem becomes to find for some object $\mathbf{x}_k$ the membership degree u_{ijk} that minimizes

$$J^3_k=\sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}(u_{ijk})^m d_{ijk}. \tag{B.2}$$

The method of Lagrange multipliers can be applied to find an optimum value to J^3_k:

$$F_k(\lambda)=\sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}(u_{ijk})^m d_{ijk}-\mu\left(\sum_{i=1}^{c}\sum_{j=1}^{p}u_{ijk}-1\right). \tag{B.3}$$

Thus, F_k is stationary in the following situations:

1. $\frac{\partial F_k}{\partial \mu}=\sum_{i=1}^{c}\sum_{j=1}^{p}u_{ijk}-1=0;$
2. $\frac{\partial F_k}{\partial u_{ijk}}=m(u_{ijk})^{m-1}\lambda_{ij}d_{ijk}-\mu=0.$

From the second item, it is possible to calculate the membership degree value u_{ijk} as follows:

$$u_{ijk}=\left(\frac{\mu}{m}\right)^{1/(m-1)}\left(\frac{1}{\lambda_{ij}d_{ijk}}\right)^{1/(m-1)}. \tag{B.4}$$

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

According to $\sum_{i=1}^{c}\sum_{j=1}^{p} u_{ijk}=1$ and Eq. (B.4), we have

$$\sum_{a=1}^{c}\sum_{b=1}^{p} u_{abk} = \left(\frac{\mu}{m}\right)^{1/(m-1)} \sum_{a=1}^{c}\sum_{b=1}^{p} \left(\frac{1}{\lambda_{ab} d_{abk}}\right)^{1/(m-1)} = 1. \tag{B.5}$$

Therefore, we obtain the following expression:

$$\left(\frac{\mu}{m}\right)^{1/(m-1)} = \frac{1}{\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{1}{\lambda_{ab} d_{abk}}\right)^{1/(m-1)}}. \tag{B.6}$$

Replacing Eq. (B.6) into Eq. (B.4), we have

$$u_{ijk} = \frac{1}{\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{1}{\lambda_{ab} d_{abk}}\right)^{1/(m-1)}} \left(\frac{1}{\lambda_{ij} d_{ijk}}\right)^{1/(m-1)}. \tag{B.7}$$

Finally, we obtain

$$u_{ijk} = \left[\sum_{a=1}^{c}\sum_{b=1}^{p}\left(\frac{\lambda_{ij} d_{ijk}}{\lambda_{ab} d_{abk}}\right)^{1/(m-1)}\right]^{-1}. \square \tag{B.8}$$

Appendix C. Proof of Proposition 8

Fixing the prototype y_{ij} and the membership u_{ijk}, the weight λ_{ij} that minimizes the criterion J^3 is updated using the following expression under the restriction $\prod_{h=1}^{p} \lambda_{ih} = 1$:

$$\lambda_{ij} = \frac{\left\{\prod_{h=1}^{p}\left[\sum_{k=1}^{n}(u_{ihk})^m d_{ihk}\right]\right\}^{1/p}}{\sum_{k=1}^{n}(u_{ijk})^m d_{ijk}}. \tag{C.1}$$

Proof. The weight may be calculated applying the method of Lagrange multipliers:

$$F_{ij}(\lambda) = \lambda_{ij}\sum_{k=1}^{n}(u_{ijk})^m d_{ijk} - \mu\left(\prod_{h=1}^{p}\lambda_{ih} - 1\right). \tag{C.2}$$

F_{ij} is stationary when

$$\frac{\partial F_{ij}}{\partial \lambda_{ij}} = \sum_{k=1}^{n}(u_{ijk})^m d_{ijk} - \mu\left(\frac{\prod_{h=1}^{p}\lambda_{ih}}{\lambda_{ij}}\right) = 0. \tag{C.3}$$

From Eq. (C.3), we obtain

$$\sum_{k=1}^{n}(u_{ijk})^m d_{ijk} = \mu\left(\frac{\prod_{h=1}^{p}\lambda_{ih}}{\lambda_{ij}}\right) \tag{C.4}$$

Since the restriction is $\prod_{h=1}^{p}\lambda_{ih} = 1$, then:

$$\lambda_{ij} = \frac{\mu}{\sum_{k=1}^{n}(u_{ijk})^m d_{ijk}}. \tag{C.5}$$

According to restriction and using Eq. (C.5), it is also true the following sentences:

$$\prod_{h=1}^{p}\lambda_{ih} = \prod_{h=1}^{p}\left[\frac{\mu}{\sum_{k=1}^{n}(u_{ihk})^m d_{ihk}}\right] = \frac{\mu^p}{\prod_{h=1}^{p}\left[\sum_{k=1}^{n}(u_{ihk})^m d_{ihk}\right]} = 1 \Rightarrow$$

$$\mu = \left\{\prod_{h=1}^{p}\left[\sum_{k=1}^{n}(u_{ihk})^m d_{ihk}\right]\right\}^{1/p}. \tag{C.6}$$

Replacing Eq. (C.6) into Eq. (C.5), we have

$$\lambda_{ij} = \frac{\left\{\prod_{h=1}^{p}\left[\sum_{k=1}^{n}(u_{ihk})^m d_{ihk}\right]\right\}^{1/p}}{\sum_{k=1}^{n}(u_{ijk})^m d_{ijk}}. \square \tag{C.7}$$

Appendix D. Proof of Proposition 9

The series $\omega_t = J^3(\nu_t)$ decreases at each iteration and converges.

Proof. According to algorithm, the following inequalities (I), (II) and (III) hold at each iteration:

$$J^3(\mathbf{Y}^t,\mathbf{U}^t,\mathbf{\Lambda}^t) \overbrace{\geq}^{(I)} J^3(\mathbf{Y}^{t+1},\mathbf{U}^t,\mathbf{\Lambda}^t) \overbrace{\geq}^{(II)} J^3(\mathbf{Y}^{t+1},\mathbf{U}^{t+1},\mathbf{\Lambda}^t) \overbrace{\geq}^{(III)} J^3(\mathbf{Y}^{t+1},\mathbf{U}^{t+1},\mathbf{\Lambda}^{t+1}). \tag{D.1}$$

Let $d(x_{jk}, y_{ij}^{(t)}) = (x_{jk} - y_{ij}^{(t)})^2$ be the distance between object x_{jk} and prototype y_{ij} in the iteration t. The inequality (I) holds because

$$J^3(\mathbf{Y}^t,\mathbf{U}^t,\mathbf{\Lambda}^t) = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}^{(t)}\sum_{k=1}^{n}\left(u_{ijk}^{(t)}\right)^m d\left(x_{jk}, y_{ij}^{(t)}\right),$$

$$J^3(\mathbf{Y}^{t+1},\mathbf{U}^t,\mathbf{\Lambda}^t) = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}^{(t)}\sum_{k=1}^{n}\left(u_{ijk}^{(t)}\right)^m d\left(x_{jk}, y_{ij}^{(t+1)}\right),$$

and according to Proposition 6,

$$y_{ij}^{(t+1)} = \underbrace{\arg\min}_{y \in \Re} \lambda_{ij}^{(t)}\sum_{k=1}^{n}\left(u_{ijk}^{(t)}\right)^m d(x_{jk}, y).$$

Moreover, inequality (II) also holds because

$$J^3(\mathbf{Y}^{t+1},\mathbf{U}^{t+1},\mathbf{\Lambda}^t) = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}^{(t)}\sum_{k=1}^{n}\left(u_{ijk}^{(t+1)}\right)^m d\left(x_{jk}, y_{ij}^{(t+1)}\right),$$

and according to Proposition 7,

$$u_{ijk}^{(t+1)} = \underbrace{\arg\min}_{u \in [0,1]} \lambda_{ij}^{(t)}\sum_{k=1}^{n}(u)^m d\left(x_{jk}, y_{ij}^{(t+1)}\right).$$

And inequality (III) also holds because

$$J^3\left(\mathbf{Y}^{t+1},\mathbf{U}^{t+1},\mathbf{\Lambda}^{t+1}\right) = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}^{(t+1)}\sum_{k=1}^{n}\left(u_{ijk}^{(t+1)}\right)^m d\left(x_{jk}, y_{ij}^{(t+1)}\right),$$

and according to Proposition 8,

$$\lambda_{ij}^{(t+1)} = \underbrace{\arg\min}_{\lambda \in \Re_+^*} \lambda\sum_{k=1}^{n}\left(u_{ijk}^{(t+1)}\right)^m d\left(x_{jk}, y_{ij}^{(t+1)}\right).$$

Finally, because the series ω_t decreases and it is bounded ($J^3(\nu_t) \geq 0$), it converges.□

Appendix E. Proof of Proposition 10

The series $\nu_t = (\mathbf{Y}^t, \mathbf{U}^t, \mathbf{\Lambda}^t)$ converges.

Proof. Assume that the stationarity of the series ν_t is achieved in the iteration $t = T$. Therefore, we have that $\nu_T = \nu_{T+1}$ and then $J^3(\nu_T) = J^3(\nu_{T+1})$, that is, $J^3(\mathbf{Y}^T,\mathbf{U}^T,\mathbf{\Lambda}^T) = J^3(\mathbf{Y}^{T+1},\mathbf{U}^{T+1},\mathbf{\Lambda}^{T+1})$. From this equality and Proposition 9, it is possible rewrite as equalities (I), (II) and (III):

$$J^3\left(\mathbf{Y}^T,\mathbf{U}^T,\mathbf{\Lambda}^T\right) \overbrace{=}^{(I)} J^3\left(\mathbf{Y}^{T+1},\mathbf{U}^T,\mathbf{\Lambda}^T\right) \overbrace{=}^{(II)} J^3\left(\mathbf{Y}^{T+1},\mathbf{U}^{T+1},\mathbf{\Lambda}^T\right)$$

$$\overbrace{=}^{(III)} J^3\left(\mathbf{Y}^{T+1},\mathbf{U}^{T+1},\mathbf{\Lambda}^{T+1}\right). \tag{E.1}$$

From the first equality (I), $\mathbf{Y}^T = \mathbf{Y}^{T+1}$ because $\mathbf{Y}$ is unique minimizing J^3 when $\mathbf{U}^T$ and $\mathbf{\Lambda}^T$ are fixed. From the second equality (II), $\mathbf{U}^T = \mathbf{U}^{T+1}$ because $\mathbf{U}$ is unique minimizing J^3 when $\mathbf{Y}^{T+1}$ and $\mathbf{\Lambda}^T$ are fixed. Moreover, from the third equality (III), $\Lambda^T = \mathbf{\Lambda}^{T+1}$ because $\mathbf{\Lambda}$ is unique minimizing J^3 when $\mathbf{Y}^{T+1}$ and $\mathbf{U}^{T+1}$ are fixed. Finally, we conclude that $\nu_T = \nu_{T+1}$. This conclusion holds for all

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

$t \geq T$ and $\nu_t = \nu_T, \forall t \geq T$ and it follows that the series ν_t converges.□

Appendix F. Proof of Proposition 12

The following relations hold for all $i(i=1,\dots,c), j(j=1,\dots,p)$ and $l=1,2,3$:

$$T^l = J^l + B^l, T_i^l = J_i^l + B_i^l, T_j^l = J_j^l + B_j^l \quad \text{and} \quad T_{ij}^l = J_{ij}^l + B_{ij}^l. \tag{F.1}$$

Proof. We will start showing that $T^1 = J^1 + B^1$ holds. Remembering:

$$T^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk} - z_j)^2. \tag{F.2}$$

The distance can be rewritten as

$$(x_{jk}-z_j)^2 = [(x_{jk}-y_{ij}) + (y_{ij}-z_j)]^2 = (x_{jk}-y_{ij})^2 + 2(x_{jk}-y_{ij})(y_{ij}-z_j) + (y_{ij}-z_j)^2 \tag{F.3}$$

Therefore we can obtain

$$T^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m [(x_{jk}-y_{ij})^2 + 2(x_{jk}-y_{ij})(y_{ij}-z_j) + (y_{ij}-z_j)^2] \tag{F.4}$$

That is

$$\begin{aligned} T^1 &= \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk}-y_{ij})^2 \\ &\quad + 2\sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk}-y_{ij})(y_{ij}-z_j) \\ &\quad + \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (y_{ij}-z_j)^2 \end{aligned} \tag{F.5}$$

Since $J^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk}-y_{ij})^2$ and $B^1 = \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (y_{ij}-z_j)^2$, thus we can replace them into Eq. (F.5):

$$T^1 = J^1 + 2\sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk}-y_{ij})(y_{ij}-z_j) + B^1 \tag{F.6}$$

We have $(x_{jk}-y_{ij})(y_{ij}-z_j) = y_{ij}(x_{jk}-y_{ij}) - z_j(x_{jk}-y_{ij})$. Thus

$$\begin{aligned} &\sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m (x_{jk}-y_{ij})(y_{ij}-z_j) \\ &= \sum_{i=1}^{c}\sum_{j=1}^{p}\sum_{k=1}^{n} (u_{ijk})^m \left[y_{ij}(x_{jk}-y_{ij}) - z_j(x_{jk}-y_{ij})\right] \\ &= \sum_{i=1}^{c}\sum_{j=1}^{p} \left\{ y_{ij}\left[\sum_{k=1}^{n} (u_{ijk})^m x_{jk} - y_{ij}\sum_{k=1}^{n} (u_{ijk})^m\right] \right. \\ &\quad \left. - z_j\left[\sum_{k=1}^{n} (u_{ijk})^m x_{jk} - y_{ij}\sum_{k=1}^{n} (u_{ijk})^m\right] \right\} \end{aligned} \tag{F.7}$$

From definition of prototype y_{ij}:

$$\begin{aligned} y_{ij} = \frac{\sum_{k=1}^{n} (u_{ijk})^m x_{jk}}{\sum_{k=1}^{n} (u_{ijk})^m} &\Rightarrow \sum_{k=1}^{n} (u_{ijk})^m x_{jk} = y_{ij}\sum_{k=1}^{n} (u_{ijk})^m \\ &\Rightarrow \sum_{k=1}^{n} (u_{ijk})^m x_{jk} - y_{ij}\sum_{k=1}^{n} (u_{ijk})^m = 0, \end{aligned} \tag{F.8}$$

the following expression can be obtained:

$$\begin{aligned} &\sum_{i=1}^{c}\sum_{j=1}^{p} \left\{ y_{ij}\left[\sum_{k=1}^{n} (u_{ijk})^m x_{jk} - y_{ij}\sum_{k=1}^{n} (u_{ijk})^m\right] \right. \\ &\quad \left. - z_j\left[\sum_{k=1}^{n} (u_{ijk})^m x_{jk} - y_{ij}\sum_{k=1}^{n} (u_{ijk})^m\right] \right\} = 0, \end{aligned} \tag{F.9}$$

which leads to the conclusion that $T^1 = J^1 + B^1$. The other expressions can be easily obtained in a similar way.□

Appendix G. Standard deviation of variables by class for UCI data sets

Tables 20–28 .

References

[1] J.R. Anderson, In: R.S. Michalski, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, 2, Morgan Kaufmann, 1986.

[2] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, 2011.

[3] Y. Anzai, Pattern Recognition & Machine Learning, Morgan Kaufmann, 1992.

[4] D.J. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[6] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, 2001.

[7] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, J. Liu, Particle competition and cooperation in networks for semi-supervised learning, IEEE Trans. Knowl. Data Eng. 24 (9) (2012) 1686–1698.

[8] G.P. Babu, M.N. Murty, Clustering with evolution strategies, Pattern Recognit. 27 (2) (1994) 321–329.

[9] L. Bai, J. Liang, C. Sui, C. Dang, Fast global k-means clustering based on local geometrical information, Inf. Sci. 245 (2013) 168–180.

[10] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley and Sons, New York, 2001 (Chapter 1).

[11] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (3) (1999) 264–323.

[12] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.

[13] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, 20, Siam, 2007.

[14] R.J. Kuo, S.S. Chen, W.C. Cheng, C.Y. Tsai, Integration of artificial immune network and K-means for cluster analysis, Knowl. Inf. Syst. (2013) 1–17.

[15] G.J. Szekely, M.L. Rizzo, Hierarchical clustering via joint between-within distances: extending ward's minimum variance method, J. Class. 22 (2) (2005) 151–183.

[16] A. Krishnamurthy, S. Balakrishnan, M. Xu, A. Singh, Efficient active algorithms for hierarchical clustering, In: Proceedings of the 29th International Conference on Machine Learning, 2012.

[17] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (2) (1985) 159–179.

[18] C. Fraley, A.E. Raftery, How many clusters? Which clustering method? Answers via model-based cluster analysis, Comput. J. 41 (8) (1998) 578–588.

[19] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.P.L.F. De Carvalho, A survey of evolutionary algorithms for clustering, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 39 (2) (2009) 133–155.

[20] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, IEEE Trans. Fuzzy Syst. 13 (4) (2005) 517–530.

[21] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.

[22] J. MacQueen, Some methods for classification and analysis of multivariate observations, In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 14, 1967, pp. 281–297.

[23] S.A. Elavarasi, J. Akilandeswari, B. Sathiyabhama, A survey on partition clustering algorithms, Int. J. Enterp. Comput. Bus. Syst. 1 (1) (2011).

[24] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[25] M.S. Yang, A survey of fuzzy clustering, Math. Comput. Model. 18 (11) (1993) 1–16.

[26] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (2) (1993) 98–110.

[27] N.R. Pal, K. Sarkar, What and when can we gain from the kernel versions of c-means algorithm, IEEE Trans. Fuzzy Syst. (2013).

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

[29] D.Q. Zhang, S.C. Chen, Kernel-based fuzzy and possibilistic c-means clustering, In: Proceedings of the International Conference on Artificial Neural Network, 2003 (June), pp. 122–125.

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011

[30] N.R. Pal, K. Pal, J.C. Bezdek, A mixed c-means clustering model, In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, vol. 1, 1997, pp. 11–21.

[31] A. Keller, F. Klawonn, Fuzzy clustering with weighting of data variables, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 8 (6) (2000) 735–746.

[32] F.A.T. de Carvalho, C.P. Tenorio, N.L.J. Cavalcanti, Partitional fuzzy clustering methods based on adaptive quadratic distances, Fuzzy Sets Syst. 157 (21) (2006) 2833–2857.

[33] M.R. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets Syst. 237 (2013) 1–46.

[34] B.A. Pimentel, R.M.C.R. Souza, A Multivariate Fuzzy C-Means Method, Appl. Soft Comput. 13 (4) (2013) 1592–1607.

[35] B.A. Pimentel, R.M.C.R. Souza, A weighted multivariate fuzzy c-means method in interval-valued scientific production data, Exp. Syst. Appl. 41 (7) (2014) 3223–3236.

[36] E. Diday, G. Govaert, Classification automatique avec distances adaptatives, RAIRO Inform. Comput. Sci. 11 (4) (1976) 329–349.

[37] R.M. Souza, F.A.T. de Carvalho, Clustering of interval data based on city-block distances, Pattern Recognit. Lett. 25 (3) (2004) 353–365.

[38] E. Diday, J.C. Simon, Clustering analysis, In: K.S. Fu (Ed.), Digital Pattern Classification, Springer, Berlin, 1976, pp. 47–94.

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

[40] H. Frigui, O. Nasraoui, Unsupervised learning of prototypes and attribute weights, Pattern Recognit. 37 (3) (2004) 567–581.

[41] A.F. Alvim, R.C. Souza, A fuzzy weighted clustering method for symbolic interval data, In: 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), 2012, pp. 1–6.

[42] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, H. Ralambondrainy, Classification Automatique des Donnees, Bordas, Paris, 1989.

[43] E. Hullermeier, M. Rifqi, S. Henzgen, R. Senge, Comparing fuzzy partitions: a generalization of the Rand index and related measures, IEEE Trans. Fuzzy Syst. 20 (3) (2012) 546–556.

[44] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.

[45] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2013, 2013. Accessible: ⟨http://archive.ics.uci.edu/ml⟩.

Bruno Almeida Pimentel received the M.Sc. degree in Computer Science from Federal University of Pernambuco, in 2013 in the Computational Intelligence area. He is currently a Ph.D. student in Computational Intelligence at Center of Informatics at Federal University of Pernambuco, Brazil. She has published 12 articles in scientific journals and conferences. His research interests include Fuzzy Clustering, Pattern Recognition and Cluster Analysis.

Renata Maria Cardoso Rodrigues de Souza received M.Sc. degree in Statistics in 1999 and the Ph.D. degree in Computer Science in 2003 from Center of Informatics at Federal University of Pernambuco, Brazil. She joined the Center of Informatics in 2005, where she is currently an Associate Professor. She has been a fellowship researcher of the CNPq (Brazilian Agency) since 2010. She has published 86 articles in scientific journals and conferences. Her research interests include Symbolic Data Analysis, Clustering, Classification, Statistical Pattern Recognition, Image Analysis, Information Retrieval.

Please cite this article as: B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.011