

Relational partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices

Francisco de A.T. de Carvalho^{a,*}, Yves Lechevallier^b, Filipe M. de Melo^a

^a Centro de Informática, Universidade Federal de Pernambuco, Av. Prof. Luiz Freire, s/n - Cidade Universitária, CEP 50740-540, Recife (PE), Brazil

^b INRIA-Institut National de Recherche en Informatique et en Automatique Domaine de Voluceau-Rocquencourt B.P. 105, 78153 Le Chesnay Cedex, France

Received 12 October 2010; received in revised form 14 August 2012; accepted 23 September 2012

Available online 2 October 2012

Abstract

This paper introduces fuzzy clustering algorithms that can partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. The aim is to obtain a collaborative role of the different dissimilarity matrices to get a final consensus partition. These matrices can be obtained using different sets of variables and dissimilarity functions. These algorithms are designed to furnish a partition and a prototype for each fuzzy cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fit between the fuzzy clusters and their representatives. These relevance weights change at each algorithm iteration and can either be the same for all fuzzy clusters or different from one fuzzy cluster to another. Experiments with real-valued data sets from the UCI Machine Learning Repository as well as with interval-valued and histogram-valued data sets show the usefulness of the proposed fuzzy clustering algorithms.

© 2012 Elsevier B.V. All rights reserved.

Keywords: Fuzzy clustering; Fuzzy medoids; Relational data; Collaborative clustering; Multiple dissimilarity matrices; Relevance weight

1. Introduction

Clustering is a method of unsupervised learning and is applied in various fields, including data mining, pattern recognition, computer vision and bioinformatics. The aim is to organize a set of items into clusters such that items within a given cluster have a high degree of similarity, while items belonging to different clusters have a high degree of dissimilarity. Hierarchical and partitioning methods are the most popular clustering techniques [1,2]. Hierarchical methods yield a complete hierarchy, i.e., a nested sequence of partitions of the input data, whereas partitioning methods seek to obtain a single partition of the input data into a fixed number of clusters, usually by optimizing an objective function.

Partitioning clustering can also be divided into hard and fuzzy methods. In hard partitioning clustering methods, each object of the data set must be assigned to precisely one cluster. Fuzzy partitioning clustering [3], on the other hand,

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.

E-mail addresses: francisco.carvalho@pq.cnpq.br, fatc@cin.ufpe.br (F.d.A.T. de Carvalho), Yves.Lechevallier@inria.fr (Y. Lechevallier), fmm@cin.ufpe.br (F.M. de Melo).

furnishes a fuzzy partition based on the idea of the partial membership of each pattern in a given cluster. This allows flexibility to express that objects belong to more than one cluster at the same time [4].

There are two common representations of the objects upon which clustering can be based: usual or symbolic feature data and relational data. When each object is described by a vector of quantitative or qualitative values, the set of vectors describing the objects is called a feature data set. When each complex object is described by a vector of sets of categories, intervals, or weight histograms, the set of vectors describing the objects is called a symbolic feature data set. Symbolic data have been mainly studied in symbolic data analysis (SDA) [5–8]. Alternatively, when each pair of objects is represented by a relationship, we have relational data. The most common case of relational data is when we have (a matrix of) dissimilarity data, say $R = [r_{kl}]$, where r_{kl} is the pairwise dissimilarity (often a distance) between objects k and l .

This paper introduces fuzzy clustering algorithms to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. The main idea is to obtain a collaborative role of the different dissimilarity matrices [9] to get a final consensus partition [10]. These dissimilarity matrices can be generated using different sets of variables and a fixed dissimilarity function (in this case, the final fuzzy partition gives a consensus between different views, i.e., between different variables, describing the objects), or using a fixed set of variables and different dissimilarity functions (in this case, the final fuzzy partition gives the consensus between different dissimilarity functions) or even using different sets of variables and dissimilarity functions.

As pointed out by Frigui et al. [11], the influence of the different dissimilarity matrices is not equally important in the definition of the fuzzy clusters in the final fuzzy partition. Thus, to obtain a meaningful fuzzy partition from all dissimilarity matrices, it is necessary to learn relevance weights for each dissimilarity matrix. Frigui et al. [11] proposed CARD, a fuzzy clustering algorithm that can partition objects taking into account multiple dissimilarity matrices and that learns a relevance weight for each dissimilarity matrix in each cluster. CARD is based mainly on the well-known fuzzy clustering algorithms for relational data, NERF [12] and FANNY [4].

The relational fuzzy clustering algorithms given in this paper are designed to give a fuzzy partition and a prototype for each cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fit between the fuzzy clusters and their representatives. These relevance weights change at each algorithm's iteration and can either be the same for all clusters or different from one cluster to another. Moreover, the fuzzy clustering algorithms proposed in this paper are mainly related to the fuzzy k-medoids algorithms [13]. Refs. [14,15] give a hard version of the fuzzy k-medoids algorithms. The approaches to compute the relevance weights of the dissimilarity matrices are inspired from both the computation of the membership degree of an object belonging to a fuzzy cluster [3] and the computation of a relevance weight for each variable in each cluster in the framework of the dynamic clustering algorithm based on adaptive distances [16].

Several applications can benefit from relational clustering algorithms based on multiple dissimilarity matrices. In image data base categorization, the relationship among the objects may be described by multiple dissimilarity matrices, and the most effective dissimilarity measures do not have a closed form or are not differentiable with respect to prototype parameters [11]. In SDA [5–8], many suitable dissimilarity measures [17,18] are not differentiable with respect to prototype parameters and also cannot be used in object-based clustering. Another issue is the clustering of mixed-feature data, where the objects are described by a vector of quantitative, qualitative, or binary values, or the clustering of mixed-feature symbolic data, where the objects are described by a vector of a set of categories, intervals, or histograms.

This paper is organized as follows. Section 2 first gives a partitioning fuzzy clustering algorithm for relational data based on a single dissimilarity matrix (Section 2.1) and then introduces partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices with relevance weight for each dissimilarity matrix estimated either locally (Section 2.2.2) or globally (Section 2.2.3). Section 3 gives empirical results to show the usefulness of these relational clustering algorithms based on multiple dissimilarity matrices. Finally, Section 4 gives the final remarks and comments.

2. Partitioning fuzzy K-medoids clustering algorithms based on multiple dissimilarity matrices

In this section, we introduce partitioning fuzzy clustering algorithms for relational data that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices.

2.1. Partitioning fuzzy K-medoids clustering algorithm based on a single dissimilarity matrix

There are some relational clustering algorithms in the literature, such as SAHN (sequential agglomerative hierarchical non-overlapping) [2] and PAM (partitioning around medoids) [4]. However, we start with the introduction of a partitioning fuzzy clustering algorithm for relational data based on a single dissimilarity matrix, because the algorithms based on multiple dissimilarity matrices given in this paper are based on it. This partitioning fuzzy clustering algorithm based on a single dissimilarity matrix is mainly related to the fuzzy k-medoids algorithms [13].

Let $E = \{e_1, \dots, e_n\}$ be a set of n objects and let a dissimilarity matrix $\mathbf{D} = [d(e_i, e_l)]$, where $d(e_i, e_l)$ measures the dissimilarity between objects e_i and e_l ($i, l = 1, \dots, n$). A particularity of this partitioning fuzzy clustering algorithm is that it assumes that the prototype G_k of fuzzy cluster C_k is a subset of fixed cardinality $1 \leq q \leq n$ of the set of objects E , i.e., $G_k \in E^{(q)} = \{A \subset E : |A| = q\}$.

The partitioning relational fuzzy clustering algorithm presented hereafter optimizes an adequacy criterion J that is defined as

$$J = \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m D(e_i, G_k) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d(e_i, e) \quad (1)$$

where $J_k = \sum_{i=1}^n (u_{ik})^m D(e_i, G_k)$ is the homogeneity in fuzzy cluster C_k ($k = 1, \dots, K$), and

$$D(e_i, G_k) = \sum_{e \in G_k} d(e_i, e) \quad (2)$$

measures the matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$, u_{ik} is the membership degree of object e_i in fuzzy cluster C_k , and $m \in (1, \infty)$ is a parameter that controls the fuzziness of membership for each object e_i .

The adequacy criterion measures the homogeneity of the fuzzy partition P as the sum of the homogeneities in each fuzzy cluster. This relational fuzzy clustering algorithm looks for a fuzzy partition $P = (C_1, \dots, C_K)$ of E into K fuzzy clusters represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, with $\mathbf{u}_i = (u_{i1}, \dots, u_{iK})$ ($i = 1, \dots, n$), and the corresponding vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ representing the fuzzy clusters in P such that the adequacy criterion (objective function) J measuring the fit between the fuzzy clusters and their prototypes is (locally) optimized. The algorithm sets an initial fuzzy partition and alternates two steps until convergence, when the criterion J reaches a stationary value representing a local minimum.

Step 1: Computation of the best prototypes. In this step, the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ is fixed.

Proposition 2.1. *The prototype $G_k = G^* \in E^{(q)}$ of fuzzy cluster C_k ($k = 1, \dots, K$), which minimizes the clustering criterion J , is such that $\sum_{i=1}^n (u_{ik})^m D(e_i, G^*) \rightarrow \text{Min}$. The prototype G_k ($k = 1, \dots, K$) is computed according to the following procedure:*

$G^* \leftarrow \emptyset$

REPEAT

Find $e_l \in E, e_l \notin G^*$ such that $l = \text{argmin}_{1 \leq h \leq n} \sum_{i=1}^n (u_{ih})^m d(e_i, e_h)$

$G^* \leftarrow G^* \cup \{e_l\}$

UNTIL $|G^*| = q$

Proof. The proof of Proposition 2.1 is straightforward. \square

Step 2: Definition of the best fuzzy partition.

In this step, the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ is fixed.

Proposition 2.2. *The fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, where $\mathbf{u}_i = (u_{i1}, \dots, u_{iK})$ ($i = 1, \dots, n$), which minimizes the clustering criterion J , is such that the membership degree u_{ik} ($i = 1, \dots, n; k = 1, \dots, K$) of*

each pattern i in each fuzzy cluster C_k , under $u_{ik} \in [0, 1]$ and $\sum_{k=1}^K u_{ik} = 1$, is calculated according to the following expression:

$$u_{ik} = \left[\sum_{h=1}^K \left(\frac{D(e_i, G_k)}{D(e_i, G_h)} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{e \in G_k} d(e_i, e)}{\sum_{e \in G_h} d(e_i, e)} \right)^{1/(m-1)} \right]^{-1} \quad (3)$$

Proof. The proof follows the same schema as that developed in the classical fuzzy K -means algorithm [3]. \square

Algorithm. The partitioning fuzzy K-medoids clustering algorithm based on a single dissimilarity matrix (denoted here as $SFCMdd$) sets an initial fuzzy partition and alternates two steps until convergence, when the criterion J reaches a stationary value representing a local minimum. This algorithm is summarized below.

Partitioning Fuzzy K-Medoids Clustering Algorithm Based on a Single Dissimilarity Matrix

(1) Initialization.

Fix K (the number of clusters), $2 \leq K \ll n$; fix m , $1 < m < +\infty$; fix T (an iteration limit); fix $\varepsilon > 0$ and $\varepsilon \ll 1$;

Fix the cardinality $1 \leq q \ll n$ of the prototypes G_k ($k = 1, \dots, K$);

Set $t = 0$;

Randomly select K distinct prototypes $G_k^{(0)} \in E^{(q)}$ ($k = 1, \dots, K$);

For each object e_i ($i = 1, \dots, n$) compute its membership degree $u_{ik}^{(0)}$ ($k = 1, \dots, K$) on fuzzy cluster C_k :

$$u_{ik}^{(0)} = \left[\sum_{h=1}^K \left(\frac{D(e_i, G_k^{(0)})}{D(e_i, G_h^{(0)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{e \in G_k^{(0)}} d(e_i, e)}{\sum_{e \in G_h^{(0)}} d(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

Compute:

$$J^{(0)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m D(e_i, G_k^{(0)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m \sum_{e \in G_k^{(0)}} d(e_i, e)$$

(2) Step 1: computation of the best prototypes.

Set $t = t + 1$.

The fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ is fixed.

Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of fuzzy cluster $C_k^{(t-1)}$ ($k = 1, \dots, K$) according to the procedure described in Proposition 2.1.

(3) Step 2: definition of the best fuzzy partition.

The vector of prototypes $\mathbf{G}^{(t)} = (G_1^{(t)}, \dots, G_K^{(t)})$ is fixed.

Compute the membership degree $u_{ik}^{(t)}$ of object e_i ($i = 1, \dots, n$) in fuzzy cluster C_k ($k = 1, \dots, K$), according to

$$u_{ik}^{(t)} = \left[\sum_{h=1}^K \left(\frac{D(e_i, G_k^{(t)})}{D(e_i, G_h^{(t)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{e \in G_k^{(t)}} d(e_i, e)}{\sum_{e \in G_h^{(t)}} d(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

(4) Stopping criterion.

Compute:

$$J^{(t)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D(e_i, G_k^{(t)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m \sum_{e \in G_k^{(t)}} d(e_i, e)$$

If $|J^{(t)} - J^{(t-1)}| \leq \varepsilon$ or $t > T$: STOP; otherwise go to 2 (Step 1).

2.2. Partitioning fuzzy K-medoids clustering algorithms based on multiple dissimilarity matrices

This section presents partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices. These algorithms are mainly related to the fuzzy k-medoids algorithms [13]. The approaches to compute the relevance weights of the dissimilarity matrices are inspired from both the computation of the membership degree of an object belonging to a fuzzy cluster [3] and the computation of a relevance weight for each variable in each cluster in the framework of the dynamic clustering algorithm based on adaptive distances [16].

2.2.1. Partitioning fuzzy K-medoids clustering algorithm based on multiple dissimilarity matrices

Let $E = \{e_1, \dots, e_n\}$ be the set of n objects and let p dissimilarity matrices $\mathbf{D}_j = [d_j(e_i, e_l)](j = 1, \dots, p)$, where $d_j(e_i, e_l)$ gives the dissimilarity between objects e_i and e_l ($i, l = 1, \dots, n$) on dissimilarity matrix \mathbf{D}_j . Assume that the prototype G_k of cluster C_k is a subset of fixed cardinality $1 \leq q \leq n$ of the set of objects E , i.e., $G_k \in E^{(q)} = \{A \subset E : |A| = q\}$.

The partitioning fuzzy K-medoids clustering algorithm introduced in Section 2.1 can take into account these p dissimilarity matrices \mathbf{D}_j if its adequacy criterion becomes

$$\begin{aligned} J &= \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m D(e_i, G_k) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p D_j(e_i, G_k) \\ &= \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p \sum_{e \in G_k} d_j(e_i, e) \end{aligned} \quad (4)$$

in which

$$D(e_i, G_k) = \sum_{j=1}^p D_j(e_i, G_k) = \sum_{j=1}^p \sum_{e \in G_k} d_j(e_i, e) \quad (5)$$

measures the global matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$, and $D_j(e_i, G_k)$ measures the local matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$ on dissimilarity matrix \mathbf{D}_j ($j = 1, \dots, p$).

The algorithm *SFCMdd* is modified into *MFCMdd* (partitioning fuzzy K-medoids clustering algorithm based on multiple dissimilarity matrices), so that in *Step 1* the prototype $G_k = G^* \in E^{(q)}$ of fuzzy cluster C_k ($k = 1, \dots, K$) is such that $\sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p D_j(e_i, G^*) \rightarrow \text{Min}$ and it is computed according to the following procedure:

$G^* \leftarrow \emptyset$

REPEAT

Find $e_l \in E, e_l \notin G^*$ such that $l = \text{argmin}_{1 \leq h \leq n} \sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p d(e_i, e_h)$

$G^* \leftarrow G^* \cup \{e_l\}$

UNTIL $|G^*| = q$

In *Step 2* the membership degree of object e_i in fuzzy cluster C_k is such that

$$u_{ik} = \left[\sum_{h=1}^K \left(\frac{D(e_i, G_k)}{D(e_i, G_h)} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p \sum_{e \in G_k} d_j(e_i, e)}{\sum_{j=1}^p \sum_{e \in G_h} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

This approach is similar to that which consists in clustering the set of objects E based on a global dissimilarity matrix $\mathbf{D} = [d(e_i, e_l)]$, with $\mathbf{D} = \sum_{j=1}^p \mathbf{D}_j$ and $d(e_i, e_l) = \sum_{j=1}^p d_j(e_i, e_l)$ ($i, l = 1, \dots, n$), which gives the same weight to the p partial dissimilarity matrices. As pointed out in [11], this later approach may not be effective since the influence of each partial dissimilarity matrix may be not equally important in order to define the cluster to which similar objects belong.

2.2.2. Partitioning fuzzy K -medoids clustering algorithms with relevance weight for each dissimilarity matrix estimated locally

This algorithm is designed to give a fuzzy partition and a prototype for each fuzzy cluster as well as to learn a relevance weight for each dissimilarity matrix that changes at each algorithm's iteration and is different from one fuzzy cluster to another.

The partitioning fuzzy clustering algorithm with relevance weight for each dissimilarity matrix estimated locally looks for a fuzzy partition $P = (C_1, \dots, C_K)$ of E into K fuzzy clusters represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, a corresponding K -dimensional vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ representing the fuzzy clusters in fuzzy partition P , and a K -dimensional vector of relevance weight vectors (one for each fuzzy cluster) $\mathbf{\Lambda} = (\lambda_1, \dots, \lambda_K)$, such that an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes is (locally) optimized. The adequacy criterion is defined as

$$J = \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m D_{(\lambda_k, s)}(e_i, G_k) \quad (6)$$

in which $D_{(\lambda_k, s)}$ is the global matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$, parameterized by $1 \leq s < \infty$ and by the relevance weight vector $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ of the dissimilarity matrices \mathbf{D}_j ($j = 1, \dots, p$) into cluster C_k ($k = 1, \dots, K$).

Two matching functions with relevance weight for each dissimilarity matrix estimated locally are considered depending on whether the sum of weights is equal to one (inspired from the computation of the membership degree of an object belonging to a fuzzy cluster [3]) or whether the product of the weights is equal to one (inspired from the computation of a relevance weight for each variable in each cluster in the framework of the dynamic clustering algorithm based on adaptive distances [16]). These matching functions are as follows:

- (a) Matching function parameterized by both the parameter s and the vector of relevance weights $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$, in which $s = 1$, $\lambda_{kj} > 0$ and $\prod_{j=1}^p \lambda_{kj} = 1$, and associated with cluster C_k ($k = 1, \dots, K$)

$$D_{(\lambda_k, s)}(e_i, G_k) = \sum_{j=1}^p (\lambda_{kj})^s D_j(e_i, G_k) = \sum_{j=1}^p \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) \quad (7)$$

- (b) Matching function parameterized by both the parameter s and the vector of relevance weights $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$, in which $1 < s < \infty$, $\lambda_{kj} \in [0, 1]$ and $\sum_{j=1}^p \lambda_{kj} = 1$, and associated with cluster C_k ($k = 1, \dots, K$)

$$D_{(\lambda_k, s)}(e_i, G_k) = \sum_{j=1}^p (\lambda_{kj})^s D_j(e_i, G_k) = \sum_{j=1}^p (\lambda_{kj})^s \sum_{e \in G_k} d_j(e_i, e) \quad (8)$$

In Eqs. (7) and (8), $D_j(e_i, G_k) = \sum_{e \in G_k} d_j(e_i, e)$ is the local dissimilarity between an example $e_i \in C_k$ and the cluster prototype computed on dissimilarity matrix \mathbf{D}_j ($j = 1, \dots, p$).

Note that this clustering algorithm assumes that the prototype of each cluster is a subset (of fixed cardinality) of the set of objects. Moreover, the relevance weight vectors λ_k ($k = 1, \dots, K$) are estimated locally and change at each iteration, i.e., they are not determined absolutely, and are different from one cluster to another.

Note also that when the product of the weights is equal to one, each relevant dissimilarity matrix in the clusters presents a weight superior to 1, whereas when the sum of weights is equal to one, each relevant dissimilarity matrix presents a weight superior to $1/p$.

This clustering algorithm sets an initial fuzzy partition and alternates three steps until convergence, when the criterion J reaches a stationary value representing a local minimum.

Step 1: Computation of the best prototypes.

In this step, the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the vector of relevance weight vectors $\mathbf{\Lambda} = (\lambda_1, \dots, \lambda_K)$ are fixed.

Proposition 2.3. *The prototype $G_k = G^* \in E^{(q)}$ of fuzzy cluster C_k ($k = 1, \dots, K$), which minimizes the clustering criterion J , is such that $\sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p (\lambda_{kj})^s D_j(e_i, G^*) \rightarrow \text{Min}$. The prototype G_k ($k = 1, \dots, K$) is computed*

according to the following procedure:

$G^* \leftarrow \emptyset$

REPEAT

Find $e_l \in E, e_l \notin G^*$ such that $l = \operatorname{argmin}_{1 \leq h \leq n} \sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p (\lambda_{kj})^s d(e_i, e_h)$

$G^* \leftarrow G^* \cup \{e_l\}$

UNTIL $|G^*| = q$

Proof. The proof of Proposition 2.3 is straightforward. \square

Step 2: Computation of the best relevance weight vector.

In this step, the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ are fixed.

Proposition 2.4. The vectors of weights are computed according to the matching function used:

- (1) If the matching function is given by Eq. (7), the vectors of weights $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ ($k = 1, \dots, K$), under $\lambda_{kj} > 0$ and $\prod_{j=1}^p \lambda_{kj} = 1$, have their weights λ_{kj} ($j = 1, \dots, p$) calculated according to

$$\lambda_{kj} = \frac{\left\{ \prod_{h=1}^p \left[\sum_{i=1}^n (u_{ik})^m D_h(e_i, G_k) \right] \right\}^{1/p}}{\left[\sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) \right]} = \frac{\left\{ \prod_{h=1}^p \left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e) \right] \right\}^{1/p}}{\left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) \right]} \quad (9)$$

- (2) If the matching function is given by Eq. (8), the vectors of weights $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ ($k = 1, \dots, K$), under $\lambda_{kj} \in [0, 1]$ and $\sum_{j=1}^p \lambda_{kj} = 1$, have their weights λ_{kj} ($j = 1, \dots, p$) calculated according to

$$\lambda_{kj} = \left[\sum_{h=1}^p \left(\frac{\sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k)}{\sum_{i=1}^n (u_{ik})^m D_h(e_i, G_k)} \right)^{1/(s-1)} \right]^{-1} = \left[\sum_{h=1}^p \left(\frac{\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)}{\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e)} \right)^{1/(s-1)} \right]^{-1} \quad (10)$$

Proof.

- (1) The matching function is given by Eq. (7).

As the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ are fixed, we can rewrite the criterion J as

$$J(\lambda_1, \dots, \lambda_K) = \sum_{k=1}^K J_k(\lambda_k) \quad \text{with } J_k(\lambda_k) = J_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p \lambda_{kj} J_{kj}$$

where $J_{kj} = \sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) = \sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)$.

Let $g(\lambda_{k1}, \dots, \lambda_{kp}) = \lambda_{k1} \times \dots \times \lambda_{kp} - 1$. We want to determine the extremes of $J_k(\lambda_{k1}, \dots, \lambda_{kp})$ with the restriction $g(\lambda_{k1}, \dots, \lambda_{kp}) = 0$. From the Lagrange multiplier method, and after some algebra, it follows that (for $j = 1, \dots, p$)

$$\lambda_{kj} = \frac{(\prod_{h=1}^p J_{kh})^{1/p}}{J_{kj}} = \frac{\{\prod_{h=1}^p (\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e))\}^{1/p}}{\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)}$$

Thus, an extreme value of J_k is reached when $J_k(\lambda_{k1}, \dots, \lambda_{kp}) = p\{J_{k1} \times \dots \times J_{kp}\}^{1/p}$. As $J_k(1, \dots, 1) = \sum_{j=1}^p J_{kj} = J_{k1} + \dots + J_{kp}$ and as it is well known that the arithmetic mean is greater than the geometric mean, i.e., $(1/p)(J_{k1} + \dots + J_{kp}) > \{J_{k1} \times \dots \times J_{kp}\}^{1/p}$ (the equality holds only if $J_{k1} = \dots = J_{kp}$), we conclude that this extreme is a minimum value.

(2) The matching function is given by Eq. (8).

As the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ are fixed, we can rewrite the criterion J as

$$J(\lambda_1, \dots, \lambda_K) = \sum_{k=1}^K J_k(\lambda_k) \quad \text{with} \quad J_k(\lambda_k) = J_k(\lambda_{k1}, \dots, \lambda_{kp}) = \sum_{j=1}^p (\lambda_{kj})^s J_{kj}$$

where $J_{kj} = \sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) = \sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)$.

Let $g(\lambda_{k1}, \dots, \lambda_{kp}) = \lambda_{k1} + \dots + \lambda_{kp} - 1$. We want to determine the extremes of $J_k(\lambda_{k1}, \dots, \lambda_{kp})$ with the restriction $g(\lambda_{k1}, \dots, \lambda_{kp}) = 0$. To do so, we shall apply the Lagrange multipliers method to solve the following system:

$$\nabla J_k(\lambda_{k1}, \dots, \lambda_{kp}) = \mu \nabla g_k(\lambda_{k1}, \dots, \lambda_{kp})$$

Then, for $k = 1, \dots, K$ and $j = 1, \dots, p$, we have

$$\frac{\partial J_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} = \mu \frac{\partial g_k(\lambda_{k1}, \dots, \lambda_{kp})}{\partial \lambda_{kj}} \Rightarrow \beta(\lambda_{kj})^{\beta-1} J_{kj} = \mu \Rightarrow \lambda_{kj} = \left(\frac{\mu}{\beta}\right)^{1/(\beta-1)} \cdot \frac{1}{(J_{kj})^{1/(\beta-1)}}$$

As we know that $\sum_{h=1}^p \lambda_{kh} = 1, \forall k$, we have $\sum_{h=1}^p \left(\frac{\mu}{\beta}\right)^{1/(\beta-1)} \cdot \frac{1}{(J_{kh})^{1/(\beta-1)}} = 1$, and after some algebra, we have that an extremum of J_k is reached when

$$\lambda_{kj} = \left[\sum_{h=1}^p \left(\frac{J_{kj}}{J_{kh}} \right)^{s-1} \right]^{-1} = \left[\sum_{h=1}^p \left(\frac{\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)}{\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e)} \right)^{s-1} \right]^{-1}$$

We have

$$\frac{\partial J_k}{\partial \lambda_{kj}} = \beta(\lambda_{kj})^{\beta-1} J_{kj} \Rightarrow \frac{\partial^2 J_k}{\partial (\lambda_{kj})^2} = \beta(\beta-1)(\lambda_{kj})^{\beta-2} J_{kj} \Rightarrow \frac{\partial^2 J_k}{\partial \lambda_{kj} \partial \lambda_{kh}} = 0 \quad \forall h \neq j$$

The Hessian matrix of J_k evaluated at $\lambda_k = (\lambda_{k1}, \dots, \lambda_{kp})$ is

$$H(\lambda_k) = \begin{pmatrix} \frac{\beta(\beta-1)J_{k1}}{\sum_{h=1}^p \left(\frac{J_{k1}}{J_{kh}} \right)^{(\beta-2)/(\beta-1)}} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\beta(\beta-1)J_{kp}}{\sum_{h=1}^p \left(\frac{J_{kp}}{J_{kh}} \right)^{(\beta-2)/(\beta-1)}} \end{pmatrix},$$

where $H(\lambda_k)$ is positive definite, so we can conclude that this extremum is a minimum. \square

Remark. Note that the closer the objects of a dissimilarity matrix \mathbf{D}_j are to the prototype G_k of a given fuzzy cluster C_k , the higher is the relevance weight of this dissimilarity matrix \mathbf{D}_j on the fuzzy cluster C_k .

Step 3: Definition of the best fuzzy partition.

In this step, the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ and the vector of relevance weight vectors $\mathbf{\Lambda} = (\lambda_1, \dots, \lambda_K)$ are fixed.

Proposition 2.5. The fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, where $\mathbf{u}_i = (u_{i1}, \dots, u_{iK})$ ($i = 1, \dots, n$), which minimizes the clustering criterion J , is such that the membership degree u_{ik} ($i = 1, \dots, n; k = 1, \dots, K$) of each pattern i in each fuzzy cluster C_k , under $u_{ik} \in [0, 1]$ and $\sum_{k=1}^K u_{ik} = 1$, is calculated according to the following expression:

$$u_{ik} = \left[\sum_{h=1}^K \left(\frac{D(\lambda_{k,s})(e_i, G_k)}{D(\lambda_{h,s})(e_i, G_h)} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_{kj})^s \sum_{e \in G_k} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_{hj})^s \sum_{e \in G_h} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1} \quad (11)$$

Proof. The Proof of Proposition 2.5 follows the same schema as that developed in the classical fuzzy K -means algorithm [3]. \square

Algorithm. The partitioning fuzzy K-medoids clustering algorithm with relevance weight for each dissimilarity matrix estimated locally (denoted hereafter as *MFCMdd-RWL-P* if the product of the weights is equal to one and as *MFCMdd-RWL-S* if the sum of the weights is equal to one) sets an initial partition and alternates three steps until convergence, when the criterion J reaches a stationary value representing a local minimum. This algorithm is summarized below.

Partitioning Fuzzy K-Medoids Clustering Algorithm with Relevance Weight for each Dissimilarity Matrix Estimated Locally

(1) Initialization.

Fix K (the number of clusters), $2 \leq K \ll n$; fix m , $1 < m < +\infty$; fix s , $1 \leq s < +\infty$; fix T (an iteration limit); fix $\varepsilon > 0$ and $\varepsilon \ll 1$;

Fix the cardinality $1 \leq q \ll n$ of the prototypes G_k ($k = 1, \dots, K$);

Set $t = 0$;

Set $\lambda_k^{(0)} = (\lambda_{k1}^{(0)}, \dots, \lambda_{kp}^{(0)}) = (1, \dots, 1)$ (for *MFCMdd-RWL-P*) or set $\lambda_k^{(0)} = (\lambda_{k1}^{(0)}, \dots, \lambda_{kp}^{(0)}) = (1/p, \dots, 1/p)$ (for *MFCMdd-RWL-S*), $k = 1, \dots, K$;

Randomly select K distinct prototypes $G_k^{(0)} \in E^{(q)}$ ($k = 1, \dots, K$);

For each object e_i ($i = 1, \dots, n$) compute its membership degree $u_{ik}^{(0)}$ ($k = 1, \dots, K$) on fuzzy cluster C_k :

$$u_{ik}^{(0)} = \left[\sum_{h=1}^K \left(\frac{D_{(\lambda_k^{(0)}, s)}(e_i, G_k^{(0)})}{D_{(\lambda_h^{(0)}, s)}(e_i, G_h^{(0)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_{kj}^{(0)})^s \sum_{e \in G_k^{(0)}} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_{hj}^{(0)})^s \sum_{e \in G_h^{(0)}} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

Compute:

$$J^{(0)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m D_{(\lambda_k^{(0)}, s)}(e_i, G_k^{(0)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m \sum_{j=1}^p (\lambda_{kj}^{(0)})^s \sum_{e \in G_k^{(0)}} d_j(e_i, e)$$

Step 1: computation of the best prototypes.

Set $t = t + 1$.

The fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ and the vector of relevance weight vectors $\mathbf{\Lambda}^{(t-1)} = (\lambda_1^{(t-1)}, \dots, \lambda_K^{(t-1)})$ are fixed.

Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of fuzzy cluster C_k ($k = 1, \dots, K$) according to the procedure described in Proposition 2.3.

(2) Step 2: computation of the best relevance weight vector.

The fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ and the vector of prototypes $\mathbf{G}^{(t)} = (G_1^{(t)}, \dots, G_K^{(t)})$ are fixed.

Compute the components $\lambda_{kj}^{(t)}$ ($j = 1, \dots, p$) of the relevance weight vector $\lambda_k^{(t)}$ ($k = 1, \dots, K$) according to Eq. (9) if the matching function is given by Eq. (7), or according to Eq. (10) if the matching function is given by Eq. (8).

(3) Step 3: definition of the best fuzzy partition.

The vector of prototypes $\mathbf{G}^{(t)} = (G_1^{(t)}, \dots, G_K^{(t)})$ and the vector of relevance weight vectors $\mathbf{\Lambda}^{(t)} = (\lambda_1^{(t)}, \dots, \lambda_K^{(t)})$ are fixed.

Compute the membership degree $u_{ik}^{(t)}$ of object e_i ($i = 1, \dots, n$) in fuzzy cluster C_k ($k = 1, \dots, K$), according to

$$u_{ik}^{(t)} = \left[\sum_{h=1}^K \left(\frac{D_{(\lambda_k^{(t)}, s)}(e_i, G_k^{(t)})}{D_{(\lambda_h^{(t)}, s)}(e_i, G_h^{(t)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_{kj}^{(t)})^s \sum_{e \in G_k^{(t)}} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_{hj}^{(t)})^s \sum_{e \in G_h^{(t)}} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

(5) *Stopping criterion.*

Compute:

$$J^{(t)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{(\lambda_k^{(t)}, s)}(e_i, G_k^{(t)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m \sum_{j=1}^p (\lambda_{kj}^{(t)})^s \sum_{e \in G_k^{(t)}} d_j(e_i, e)$$

If $|J^{(t)} - J^{(t-1)}| \leq \varepsilon$ or $t > T$: STOP; otherwise go to 2 (Step 1).

2.2.3. Partitioning fuzzy K-medoids clustering algorithms with relevance weight of each dissimilarity matrix estimated globally

The partitioning fuzzy K-medoids clustering algorithm presented in Section 2.2.2 can present numerical instabilities (overflow or division by zero) in the computation of the relevance weight of each dissimilarity matrix in each fuzzy cluster when the algorithm produces fuzzy clusters such that $\sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) \rightarrow 0$. To decrease significantly the probability of this kind of numerical instabilities, we present in this section an algorithm designed to give a fuzzy partition and a prototype for each fuzzy cluster as well as to learn a relevance weight for each dissimilarity matrix that changes at each algorithm's iteration but that is the same for all fuzzy clusters.

The partitioning fuzzy K-medoids clustering algorithm with relevance weight for each dissimilarity matrix estimated globally looks for a fuzzy partition $P = (C_1, \dots, C_K)$ of E into K fuzzy clusters represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, a corresponding K -dimensional vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ representing the fuzzy clusters in fuzzy partition P , and a single relevance weight vector λ , such that an adequacy criterion (objective function) measuring the fit between the fuzzy clusters and their prototypes is (locally) optimized. The adequacy criterion is defined as

$$J = \sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m D_{(\lambda, s)}(e_i, G_k) \quad (12)$$

in which $D_{(\lambda, s)}$ is the global matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$, parameterized by $1 \leq s < \infty$ and by the relevance weight vector $\lambda = (\lambda_1, \dots, \lambda_p)$ of the dissimilarity matrices \mathbf{D}_j ($j = 1, \dots, p$) into cluster C_k ($k = 1, \dots, K$).

Two matching functions with relevance weight for each dissimilarity matrix estimated globally are considered depending on whether the sum of the weights is equal to one or the product of the weights is equal to one. These matching functions are:

- (a) Matching function parameterized by both the parameter s and the vector of relevance weights $\lambda = (\lambda_1, \dots, \lambda_p)$, in which $s = 1$, $\lambda_j > 0$, and $\prod_{j=1}^p \lambda_j = 1$, and associated with cluster C_k ($k = 1, \dots, K$)

$$D_{(\lambda, s)}(e_i, G_k) = \sum_{j=1}^p (\lambda_j)^s D_j(e_i, G_k) = \sum_{j=1}^p \lambda_j \sum_{e \in G_k} d_j(e_i, e) \quad (13)$$

- (b) Matching function parameterized by both the parameter s and the vector of relevance weights $\lambda = (\lambda_1, \dots, \lambda_p)$, in which $1 < s < \infty$, $\lambda_j \in [0, 1]$, and $\sum_{j=1}^p \lambda_j = 1$, and associated with cluster C_k ($k = 1, \dots, K$)

$$D_{(\lambda, s)}(e_i, G_k) = \sum_{j=1}^p (\lambda_j)^s D_j(e_i, G_k) = \sum_{j=1}^p (\lambda_j)^s \sum_{e \in G_k} d_j(e_i, e) \quad (14)$$

In Eqs. (7) and (8), $D_j(e_i, G_k) = \sum_{e \in G_k} d_j(e_i, e)$ is the local dissimilarity between an example $e_i \in C_k$ and the cluster prototype computed on dissimilarity matrix \mathbf{D}_j ($j = 1, \dots, p$).

Note that this clustering algorithm also assumes that the prototype of each cluster is a subset (of fixed cardinality) of the set of objects. Moreover, the relevance weight vector λ is estimated globally: it changes at each iteration but is the same for all clusters.

This fuzzy K-medoids clustering algorithm sets an initial partition and alternates three steps until convergence, when the criterion J reaches a stationary value representing a local minimum.

Step 1: Computation of the best prototypes.

In this step, the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the relevance weight vector λ are fixed.

Proposition 2.6. *The prototype $G_k = G^* \in E^{(q)}$ of fuzzy cluster C_k ($k = 1, \dots, K$), which minimizes the clustering criterion J , is such that $\sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p (\lambda_j)^s D_j(e_i, G^*) \rightarrow \text{Min}$. The prototype G_k ($k = 1, \dots, K$) is computed according to the following procedure:*

$G^* \leftarrow \emptyset$

REPEAT

Find $e_l \in E, e_l \notin G^*$ such that $l = \text{argmin}_{1 \leq h \leq n} \sum_{i=1}^n (u_{ik})^m \sum_{j=1}^p (\lambda_j)^s d(e_i, e_h)$

$G^* \leftarrow G^* \cup \{e_l\}$

UNTIL $|G^*| = q$

Proof. The proof of Proposition 2.6 is straightforward. \square

Step 2: Computation of the best relevance weight vector.

In this step, the fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ are fixed.

Proposition 2.7. *The vectors of weights are computed according to the matching function used:*

- (1) *If the matching function is given by Eq. (13), the vector of weights $\lambda = (\lambda_1, \dots, \lambda_p)$, under $\lambda_j > 0$ and $\prod_{j=1}^p \lambda_j = 1$, have their weights λ_j ($j = 1, \dots, p$) calculated according to*

$$\begin{aligned} \lambda_j &= \frac{\left\{ \prod_{h=1}^p \left(\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m D_h(e_i, G_k) \right] \right) \right\}^{1/p}}{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) \right]} \\ &= \frac{\left\{ \prod_{h=1}^p \left(\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e) \right] \right) \right\}^{1/p}}{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) \right]} \end{aligned} \quad (15)$$

- (2) *If the matching function is given by Eq. (14), the vector of weights $\lambda = (\lambda_1, \dots, \lambda_p)$, under $\lambda_j \in [0, 1]$ and $\sum_{j=1}^p \lambda_j = 1$, have their weights λ_j ($j = 1, \dots, p$) calculated according to*

$$\begin{aligned} \lambda_j^{(t)} &= \left[\sum_{h=1}^p \left(\frac{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) \right]}{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m D_h(e_i, G_k) \right]} \right)^{1/(s-1)} \right]^{-1} \\ &= \left[\sum_{h=1}^p \left(\frac{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) \right]}{\sum_{k=1}^K \left[\sum_{i=1}^n (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e) \right]} \right)^{1/(s-1)} \right]^{-1} \end{aligned} \quad (16)$$

Proof. The proof proceeds in a similar way as presented in Proposition 2.4. \square

Remark 1. Note that the closer the objects of a dissimilarity matrix \mathbf{D}_j are to the prototypes G_1, \dots, G_K of the corresponding fuzzy clusters C_1, \dots, C_K the higher is the relevance weight of this dissimilarity matrix \mathbf{D}_j .

Remark 2. Numerical instabilities (overflow, division by zero) can always occur in the computation of the relevance weight of each dissimilarity matrix when the algorithm produces fuzzy clusters such that $\sum_{k=1}^K \sum_{i=1}^n (u_{ik})^m D_j(e_i, G_k) \rightarrow 0$. However, the probability of this kind of numerical instabilities is higher for the algorithms presented in Section 2.2.2.

Step 3: Definition of the best partition.

In this step, the vector of prototypes $\mathbf{G} = (G_1, \dots, G_K)$ and the relevance weight vector λ are fixed.

Proposition 2.8. The fuzzy partition represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, where $\mathbf{u}_i = (u_{i1}, \dots, u_{iK})$ ($i = 1, \dots, n$), which minimizes the clustering criterion J , is such that the membership degree u_{ik} ($i = 1, \dots, n; k = 1, \dots, K$) of each pattern i in each fuzzy cluster C_k , under $u_{ik} \in [0, 1]$ and $\sum_{k=1}^K u_{ik} = 1$, is calculated according to the following expression:

$$u_{ik} = \left[\sum_{h=1}^K \left(\frac{D_{(\lambda, s)}(e_i, G_k)}{D_{(\lambda, s)}(e_i, G_h)} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_j^s) \sum_{e \in G_k} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_j^s) \sum_{e \in G_h} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1} \quad (17)$$

Proof. The Proof of Proposition 2.8 follows the same schema as that developed in the classical fuzzy K -means algorithm [3]. \square

Algorithm. The partitioning fuzzy K -medoids clustering algorithm with relevance weight for each dissimilarity matrix estimated globally (denoted hereafter as *MFCMdd-RWG-P* if the product of the weights is equal to one and as *MFCMdd-RWG-S* if the sum of the weights is equal to one) sets an initial partition and alternates three steps until convergence, when the criterion J reaches a stationary value representing a local minimum. This algorithm is summarized below.

Partitioning Fuzzy K -Medoids Clustering Algorithm with Relevance Weight for Each Dissimilarity Matrix Estimated Globally

(1) Initialization.

Fix K (the number of clusters), $2 \leq K \ll n$; fix m , $1 < m < +\infty$; fix s , $1 \leq s < +\infty$; fix T (an iteration limit); fix $\varepsilon > 0$ and $\varepsilon \ll 1$;

Fix the cardinality $1 \leq q \ll n$ of the prototypes G_k ($k = 1, \dots, K$);

Set $t = 0$;

Set $\lambda^{(0)} = (\lambda_1^{(0)}, \dots, \lambda_p^{(0)}) = (1, \dots, 1)$ (for *MFCMdd-RWG-P*) or set $\lambda^{(0)} = (\lambda_1^{(0)}, \dots, \lambda_p^{(0)}) = (1/p, \dots, 1/p)$ (for *MFCMdd-RWG-S*), $k = 1, \dots, K$;

Randomly select K distinct prototypes $G_k^{(0)} \in E^{(q)}$ ($k = 1, \dots, K$);

For each object e_i ($i = 1, \dots, n$) compute its membership degree $u_{ik}^{(0)}$ ($k = 1, \dots, K$) on fuzzy cluster C_k :

$$u_{ik}^{(0)} = \left[\sum_{h=1}^K \left(\frac{D_{(\lambda^{(0)}, s)}(e_i, G_k^{(0)})}{D_{(\lambda^{(0)}, s)}(e_i, G_h^{(0)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_j^{(0)})^s \sum_{e \in G_k^{(0)}} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_j^{(0)})^s \sum_{e \in G_h^{(0)}} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

Compute:

$$J^{(0)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m D_{(\lambda^{(0)}, s)}(e_i, G_k^{(0)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(0)})^m \sum_{j=1}^p (\lambda_j^{(0)})^s \sum_{e \in G_k^{(0)}} d_j(e_i, e)$$

(2) Step 1: computation of the best prototypes.

Set $t = t + 1$.

The fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ and the vector of relevance weights $\lambda^{(t-1)} = (\lambda_1^{(t-1)}, \dots, \lambda_p^{(t-1)})$ are fixed.

Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of fuzzy cluster $C_k^{(t-1)}$ ($k = 1, \dots, K$) according to the procedure described in Proposition 2.6.

(3) Step 2: computation of the best relevance weight vector.

The fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ and the vector of prototypes $\mathbf{G}^{(t)} = (G_1^{(t)}, \dots, G_K^{(t)})$ are fixed.

Compute the components $\lambda_j^{(t)}$ ($j = 1, \dots, p$) of the relevance weight vector $\lambda^{(t)}$ ($k = 1, \dots, K$) according to Eq. (15) if the matching function is given by Eq. (13), or according to Eq. (16) if the matching function is given by Eq. (14).

(4) *Step 3: definition of the best fuzzy partition.*

The vector of prototypes $\mathbf{G}^{(t)} = (G_1^{(t)}, \dots, G_K^{(t)})$ and the vector of relevance weights $\lambda^{(t)}$ are fixed.

Compute the membership degree $u_{ik}^{(t)}$ of object e_i ($i = 1, \dots, n$) in fuzzy cluster C_k ($k = 1, \dots, K$), according to

$$u_{ik}^{(t)} = \left[\sum_{h=1}^K \left(\frac{D_{(\lambda^{(t)}, s)}(e_i, G_k^{(t)})}{D_{(\lambda^{(t)}, s)}(e_i, G_h^{(t)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^K \left(\frac{\sum_{j=1}^p (\lambda_j^{(t)})^s \sum_{e \in G_k^{(t)}} d_j(e_i, e)}{\sum_{j=1}^p (\lambda_j^{(t)})^s \sum_{e \in G_h^{(t)}} d_j(e_i, e)} \right)^{1/(m-1)} \right]^{-1}$$

(5) *Stopping criterion.*

Compute:

$$J^{(t)} = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{(\lambda^{(t)}, s)}(e_i, G_k^{(t)}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m \sum_{j=1}^p (\lambda_j^{(t)})^s \sum_{e \in G_k^{(t)}} d_j(e_i, e)$$

If $|J^{(t)} - J^{(t-1)}| \leq \varepsilon$ or $t > T$: STOP; otherwise go to 2 (Step 1).

2.3. Convergence properties of the algorithms

In this section, we illustrate the convergence properties of the presented algorithms giving the proof of the convergence of the partitioning fuzzy K-medoids clustering algorithm *MFCMdd-RWL-P* with relevance weight for each dissimilarity matrix estimated locally introduced in Section 2.2.2.

Partitioning fuzzy K-medoids clustering algorithm *MFCMdd-RWL-P* looks for a fuzzy partition $P^* = \{C_1^*, \dots, C_K^*\}$ of E into K fuzzy clusters represented by $\mathbf{U}^* = (\mathbf{u}_1^*, \dots, \mathbf{u}_n^*)$, a corresponding K -dimensional vector of prototypes $\mathbf{G}^* = (G_1^*, \dots, G_K^*)$ representing the fuzzy clusters in fuzzy partition P , and a K -dimensional vector of relevance weight vectors (one for each fuzzy cluster) $\Lambda^* = (\lambda_1^*, \dots, \lambda_K^*)$, such that

$$J(\mathbf{G}^*, \Lambda^*, \mathbf{U}^*) = \min\{J(\mathbf{G}, \Lambda, \mathbf{U}) : \mathbf{G} \in \mathbb{L}^K, \Lambda \in \Xi^K, \mathbf{U} \in \mathbb{U}^n\} \quad (18)$$

where

- \mathbb{U} is the space of fuzzy partition membership degrees such that $\mathbf{u}_k \in \mathbb{U}$ ($k = 1, \dots, K$). In this paper $\mathbb{U} = \{\mathbf{u} = (u_1, \dots, u_K) \in [0, 1] \times \dots \times [0, 1] = [0, 1]^K : \sum_{k=1}^K u_k = 1\}$ and $\mathbf{U} \in \mathbb{U}^n = \mathbb{U} \times \dots \times \mathbb{U}$;
- \mathbb{L} is the representation space of prototypes such that $G_k \in \mathbb{L}$ ($k = 1, \dots, K$) and $\mathbf{G} \in \mathbb{L}^K = \mathbb{L} \times \dots \times \mathbb{L}$. In this paper, $\mathbb{L} = E^{(q)} = \{A \subset E : |A| = q\}$; and
- Ξ is the space of vectors of weights such that $\lambda_k \in \Xi$ ($k = 1, \dots, K$). In this paper, $\Xi = \{\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p : \lambda_j > 0 \text{ and } \prod_{j=1}^p \lambda_j = 1\}$ and $\Lambda \in \Xi^K = \Xi \times \dots \times \Xi$.

According to [19], the properties of convergence of this kind of algorithm can be studied from two series: $v_t = (\mathbf{G}^t, \Lambda^t, \mathbf{U}^t) \in \mathbb{L}^K \times \Xi^K \times \mathbb{U}^n$ and $u_t = J(v_t) = J(\mathbf{G}^t, \Lambda^t, \mathbf{U}^t)$, $t = 0, 1, \dots$. From an initial term $v_0 = (\mathbf{G}^0, \Lambda^0, \mathbf{U}^0)$, the algorithm computes the different terms of the series v_t until the convergence (to be shown) when the criterion J achieves a stationary value.

Proposition 2.9. *The series $u_t = J(v_t)$ decreases at each iteration and converges.*

Proof. Following [19], we first show that the inequalities (I), (II) and (III),

$$\underbrace{J(\mathbf{G}^t, \Lambda^t, \mathbf{U}^t)}_{u_t} \stackrel{\text{(I)}}{\geq} J(\mathbf{G}^{t+1}, \Lambda^t, \mathbf{U}^t) \stackrel{\text{(II)}}{\geq} J(\mathbf{G}^{t+1}, \Lambda^{t+1}, \mathbf{U}^t) \stackrel{\text{(III)}}{\geq} \underbrace{J(\mathbf{G}^{t+1}, \Lambda^{t+1}, \mathbf{U}^{t+1})}_{u_{t+1}}$$

hold (i.e., the series decreases at each iteration).

The inequality (I) holds because

$$J(\mathbf{G}^t, \Lambda^t, \mathbf{U}^t) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{\lambda_k^{(t)}}(e_i, G_k^{(t)})$$

$$J(\mathbf{G}^{t+1}, \mathbf{\Lambda}^t, \mathbf{U}^t) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{\lambda_k^{(t)}}(e_i, G_k^{(t+1)})$$

and according to Proposition 2.6,

$$\mathbf{G}^{(t+1)} = (G_1^{(t+1)}, \dots, G_K^{(t+1)}) = \underbrace{\operatorname{argmin}}_{\mathbf{G}=(G_1, \dots, G_K) \in \mathbb{L}^K} \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{\lambda_k^{(t)}}(e_i, G_k)$$

Inequality (II) also holds because

$$J(\mathbf{G}^{t+1}, \mathbf{\Lambda}^{(t+1)}, \mathbf{U}^t) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{\lambda_k^{(t+1)}}(e_i, G_k^{(t+1)})$$

and according Proposition 2.7,

$$\mathbf{\Lambda}^{(t+1)} = (\lambda_1^{(t+1)}, \dots, \lambda_K^{(t+1)}) = \underbrace{\operatorname{argmin}}_{\mathbf{\Lambda}=(\lambda_1, \dots, \lambda_K) \in \Xi^K} \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t)})^m D_{\lambda_k}(e_i, G_k^{(t+1)})$$

Inequality (III) holds as well because

$$J(\mathbf{G}^{t+1}, \mathbf{\Lambda}^{t+1}, \mathbf{U}^{t+1}) = \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t+1)})^m D_{\lambda_k^{(t+1)}}(e_i, G_k^{(t+1)})$$

and according Proposition 2.8,

$$\mathbf{U}^{t+1} = (\mathbf{u}_1^{t+1}, \dots, \mathbf{u}_n^{t+1}) = \underbrace{\operatorname{argmin}}_{\mathbf{U}=(\mathbf{u}_1, \dots, \mathbf{u}_n) \in \mathbb{U}^n} \sum_{k=1}^K \sum_{i=1}^n (u_{ik}^{(t+1)})^m D_{\lambda_k^{(t+1)}}(e_i, G_k^{(t+1)})$$

Finally, because the series u_t decreases and it is bounded ($J(v_t) \geq 0$), it converges. \square

Proposition 2.10. *The series $v_t = (\mathbf{G}^t, \mathbf{\Lambda}^t, \mathbf{U}^t)$ converges.*

Proof. Assume that the stationarity of the series u_t is achieved in the iteration $t = T$. Then, we have that $u_T = u_{T+1}$ and then $J(v_T) = J(v_{T+1})$. From $J(v_T) = J(v_{T+1})$, we have $J(\mathbf{G}^t, \mathbf{\Lambda}^t, \mathbf{U}^t) = J(\mathbf{G}^{T+1}, \mathbf{\Lambda}^{T+1}, \mathbf{U}^{T+1})$, and this equality, according to Proposition 2.9, can be rewritten as equalities (I), (II) and (III):

$$\begin{aligned} J(\mathbf{G}^t, \mathbf{\Lambda}^t, \mathbf{U}^t) &\stackrel{\text{I}}{=} J(\mathbf{G}^{T+1}, \mathbf{\Lambda}^T, \mathbf{U}^T) \stackrel{\text{II}}{=} J(\mathbf{G}^{T+1}, \mathbf{\Lambda}^{T+1}, \mathbf{U}^T) \\ &\stackrel{\text{III}}{=} J(\mathbf{G}^{T+1}, \mathbf{\Lambda}^{T+1}, \mathbf{U}^{T+1}) \end{aligned}$$

From the first equality (I), we have $\mathbf{G}^T = \mathbf{G}^{T+1}$, because \mathbf{G} is unique, minimizing J when the partition \mathbf{U}^T and the vector of vectors of weights $\mathbf{\Lambda}^T$ are fixed. From the second equality (II), we have $\mathbf{\Lambda}^T = \mathbf{\Lambda}^{T+1}$, because $\mathbf{\Lambda}$ is unique, minimizing J when the partition \mathbf{U}^T and the vector of prototypes \mathbf{G}^{T+1} are fixed. Moreover, from the third equality (III), we have $\mathbf{U}^T = \mathbf{U}^{T+1}$, because \mathbf{U} is unique, minimizing J when the vector of prototypes \mathbf{G}^{T+1} and the vector of vectors of weights $\mathbf{\Lambda}^T$ are fixed.

Finally, we conclude that $v_T = v_{T+1}$. This conclusion holds for all $t \geq T$ and $v_t = v_T, \forall t \geq T$ and it follows that the series v_t converges. \square

Table 1
Relevance weights for the clusters.

Data matrix	<i>MFCMdd-RWL-P</i>		<i>CARD-R</i>	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
Feature 1	0.0373	6.6939	0.0014	0.9985
Feature 2	26.7705	0.1493	0.9800	0.0199

3. Empirical results

To evaluate the performance of these partitioning relational fuzzy clustering algorithms in comparison with the *NERF* and *CARD-R* relational fuzzy clustering algorithms, applications with synthetic and real data sets described by real-valued variables (available at the UCI Repository <http://www.i.cs.uci.edu/mlearn/MLRepository.html>) as well as with data sets described by symbolic variables of several types (interval-valued and histogram-valued variables) are considered.

These relational fuzzy clustering algorithms will be applied to these data sets to obtain first a fuzzy partition $P = (C_1, \dots, C_K)$ of E into K fuzzy clusters represented by $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, with $\mathbf{u}_i = (u_{i1}, \dots, u_{iK})$ ($i = 1, \dots, n$). Then, a hard partition $Q = (Q_1, \dots, Q_K)$ will be obtained from this fuzzy partition by defining the hard cluster Q_k ($k = 1, \dots, K$) as $Q_k = \{e_i : u_{ik} \geq u_{im} \forall m \in \{1, \dots, K\}\}$.

To compare the clustering results furnished by the clustering methods, an external index – the corrected Rand (*CR*) index – will be considered. The *CR* index [20] assesses the degree of agreement (similarity) between an a priori partition and a partition furnished by the clustering algorithm. Moreover, the *CR* index is not sensitive to the number of classes in the partitions or the distribution of the items in the clusters [20]. Finally, the *CR* index takes its values from the interval $[-1, 1]$, in which the value 1 indicates perfect agreement between partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance [21].

Before going ahead, to illustrate the performance of these partitioning relational fuzzy clustering algorithms, we will consider a 2-dimensional synthetic Gaussian clusters, proposed by [11], obtained according to

$$\mu_1 = (-0.4, 0.1), \quad \Sigma_1 = \begin{pmatrix} 236.6 & 0.6 \\ 0.6 & 1.0 \end{pmatrix}$$

and

$$\mu_2 = (0.1, 32.0), \quad \Sigma_2 = \begin{pmatrix} 1.0 & -0.2 \\ -0.2 & 215.2 \end{pmatrix}$$

There are 150 data points per cluster, and each cluster has one low-variance and one high-variance feature.

First, from the above data it is obtained a single relational matrix that represents the pairwise Euclidean distance taking into account both features. Then, *NERF* is performed on this single relational matrix. Next, from each feature it is obtained a relational matrix (also using pairwise Euclidean distance). Then, *CARD-R* and *MFCMdd-RWL-P* are performed on these two relational matrices.

In this illustrative example, each relational fuzzy clustering algorithm was run 100 times, and the best result was selected according to the adequacy criterion. The parameters m , T , and ε were set, respectively, to 2, 350, and 10^{-10} . The parameter s and the cardinality of the prototypes was fixed to 1 for the *MFCMdd-RWL-P* algorithm. The number of clusters was fixed to 2. The hard cluster partitions obtained from these fuzzy clustering methods were compared with the known a priori class partition. The comparison criterion used was the *CR* index, which was calculated for the best result.

The *CR* index was 0.7272, 0.9734 and 1.0000 for, respectively, *NERF*, *CARD-R* and *MFCMdd-RWL-P*. As pointed out by [11], *NERF* treat both features equally important in both clusters (i.e., they have tendency to identify spherical clusters). *CARD-R* and *MFCMdd-RWL-P* learned different relevance weights for each relational matrix in each cluster, and as a result, the data is partitioned according to the a priori classes. Table 1 shows the relevance weights given by *CARD-R* and *MFCMdd-RWL-P*.

Table 2

Configurations of quantitative data sets: mean vectors of the bivariate normal distributions of the classes.

μ	Class 1	Class 2	Class 3	Class 4
μ_1	45	70	45	42
μ_2	30	38	35	20

Table 3

Configurations of quantitative data sets: covariance matrices of the bivariate normal distributions of the classes.

Σ	Synthetic data set 1				Synthetic data set 2			
	Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4
σ_1	100	20	50	1	15	15	15	15
σ_2	1	70	40	10	5	5	5	5
ρ_{12}	0.88	0.87	0.90	0.89	0.88	0.87	0.90	0.89
Σ	Synthetic data set 3				Synthetic data set 4			
	Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4
σ_2	15	11	1	5	7	7	7	7
ρ_{12}	0.78	0.77	0.773	0.777	0.78	0.77	0.773	0.777

3.1. Synthetic real-valued data sets

This paper considers data sets described by two real-valued variables. Each data set has 450 points scattered among four classes of unequal sizes and elliptical shapes: two classes of size 150 each and two classes of sizes 50 and 100. Each class in these quantitative data sets was drawn according to a bivariate normal distribution.

Four different configurations of real-valued data drawn from bivariate normal distributions according to each class are considered. These distributions have the same mean vector (Table 2), but different covariance matrices (Table 3):

- (1) The variance is different between the variables and from one class to another (synthetic data set 1).
- (2) The variance is different between the variables, but is almost the same from one class to another (synthetic data set 2).
- (3) The variance is almost the same between the variables and different from one class to another (synthetic data set 3).
- (4) Finally, the variance is almost the same between the variables and from one class to another (synthetic data set 4).

Several dissimilarity matrices are obtained from these data sets. One of these dissimilarity matrices has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously the two real-valued attributes. All the other dissimilarity matrices have the cells that are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute.

Because all the attributes are real-valued, distance functions belonging to the family of Minkowsky distance (Manhattan or “city-block” distance, Euclidean distance, Chebyshev distance, etc.) are suitable to compute dissimilarities between the objects. In this paper, the dissimilarity between pairs of objects was computed according to the Euclidean (L_2) distance.

All dissimilarity matrices were normalized according to their overall dispersion [22] to have the same dynamic range. This means that each dissimilarity $d(e_k, e_{k'})$ in a given dissimilarity matrix was normalized as $d(e_k, e_{k'})/T$, where $T = \sum_{k=1}^n d(e_k, g)$ is the overall dispersion and $g = e_l \in E = \{e_1, \dots, e_n\}$ is the overall prototype, which is computed according to $l = \operatorname{argmin}_{1 \leq h \leq n} \sum_{k=1}^n d(e_k, e_h)$.

For these data sets, *NERF* and *SFCMdd* were performed on the dissimilarity matrix that has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously the two real-valued attributes. *CARD-R*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S* and *MFCMdd-RWG-S* were performed simultaneously on all dissimilarity matrices that have the cells that are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute.

Table 4

Performance of the algorithms on the synthetic data sets: average and standard deviation (in parenthesis) of the *CR* index.

Algorithms	Synthetic data sets			
	1	2	3	4
<i>NERF</i>	0.1334 (0.0206)	0.1416 (0.0173)	0.2381 (0.0279)	0.2942 (0.0285)
<i>SFCMdd</i>	0.1360 (0.0218)	0.1417 (0.0173)	0.2450 (0.0336)	0.2911 (0.0241)
<i>MFCMdd</i>	0.1332 (0.0245)	0.2184 (0.0289)	0.2611 (0.0289)	0.2875 (0.0324)
<i>MFCMdd-RWG-P</i>	0.1382 (0.0275)	0.2265 (0.0274)	0.2589 (0.0271)	0.2959 (0.0284)
<i>MFCMdd-RWG-S</i>	0.1389 (0.0244)	0.2206 (0.0287)	0.2588 (0.0313)	0.2899 (0.0360)
<i>MFCMdd-RWL-P</i>	0.5330 (0.0215)	0.2367 (0.0314)	0.2407 (0.0281)	0.2772 (0.0273)
<i>MFCMdd-RWL-S</i>	0.5217 (0.0283)	0.2082 (0.0609)	0.2126 (0.0217)	0.2635 (0.0234)
<i>CARD-R</i>	0.4810 (0.029)	0.2571 (0.021)	0.1285 (0.013)	0.1625 (0.019)

Table 5

Performance of the algorithms on the synthetic data sets: 95% confidence interval for the average of the *CR* index.

Algorithms	Synthetic data sets			
	1	2	3	4
<i>NERF</i>	0.1293–0.1374	0.1382–0.1449	0.2326–0.2435	0.2886–0.2997
<i>SFCMdd</i>	0.1318–0.1401	0.1383–0.1450	0.2384–0.2515	0.2863–0.2958
<i>MFCMdd</i>	0.1284–0.1379	0.2128–0.2239	0.2554–0.2666	0.2811–0.2938
<i>MFCMdd-RWG-P</i>	0.1328–0.1435	0.2211–0.2318	0.2535–0.2642	0.2903–0.3014
<i>MFCMdd-RWG-S</i>	0.1340–0.1437	0.2149–0.2262	0.2525–0.2650	0.2827–0.2970
<i>MFCMdd-RWL-P</i>	0.5288–0.5371	0.2305–0.2428	0.2351–0.2462	0.2718–0.2825
<i>MFCMdd-RWL-S</i>	0.5160–0.5273	0.1961–0.2202	0.2082–0.2169	0.2588–0.2681
<i>CARD-R</i>	0.4751–0.4868	0.2529–0.2612	0.1259–0.1310	0.1587–0.1662

The relational fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a four-cluster fuzzy partition. The hard cluster partitions (obtained from the fuzzy partitions given by the relational fuzzy clustering algorithms) were compared with the known a priori class partition. For the synthetic data sets, the *CR* index was estimated in the framework of a Monte Carlo simulation with 100 replications. The average and the standard deviation of this index between these 100 replications were calculated. In each replication, a relational clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) 100 times and the best result was selected according to the adequacy criterion. The parameters m , T , and ε were set, respectively, to 2, 350, and 10^{-10} . The parameter s was set to 1 for the algorithms *MFCMdd-RWL-P* and *MFCMdd-RWG-P*, and to 2 for the algorithms *MFCMdd-RWL-S*, and *MFCMdd-RWG-S*. The *CR* index was calculated for the best result.

Table 4 shows the performance of the *NERF* and *CARD-R* algorithms, as well as the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWG-S* algorithms (with prototypes of cardinality $|G_k| = 1, k = 1, \dots, 4$) on the synthetic data sets according to the average and the standard deviation of the *CR* index. Table 5 shows the 95% confidence interval for the average of the *CR* index.

The performance of the *MFCMdd-RWL-P*, *MFCMdd-RWL-S*, and *CARD-R* algorithms was clearly superior when the variance was different between the variables and from one class to another (synthetic data set 1), in comparison with all the other algorithms. *NERF* and *SFCMdd* presented clearly the worst performance when the variance was different between the variables but almost the same from one class to another (synthetic data set 2).

Moreover, the *MFCMdd*, *MFCMdd-RWG-P*, and *MFCMdd-RWG-S* algorithms were superior in comparison with all the other algorithms when the variance was almost the same between the variables and different from one class to another (synthetic data set 3). Finally, *NERF*, *SFCMdd*, *MFCMdd*, *MFCMdd-RWG-P*, and *MFCMdd-RWG-S* performed better than *MFCMdd-RWL-P*, *MFCMdd-RWL-S*, and *CARD-R* when the variance was almost the same between the variables and from one class to another (synthetic data set 4). For these last two configurations, *CARD-R* presented the worst performance.

In conclusion, *MFCMdd-RWL-P* and *MFCMdd-RWL-S* (as well as *CARD-R*) were clearly superior in the synthetic data sets where the variance was different between the variables and from one class to another, whereas *MFCMdd-*

Table 6

Iris data set: *CR* index.

$ G_k $	<i>SFCMdd</i>	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	0.7302	0.6412	0.8507	0.8856	0.8680	0.6412
2	0.7287	0.6412	0.8176	0.8856	0.8680	0.6764
3	0.8015	0.6764	0.8176	0.8856	0.8680	0.6764
5	0.7429	0.6451	0.8176	0.8856	0.8856	0.6451
10	0.8016	0.6637	0.8680	0.8856	0.8682	0.6757

RWG-P and *MFCMdd-RWG-S* were superior in the synthetic data sets where the variance was almost the same between the variables and different from one class to another, as well as where the variance was almost the same between the variables and from one class to another.

3.2. UCI Machine Learning Repository data sets

This paper considers data sets on iris plants, thyroid gland, and wine. These data sets are found at <http://www.ics.uci.edu/mllearn/MLRepository.html>.

All these data sets are described by a data matrix of “objects \times real-valued attributes”. Several dissimilarity matrices were obtained from these data matrices. One of these dissimilarity matrices has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the real-valued attributes. All the other dissimilarity matrices have the cells that are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute. Because all the attributes are real-valued, distance functions belonging to the family of Minkowsky distance (Manhattan or “city-block” distance, Euclidean distance, Chebyshev distance, etc.) are suitable to compute dissimilarities between the objects. In this paper, the dissimilarity between pairs of objects was computed according to the Euclidean (L_2) distance.

For these data sets, *NERF* and *SFCMdd* were performed on the dissimilarity matrix that has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the real-valued attributes. *CARD-R*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWG-S* were performed simultaneously on all dissimilarity matrices that have the cells that are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute.

All dissimilarity matrices were normalized according to their overall dispersion [22] to have the same dynamic range.

Each relational fuzzy clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) 100 times, and the best result was selected according to the adequacy criterion. The parameters m , T , and ε were set, respectively, to 2, 350, and 10^{-10} . The parameter s was set to 1 for the *MFCMdd-RWL-P* and *MFCMdd-RWG-P* algorithms, and to 2 for the *MFCMdd-RWL-S* and *MFCMdd-RWG-S* algorithms. The hard cluster partitions obtained from these fuzzy clustering methods were compared with the known a priori class partition. The comparison criterion used was the *CR* index, which was calculated for the best result.

3.2.1. Iris plant data set

This data set consists of three types (classes) of iris plants: iris setosa, iris versicolor, and iris virginica. The three classes each have 50 instances (objects). One class is linearly separable from the other two; the latter two are not linearly separable from each other. Each object is described by four real-valued attributes: (1) sepal length, (2) sepal width, (3) petal length, and (4) petal width.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 3-cluster fuzzy partition. The 3-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 3-class partition. Table 6 shows the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWG-S* algorithms on the iris plant data set according to the *CR* index, considering prototypes of cardinality $|G_k| = 1, 2, 3, 5$ and 10 ($k = 1, 2, 3$). *NERF* had 0.7294, whereas *CARD-R* had 0.8856 for *CR* index.

Table 7

Iris data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWL-S</i>			<i>CARD-R</i>		
		Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
Sepal length	0.5311	0.0425	0.0604	0.0808	0.0821	0.0451	0.0852
Sepal width	0.3028	0.0083	0.0588	0.0675	0.0641	0.0107	0.0905
Petal length	2.7631	0.6232	0.4136	0.4829	0.4228	0.5849	0.4657
Petal width	2.2499	0.3258	0.4671	0.3686	0.4308	0.3592	0.3584

For this data set, the best performance was presented by *CARD-R* and *MFCMdd-RWL-S*. The *MFCMdd-RWG-P* and *MFCMdd-RWL-P* algorithms also performed very well on this data set. The worst performance was presented by *MFCMdd* and *MFCMdd-RWG-S*.

Table 7 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by the *MFCMdd-RWG-P* algorithm with prototypes of cardinality 5) and locally for each cluster and dissimilarity matrix (according to the results given by the *MFCMdd-RWL-S* algorithm with prototypes of cardinality 5 and by the *CARD-R* algorithm).

Concerning the 3-cluster partition given by *MFCMdd-RWG-P*, dissimilarity matrices computed taking into account only “(3) petal length” or only “(4) petal width” attributes have the highest relevance weight; thus, the objects described by these dissimilarity matrices are closer to the prototypes of the clusters than are the objects described by dissimilarity matrices computed taking into account only “(1) sepal length” or only “(2) sepal width” attributes.

Table 7 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-S* and *CARD-R* algorithms, each cluster (1, 2 and 3) is associated to the same known a priori class.

For the 3-cluster fuzzy partition given by *MFCMdd-RWL-S*, dissimilarity matrices computed taking into account only “(3) petal length” and “(4) petal width” (in that order) are the most important in the definition of cluster 1, dissimilarity matrices computed taking into account only “(4) petal width” and “(3) petal length” (in that order) are the most important in the definition of cluster 2, whereas dissimilarity matrices computed taking into account only “(3) petal length” and “(4) petal width” (in that order) are the most important in the definition of cluster 3.

For the 3-cluster fuzzy partition given by *CARD-R*, dissimilarity matrices computed taking into account only “(4) petal width” and “(3) petal length” (in that order) are the most important in the definition of cluster 1, dissimilarity matrices computed taking into account only “(3) petal length” and “(4) petal width” (in that order) are the most important in the definition of cluster 2, whereas dissimilarity matrices computed taking into account only “(3) petal length” and “(4) petal width” (in that order) are the most important in the definition of cluster 3.

One can observe that both algorithms (*MFCMdd-RWL-S* and *CARD-R*) presented the same set of relevant variables in the formation of each cluster (even if the relevance order was different for clusters 1 and 2). This was expected because the 3-cluster hard partitions given by these algorithms presented a high degree of similarity with the known a priori 3-class partition.

3.2.2. Thyroid gland data set

This data set consists of three classes concerning the state of the thyroid gland: normal, hyperthyroidism, and hypothyroidism. The classes (1, 2, and 3) have 150, 35, and 30 instances, respectively. Each object is described by five real-valued attributes: (1) T3-resin uptake test, (2) total serum thyroxine, (3) total serum triiodothyronine, (4) basal thyroidstimulating hormone (TSH) and (5) maximal absolute difference in TSH value.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 3-cluster fuzzy partition. The 3-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 3-class partition. Table 8 shows the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S* and *MFCMdd-RWG-S* algorithms on the thyroid data set according to *CR* index, considering prototypes of cardinality $|G_k| = 1, 2, 3, 5$, and 10 ($k = 1, 2, 3$). *NERF* had 0.4413, whereas *CARD-R* had 0.2297 for *CR* index.

Table 8

Thyroid data set: *CR* index.

$ G_k $	<i>SFCMdd</i>	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	0.2483	0.7025	0.8631	0.2212	0.6549	0.5484
2	0.2767	0.3380	0.8776	0.2441	0.6257	0.7811
3	0.2849	0.6702	0.8930	0.2470	0.3205	0.7486
5	0.2059	0.2634	0.8332	0.2503	0.3233	0.2634
10	0.1341	0.3685	0.8332	0.2503	0.3306	0.3349

Table 9

Thyroid data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-S</i>	<i>MFCMdd-RWL-P</i>			<i>CARD-R</i>		
		Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
T3-resin uptake test	0.2384	0.2808	0.0694	1.8999	0.0037	0.0184	0.0641
Total serum thyroxin	0.1911	0.4915	0.1770	4.3718	0.0039	0.0383	0.2538
Total serum triiodothyronine	0.2027	0.9651	0.0642	5.3598	0.0029	0.9044	0.6654
Basal thyroid stimulating hormone (TSH)	0.1539	8.2143	35.1958	0.1468	0.9345	0.0350	0.0051
Maximal absolute difference in TSH value	0.2136	0.9136	35.9785	0.1529	0.0548	0.0036	0.0113

For this data set, the best performance was presented by *MFCMdd-RWL-P*. Algorithms *MFCMdd-RWG-S* (with prototypes of cardinality 2 and 3) and *MFCMdd* (with prototypes of cardinality 1) also performed well on this data set. The worst performance was presented by *SFCMdd*, *MFCMdd-RWL-S* and *CARD-R*.

Table 9 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by the *MFCMdd-RWG-S* algorithm with prototypes of cardinality 2) and locally for each cluster and dissimilarity matrix (according to the best results given by the *MFCMdd-RWL-P* algorithm with prototypes of cardinality 3 and by the *CARD-R* algorithm).

Concerning the 3-cluster partition given by *MFCMdd-RWG-S*, dissimilarity matrices computed taking into account only “(1) T3-resin uptake test” and only “(4) basal thyroidstimulating hormone (TSH)” attributes had the highest (0.2384) and the lowest (0.1539) relevance weights, respectively, in the definition of the fuzzy clusters.

Table 9 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-P* and *CARD-R* algorithms, each cluster (1, 2 and 3) is associated to the same known a priori class.

One can observe that these algorithms (*MFCMdd-RWL-P* and *CARD-R*) presented almost the same set of relevant dissimilarity matrices in the formation of clusters 1 and 3 and different sets of relevant dissimilarity matrices in the formation of cluster 2. Note that the *CR* index between the 3-cluster hard partitions given by, respectively, *MFCMdd-RWL-P* and *CARD-R*, and the known a priori 3-class partition, is 0.8930 and 0.2297. Consequently, the 3-cluster hard partitions given by these algorithms can be quite different.

3.2.3. Wine data set

This data set consists of three types (classes) of wines grown in the same region in Italy, but derived from three different cultivars. The classes (1, 2, and 3) have 59, 71 and 48 instances, respectively. Each wine is described by 13 real-valued attributes representing the quantities of 13 components found in each of the three types of wines. These attributes are: (1) alcohol, (2) malic acid, (3) ash, (4) alkalinity of ash, (5) magnesium, (6) total phenols, (7) flavonoids, (8) non-flavonoid phenols, (9) proanthocyanins, (10) color intensity, (11) hue, (12) OD280/OD315 of diluted wines, and (13) proline.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 3-cluster fuzzy partition. The 3-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 3-class partition. Table 10 shows the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWG-S* algorithms on the wine data set according to the *CR* index, considering

Table 10

Wine data set: *CR* index.

$ G_k $	<i>SFCMdd</i>	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	0.3614	0.7557	0.7283	0.3897	0.7557	0.7557
2	0.3614	0.8158	0.7723	0.3459	0.8332	0.8158
3	0.3614	0.8169	0.7865	0.3474	0.8332	0.8169
5	0.3539	0.8185	0.7724	0.3523	0.8024	0.8185
10	0.3447	0.8024	0.7420	0.3395	0.8185	0.8348

Table 11

Wine data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-S</i>	<i>MFCMdd-RWL-P</i>			<i>CARD-R</i>		
		Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
Alcohol	0.0751	1.1026	0.6761	1.0987	0.0405	0.0148	0.0579
Malic acid	0.0705	1.1717	1.8609	0.5828	0.0324	0.7508	0.0284
Ash	0.0960	0.5500	0.6377	1.1293	0.0345	0.0192	0.0447
Alkalinity of ash	0.0827	0.9572	0.5790	0.7871	0.0661	0.0116	0.0504
Magnesium	0.0917	0.5491	0.8072	0.7878	0.0484	0.0145	0.0432
Total phenols	0.0642	0.8449	1.3162	1.2308	0.0911	0.0200	0.1171
Flavonoids	0.0549	1.5870	1.5817	1.8660	0.1482	0.0268	0.1704
Non-flavonoid phenols	0.0804	0.8636	1.3401	0.6930	0.0688	0.0276	0.0286
Proanthocyanins	0.0808	1.0337	0.7062	0.9928	0.0429	0.0245	0.0753
Color intensity	0.0808	2.1246	1.2747	0.4482	0.1486	0.0188	0.0262
Hue	0.0767	1.0007	1.6444	0.8422	0.0626	0.0274	0.0550
OD280/OD315 of diluted wines	0.0726	0.8517	1.1636	1.4302	0.1648	0.0306	0.1030
Proline	0.0730	1.2347	0.5545	2.6132	0.0505	0.0127	0.1989

prototypes of cardinality $|G_k| = 1, 2, 3, 5$, and 10 ($k = 1, 2, 3$). *NERF* had 0.3539, whereas *CARD-R* had 0.3808 for *CR* index.

For this data set, the best performance was presented by *MFCMdd-RWG-S*, *MFCMdd-RWG-P*, and *MFCMdd*. The *MFCMdd-RWL-P* algorithm also performed well on this data set. The worst performance was presented by *MFCMdd-RWL-S*, *SFCMdd*, *NERF*, and *CARD-R*.

Table 11 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by *MFCMdd-RWG-S* with prototypes of cardinality 10) and locally for each cluster and dissimilarity matrix (according to the best results given by *MFCMdd-RWL-P* with prototypes of cardinality 3 and by the *CARD-R* algorithm).

Concerning the 3-cluster partition given by *MFCMdd-RWG-S*, dissimilarity matrices computed taking into account only the “(3) ash” and only the “(7) flavonoids” attributes had the highest and the lowest relevance weights, respectively, in the definition of the fuzzy clusters.

Table 11 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-P* and *CARD-R* algorithms, each cluster (1, 2 and 3) is associated to the same known a priori class.

One can observe that for the fuzzy partition given by the *MFCMdd-RWL-P* algorithm, 7 dissimilarity matrices were relevant in the formation of clusters 1 and 2 and 6 dissimilarity matrices were relevant in the formation of cluster 3, whereas for the fuzzy partition given by the *CARD-R* algorithm, 4 dissimilarity matrices were relevant in the formation of cluster 1, only one dissimilarity matrix was relevant in the formation of clusters 2 and 4 dissimilarity matrices were relevant in the formation of cluster 3. Moreover, 2, 1 and 4 dissimilarity matrices were simultaneously relevant in both partitions for, respectively, the formation of clusters 1, 2 and 3. Note that the *CR* index between the 3-cluster hard partitions given by, respectively, *MFCMdd-RWL-P* and *CARD-R*, and the known a priori 3-class partition, is 0.7865 and 0.3808. Consequently, the 3-cluster hard partitions given by these algorithms can be quite different.

3.3. Symbolic data sets

Symbolic data have been mainly studied in SDA, where very often an object represents a group of individuals, and the variables used to describe it need to assume a value that expresses the variability inherent to the description of a group. Thus, in SDA a variable can be interval-valued (it may assume as value an interval from a set of real numbers), set-valued (it may assume as value a set of categories), list-valued (it may assume as value an ordered list of categories), bar-chart-valued (it may assume as value a bar chart), or even histogram-valued (it may assume as value an histogram). SDA aims to introduce new methods as well as to extend classical data analysis techniques (clustering, factorial techniques, decision trees, etc.) to manage these kinds of data (sets of categories, intervals, histograms), called symbolic data [5–8]. SDA is then an area related to multivariate analysis, pattern recognition, and artificial intelligence.

This paper considers the following data sets described by symbolic (interval-valued and/or histogram-valued) variables: car and ecotoxicology data sets (<http://www.info.fundp.ac.be/asso/>) as well as a horse data set (<http://www.ceremade.dauphine.fr/fouati/sodas-pagegarde.htm>). The car and ecotoxicology data sets are described by a data matrix of “objects \times interval-valued attributes”. The horse data set is described by a data matrix of “objects \times attributes” where the attributes are interval-valued and bar-chart-valued.

Let $E = \{e_1, \dots, e_n\}$ be a set of n objects described by p symbolic variables. Each object e_i ($i = 1, \dots, n$) is represented as a vector $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ of symbolic features values x_{ij} ($j = 1, \dots, p$). If the j -th symbolic variable is interval-valued, the symbolic feature value is an interval, i.e., $x_{ij} = [a_{ij}, b_{ij}]$ with $a_{ij}, b_{ij} \in \mathbb{R}$ and $a_{ij} \leq b_{ij}$. However, if the j -th symbolic variable is bar-chart-valued, the symbolic feature value is a bar chart, i.e., $x_{ij} = (D_j, \mathbf{q}_{ij})$ ($i = 1, \dots, n; j = 1, \dots, p$) where D_j (the domain of the variable j) is a set of categories and $\mathbf{q}_{ij} = (q_{ij1}, \dots, q_{ijH_j})$ is a vector of weights.

A number of dissimilarity functions have been introduced in the literature for symbolic data to compare symbolic features values [17,18]. In this paper, we will consider suitable dissimilarity functions to compare a pair of objects (e_i, e_l) ($i, l = 1, \dots, n$) according to the pair (x_{ij}, x_{lj}) ($j = 1, \dots, p$) of symbolic feature values given by the j -th symbolic variable.

If the j -th symbolic variable is interval-valued, the dissimilarity between the pair of intervals $x_{ij} = [a_{ij}, b_{ij}]$ and $x_{lj} = [a_{lj}, b_{lj}]$ will be computed according to the function given in [23]

$$d_j(x_{ij}, x_{lj}) = [\max(b_{ij}, b_{lj}) - \min(a_{ij}, a_{lj})] - \left[\frac{(b_{ij} - a_{ij}) + (b_{lj} - a_{lj})}{2} \right] \quad (19)$$

If the j -th symbolic variable is bar-chart-valued, the dissimilarity between the pair of bar chart $x_{ij} = (D_j, \mathbf{q}_{ij}) = (D_j, (q_{ij1}, \dots, q_{ijH_j}))$ and $x_{lj} = (D_j, \mathbf{q}_{lj}) = (D_j, (q_{lj1}, \dots, q_{ljH_j}))$ will be computed according to the function given in [24]:

$$d_j(x_{ij}, x_{lj}) = 1 - \sum_{m=1}^{H_j} \sqrt{\left(\frac{q_{ijm}}{\sum_{m=1}^{H_j} q_{ijm}} \right) \left(\frac{q_{ljm}}{\sum_{m=1}^{H_j} q_{ljm}} \right)} \quad (20)$$

Note that despite the usefulness of these dissimilarity functions to compare interval-valued or bar-chart-valued symbolic data, they cannot be used in object-based clustering because they are not differentiable with respect to the prototype parameters.

Several dissimilarity matrices are obtained from these data matrices. Concerning the car and fish data sets, one of these dissimilarity matrices has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the interval-valued attributes, i.e., given two objects e_i and e_l , described, respectively by $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ and $\mathbf{x}_l = (x_{l1}, \dots, x_{lp})$, the dissimilarity between them taking into account simultaneously all the interval-valued attributes is computed as

$$d(\mathbf{x}_i, \mathbf{x}_l) = \sum_{j=1}^p d_j(x_{ij}, x_{lj}) \quad (21)$$

where $d_j(x_{ij}, x_{lj})$ is given by Eq. (19).

Table 12

Car data set: *CR* index.

$ G_k $	<i>SFCMdd</i>	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	0.2584	0.5889	0.5791	0.4931	0.6142	0.6142
2	0.2373	0.6142	0.6142	0.5654	0.6142	0.6332
3	0.2373	0.6142	0.6142	0.5257	0.6142	0.6142

The horse data set is described by interval-valued as well as bar-chart-valued attributes and so it is not suitable to produce a dissimilarity matrix having the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the attributes.

For all symbolic data sets, all the other dissimilarity matrices have the cells that are the dissimilarities between pairs of objects computed taking into account only a single attribute.

For the car and fish data sets, *NERF* and *SFCMdd* were performed on the dissimilarity matrix that has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the attributes. For all data sets, *CARD-R*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWG-S* were performed simultaneously on all dissimilarity matrices that have the cells that are the dissimilarities between pairs of objects computed taking into account only a single attribute.

All these dissimilarity matrices were also normalized according to their overall dispersion [22] to have the same dynamic range.

Each relational fuzzy clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) 100 times, and the best result was selected according to the adequacy criterion. The parameters m , T , and ε were set, respectively, to 2, 350, and 10^{-10} . The parameter s was set to 1 for *MFCMdd-RWL-P* and *MFCMdd-RWG-P*, and to 2 for *MFCMdd-RWL-S* and *MFCMdd-RWG-S*. The hard cluster partitions obtained from these fuzzy clustering methods were compared with the known a priori class partition. The comparison criterion used was the *CR* index, which was calculated for the best result.

3.3.1. Car data set

This data set consists of four types (classes) of cars. The classes (1-utility, 2-sedan, 3-sports, and 4-luxury) have 10, 8, 8 and 7 instances, respectively. Each car is described by 8 interval-valued attributes : (1) price, (2) engine capacity, (3) top speed, (4) acceleration, (5) step, (6) length, (7) width, and (8) height.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 4-cluster fuzzy partition. The 4-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 4-class partition. Table 12 shows the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S* and *MFCMdd-RWG-S* algorithms on the car data set according to *CR* index, considering prototypes of cardinality $|G_k| = 1, 2$, and 3 ($k = 1, 2, 3, 4$). *NERF* had 0.2543, whereas *CARD-R* had 0.5257 for *CR* index.

For this data set, the best performance was presented by *MFCMdd-RWG-S*, *MFCMdd-RWG-P*, *MFCMdd-RWL-P*, and *MFCMdd*. The *MFCMdd-RWL-S* and *CARD-R* algorithms also performed well on this data set. The worst performance was presented by *SFCMdd* and *NERF*. Moreover, the performance of *MFCMdd-RWG-S*, *MFCMdd-RWG-P*, *MFCMdd-RWL-P*, and *MFCMdd*, according to this index, was also superior to the performance presented by object-based fuzzy clustering algorithms with adaptive Euclidean distances, which learn a relevance weight globally for each variable ($CR = 0.499$ [25]) or locally for each variable and each cluster ($CR = 0.526$ [26]).

Table 13 gives the vector of relevance weights globally for all dissimilarity matrices (according to the result given by *MFCMdd-RWG-S* with prototypes of cardinality 2) and locally for each cluster and dissimilarity matrix (according to the results given by *MFCMdd-RWL-P* with prototypes of cardinality 2 and by the *CARD-R* algorithm).

Concerning the 4-cluster partition given by *MFCMdd-RWG-S*, dissimilarity matrices computed taking into account only “(8) height” and only “(1) price” attributes had the highest and the lowest relevance weight, respectively, in the definition of the fuzzy clusters.

Table 13 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-P* and *CARD-R* algorithms, each cluster (1, 2, 3 and 4) is associated to the same known a priori class.

Table 13

Car data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-S</i>	<i>MFCMdd-RWL-P</i>				<i>CARD-R</i>			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Price	0.1084	0.7409	0.6685	2.6030	1.3447	0.0712	0.0828	0.4945	0.1708
Engine capacity	0.1136	0.8587	0.7948	1.3894	1.1561	0.0792	0.1014	0.1326	0.1412
Top speed	0.1156	1.1680	1.4221	1.1149	1.0450	0.1353	0.2132	0.0925	0.1336
Acceleration	0.1288	1.5854	1.3830	0.6675	0.8053	0.2493	0.1900	0.0537	0.1016
Step	0.1384	0.5988	0.7489	0.8958	0.8269	0.1350	0.0687	0.0628	0.0925
Length	0.1267	1.6487	1.2328	0.7401	0.8198	0.1902	0.1120	0.0556	0.0979
Width	0.1195	0.9794	1.1231	0.8211	0.8915	0.0790	0.1369	0.0677	0.1132
Height	0.1486	0.8775	0.9226	0.6822	1.2643	0.0603	0.0945	0.0401	0.1488

Table 14

Ecotoxicology data set: *CR* index.

$ G_k $	<i>SFCMdd</i>	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	−0.1401	0.2489	0.2245	0.1606	0.2012	0.1171
2	0.0331	0.4880	0.4880	0.3949	0.4880	0.0266

It can be observed that for the fuzzy partition given by the *MFCMdd-RWL-P* algorithm, 3 dissimilarity matrices were relevant in the formation of cluster 1, 4 dissimilarity matrices were relevant in the formation of cluster 2, 3 dissimilarity matrices were relevant in the formation of cluster 3, and 4 dissimilarity matrices were relevant in the formation of cluster 4, whereas for the fuzzy partition given by the *CARD-R* algorithm, 4 dissimilarity matrices were relevant in the formation of cluster 1, 3 dissimilarity matrices were relevant in the formation of cluster 2, 2 dissimilarity matrices were relevant in the formation of cluster 3, and 4 dissimilarity matrices were relevant in the formation of cluster 4. Moreover, 3, 3, 2 and 4 dissimilarity matrices were simultaneously relevant in both partitions for, respectively, the formation of clusters 1, 2, 3 and 4. Note that the *CR* index between the 4-cluster hard partitions given by, respectively, *MFCMdd-RWL-P* and *CARD-R*, and the known a priori 4-class partition, is 0.6142 and 0.5257. Consequently, the corresponding clusters in each partition are supposed to be quite similar. This can explain the high number of variables which are simultaneously relevant in both partitions for, respectively, the formation of clusters 1, 2, 3 and 4.

3.3.2. Ecotoxicology data set

This data set concerns 12 species (classes) of fresh water fish, with each species described by 13 interval-valued attributes. These species are grouped into four a priori classes of unequal sizes according to diet: two classes (1-carnivorous, 2-detritivorous) of size 4 and two classes (3-omnivorous, 4-herbivorous) of size 2. Each fish is described by 13 interval-valued attributes: (1) length, (2) weight, (3) muscle, (4) intestine, (5) stomach, (6) gills, (7) liver, (8) kidneys, (9) liver/muscle, (10) kidneys/muscle, (11) gills/muscle, (12) intestine/muscle, and (13) stomach/muscle.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 4-cluster fuzzy partition. The 4-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 4-class partition. Table 14 shows the performance of the *SFCMdd*, *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S* and *MFCMdd-RWG-S* algorithms on the ecotoxicology data set according to *CR* index, considering prototypes of cardinality $|G_k| = 1$, and 2 ($k = 1, 2, 3, 4$). *NERF* had −0.1401, whereas *CARD-R* had 0.1606 for *CR* index.

For this data set, the best performance was presented by *MFCMdd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, and *MFCMdd-RWL-S*. *CARD-R* also performed quite well on this data set. The worst performance was presented by *SFCMdd* and *NERF*. Moreover, the performance of *MFCMdd*, *MFCMdd-RWL-P*, and *MFCMdd-RWG-P* with prototypes of cardinality 2 according to this index, was also superior to the performance presented by object-based fuzzy clustering algorithms with adaptive Euclidean distances that learn a relevance weight globally for each variable ($CR = 0.201$ [25]) or locally for each variable and each cluster ($CR = 0.274$ [25]).

Table 15

Ecotoxicology data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWL-P</i>				<i>CARD-R</i>			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Length	0.9199	1.1505	0.4777	0.8768	1.0519	0.0346	0.0160	0.0846	0.0290
Weight	0.8405	2.2442	0.2848	0.8297	0.9325	0.1726	0.0102	0.1278	0.0100
Muscle	1.0994	4.0608	0.9308	0.9655	0.7518	0.0551	0.0240	0.0871	0.0292
Intestine	1.1005	2.4509	0.8445	0.8597	0.9294	0.1353	0.0389	0.0635	0.0223
Stomach	0.9743	3.0993	0.4379	0.7308	0.8355	0.1431	0.0062	0.1304	0.0583
Gills	1.2068	1.7952	0.7454	1.1022	1.0023	0.1020	0.0145	0.1547	0.0213
Liver	1.0512	1.8715	0.7555	0.7577	0.8216	0.1847	0.0095	0.0238	0.1866
Kidneys	0.8557	3.3711	0.4556	0.5570	0.9869	0.1159	0.0358	0.0358	0.0488
Liver/muscle	1.0762	0.4340	2.5417	0.8674	0.8886	0.0177	0.1562	0.0129	0.0761
Kidneys/muscle	0.9287	0.2902	1.5922	0.8046	1.2491	0.0110	0.0963	0.0322	0.0281
Gills/muscle	1.0028	0.1192	3.8796	2.0996	2.2385	0.0058	0.3839	0.1111	0.0576
Intestine/muscle	1.0688	0.2703	2.0682	1.3147	0.6815	0.0133	0.1541	0.0435	0.0442
Stomach/muscle	0.9430	0.2728	2.5608	2.5265	1.2682	0.0082	0.0538	0.0920	0.3877

Table 15 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by *MFCMdd-RWG-P* with prototypes of cardinality 2) and locally for each cluster and dissimilarity matrix (according to the best results given by *MFCMdd-RWL-P* with prototypes of cardinality 2 and by the *CARD-R* algorithm).

Concerning the 4-cluster partition given by *MFCMdd-RWG-P*, dissimilarity matrices computed taking into account only “(6) gills” and only “(2) weight” attributes had the highest and the lowest relevance weight, respectively, in the definition of the fuzzy clusters.

Table 15 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-P* and *CARD-R* algorithms, each cluster (1, 2, 3 and 4) is associated to the same known a priori class.

It can be observed that for the fuzzy partition given by the *MFCMdd-RWL-S* algorithm, 8 dissimilarity matrices were relevant in the formation of cluster 1, 5 dissimilarity matrices were relevant in the formation of cluster 2, 4 dissimilarity matrices were relevant in the formation of cluster 3, and 5 dissimilarity matrices were relevant in the formation of cluster 4, whereas for the fuzzy partition given by the *CARD-R* algorithm, 6 dissimilarity matrices were relevant in the formation of cluster 1, 3 dissimilarity matrices were relevant in the formation of cluster 2, 7 dissimilarity matrices were relevant in the formation of cluster 3, and 2 dissimilarity matrices were relevant in the formation of cluster 4. Moreover, 6, 3, 3 and 1 dissimilarity matrices were simultaneously relevant in both partitions for, respectively, the formation of clusters 1, 2, 3 and 4. Note that the CR index between the 4-cluster hard partitions given by, respectively, *MFCMdd-RWL-S* and *CARD-R*, and the known a priori 4-class partition, is 0.4880 and 0.1606. Consequently, the 4-cluster hard partitions given by these algorithms can also be quite different.

3.3.3. Horse data set

This data set describes 12 horses. Each horse is described by seven interval-valued variables, namely, height at the withers (min), height at the withers (max), weight (min), weight (max), mares, stallions, and birth, and three histogram-valued variables, namely country, robe, and aptitude. The horses are grouped into four a priori classes: 1-racehorse, 2-leisure horse, 3-poney, and 4-draft horse; these classes have 4, 3, 3 and 2 instances, respectively.

The fuzzy clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a 4-cluster fuzzy partition. The 4-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori 4-class partition. Table 16 shows the performance of the *MFCM dd*, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, *MFCMdd-RWL-S* and *MFCMdd-RWG-S* algorithms on the horse data set according to CR index, considering prototypes of cardinality $|G_k| = 1$, and 2 ($k = 1, 2, 3, 4$). *CARD-R* had 0.2275 for this index.

For this data set, the best performance was presented by *MFCMdd-RWL-S*, *MFCMdd-RWG-P*, and *MFCMdd-RWL-P*. *CARD-R* also performed quite well on this data set. The worst performance was presented by *MFCMdd* and *MFCMdd-RWG-S*. Moreover, the performance of *MFCMdd-RWG-P*, *MFCMdd-RWL-S*, and *MFCMdd-RWL-P* according to this index, was also superior to the performance presented by object-based hard clustering algorithms with

Table 16

Horse data set: *CR* index.

$ G_k $	<i>MFCMdd</i>	<i>MFCMdd-RWL-P</i>	<i>MFCMdd-RWL-S</i>	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWG-S</i>
1	0.0946	0.3662	0.4252	0.3662	0.0946
2	0.0041	0.2510	0.3587	0.3294	0.0671

Table 17

Horse data set: vectors of relevance weights.

Data matrix	<i>MFCMdd-RWG-P</i>	<i>MFCMdd-RWL-S</i>				<i>CARD-R</i>			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Country	1.0180	0.0162	0.1104	0.0792	0.0462	0.0079	0.0989	0.0552	0.0976
Robe	0.8002	0.0127	0.0668	0.0754	0.0421	0.0057	0.0740	0.0493	0.1280
Ability	0.8082	0.0089	0.1549	0.0691	0.0332	0.0043	0.1201	0.0355	0.1460
Size (min)	0.9989	0.0428	0.0893	0.1989	0.1019	0.0318	0.1257	0.1571	0.0532
Size (max)	0.9453	0.0608	0.0998	0.1218	0.0944	0.3421	0.1309	0.1076	0.0399
Weight (min)	1.1582	0.4320	0.0760	0.1394	0.1305	0.2833	0.0985	0.2041	0.0602
Weight	1.0650	0.0182	0.1077	0.0678	0.1325	0.0090	0.0870	0.0681	0.1325
Mares	1.0745	0.0184	0.1090	0.0669	0.1388	0.0090	0.0874	0.0673	0.1436
Stallions	1.0801	0.0175	0.1114	0.0683	0.1360	0.0084	0.0902	0.0681	0.1388
Birth	1.1231	0.3720	0.0742	0.1127	0.1438	0.2981	0.0869	0.1873	0.0597

adaptive Euclidean distances, which learn a relevance weight globally for each variable ($CR = 0.209$ [27]) or locally for each variable and each cluster ($CR = 0.138$ [27]).

Table 17 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by *MFCMdd-RWG-P* with prototypes of cardinality 1) and locally for each cluster and dissimilarity matrix (according to the best result given by *MFCMdd-RWL-S* with prototypes of cardinality 1 and by the *CARD-R* algorithm).

Concerning the 4-cluster partition given by *MFCMdd-RWG-P*, dissimilarity matrices computed taking into account only “(6) weight (min)” and only “(2) robe” attributes had the highest and the lowest relevance weight, respectively, in the definition of the fuzzy clusters.

Table 17 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. In the partitions given by the *MFCMdd-RWL-S* and *CARD-R* algorithms, each cluster (1, 2, 3 and 4) is associated to the same known a priori class.

It can be observed that for the fuzzy partition given by the *MFCMdd-RWL-S* algorithm, 2 dissimilarity matrices were relevant in the formation of cluster 1, 5 dissimilarity matrices were relevant in the formation of cluster 2, 4 dissimilarity matrices were relevant in the formation of cluster 3, and 6 dissimilarity matrices were relevant in the formation of cluster 4, whereas for the fuzzy partition given by the *CARD-R* algorithm, 3 dissimilarity matrices were relevant in the formation of cluster 1, 3 dissimilarity matrices were relevant in the formation of cluster 2, 4 dissimilarity matrices were relevant in the formation of cluster 3, and 5 dissimilarity matrices were relevant in the formation of cluster 4. Moreover, 2, 1, 4 and 3 dissimilarity matrices were simultaneously relevant in both partitions for, respectively, the formation of clusters 1, 2, 3 and 4. Note that the *CR* index between the 4-cluster hard partitions given by, respectively, *MFCMdd-RWL-S* and *CARD-R*, and the known a priori 4-class partition, is 0.4252 and 0.2275. Consequently, the 4-cluster hard partitions given by these algorithms can also be quite different.

4. Concluding remarks

This paper introduced fuzzy k-medoids clustering algorithms that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. These matrices can be generated using different sets of variables and dissimilarity functions. These algorithms are designed to furnish a fuzzy partition and a prototype for each fuzzy cluster as well as a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fit between clusters and their representatives. As a particularity of these clustering algorithms, they assume that the prototype of each fuzzy cluster is a subset (of fixed cardinality) of the set of objects.

For each algorithm, the paper gives the solution for the best prototype of each fuzzy cluster, the best relevance weight of each dissimilarity matrix, and the best fuzzy partition according to the clustering criterion. Moreover, the convergence properties of these clustering algorithms are also presented. The relevance weights change at each algorithm iteration and can either be the same for all clusters or different from one cluster to another. Moreover, they are determined automatically in such a way that the closer the objects of a given dissimilarity matrix are to the prototype of a given fuzzy cluster, the higher is the relevance weight of this dissimilarity matrix on this fuzzy cluster.

The usefulness of these partitioning fuzzy k-medoids clustering algorithms was shown on synthetic as well as standard real-valued data sets, interval-valued data sets and mixed-feature (interval-valued and histogram-valued) symbolic data sets. The accuracy of these clustering algorithms was assessed by the *CR* index. Dissimilarity matrices were obtained from real-valued data sets through the Euclidean distance, whereas they were obtained from interval-valued data sets as well as mixed-feature (interval-valued and bar-chart-valued) symbolic data sets through non-standard dissimilarity functions, suitable to interval-valued as well as bar-chart-valued symbolic data, but which cannot be used in object-based clustering because they are not differentiable with respect to the prototype parameters.

Concerning the synthetic data sets, the performance of the *MFCMdd-RWL-P*, *MFCMdd-RWL-S*, *MFCMdd-RWG-P*, and *MFCMdd-RWG-S* fuzzy clustering algorithms depends on the dispersion of the variables that describe the objects. *MFCMdd-RWL-P* and *MFCMdd-RWL-S* were clearly superior in the synthetic data sets where the variance was different between the variables and from one class to another, whereas *MFCMdd-RWG-P* and *MFCMdd-RWG-S* were superior in the synthetic data sets where the variance was almost the same between the variables and different from one class to another and where the variance was almost the same between the variables and from one class to another.

Concerning the real-valued and the interval-valued data sets, the best performance globally, according to *CR* index, was presented by the fuzzy K-medoids clustering algorithms where the product of the relevance weights of the dissimilarity matrices is equal to one (*MFCMdd-RWL-P* and *MFCMdd-RWG-P*). As expected, the worst performance was presented by *NERF* and *SRFCM*, which were performed on the dissimilarity matrix that has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the attributes. Moreover, *MFCMdd-RWL-P*, *MFCMdd-RWG-P*, and *MFCMdd-RWL-S* also performed well on mixed feature-type symbolic data (horse data set). Finally, as the experimental results have shown, an increase in the cardinality of the prototypes does not necessarily improve the performance of the partitioning fuzzy K-medoids clustering algorithms with relevance weight for each dissimilarity matrix.

Acknowledgments

Authors are grateful to the anonymous referees for their careful revision, valuable suggestions, and comments which improved this paper. This research was partially supported by grants from CNPq (Brazilian Agency) and from a conjoint research project FACEPE (Brazilian Agency) and INRIA (France).

References

- [1] R. Xu, D. Wunsch, Survey of clustering algorithms, *IEEE Trans. Neural Networks* 16 (3) (2005) 645–678.
- [2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, *ACM Comput. Surv.* 31 (3) (1999) 264–323.
- [3] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [4] L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data*, Wiley, New York, 1990.
- [5] L. Billard, E. Diday, From the statistics of data to the statistics of knowledge: symbolic data analysis, *J. Am. Stat. Assoc.* 98 (462) (2003) 470–487.
- [6] L. Billard, E. Diday, *Symbolic Data Analysis. Conceptual Statistics and Data Mining*, Wiley, Chichester, 2006.
- [7] H.-H. Bock, E. Diday, *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*, Springer, Berlin, Heidelberg, 2000.
- [8] E. Diday, M. Noirhomme-Fraiture, *Symbolic Data Analysis and the Sodas Software*, Wiley, Chichester, 2008.
- [9] W. Pedrycz, Collaborative fuzzy clustering, *Pattern Recognition Lett.* 23 (2002) 675–686.
- [10] B. Leclerc, G. Cucumel, Consensus en classification: une revue bibliographique, *Math. Sci. Hum.* 100 (1987) 109–128.
- [11] H. Frigui, C. Hwang, F.C.-H. Rhee, Clustering and aggregation of relational data with applications to image database categorization, *Pattern Recognition* 40 (11) (2007) 3053–3068.
- [12] R.J. Hathaway, J.C. Bezdek, Nerf c-means: non-Euclidean relational fuzzy clustering, *Pattern Recognition* 27 (3) (1994) 429–437.
- [13] R. Krishnapuram, Anupam Joshi, Liyu Yi, A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering, in: *Proceedings of the IEEE International Fuzzy Systems Conference*, 1999, pp. 1281–1286.

- [14] Y. Lechevallier, Optimisation de quelques criteres en classification automatique et application a l'etude des modifications des proteines seriques en pathologie clinique. Thèse de 3eme cycle. Universite Paris-VI, 1974.
- [15] F.A.T. De Carvalho, M. Csernel, Y. Lechevallier, *Pattern Recognition Lett.* 30 (2009) 1037–1045.
- [16] E. Diday, G. Govaert, Classification automatique avec distances adaptatives, *R.A.I.R.O. Inf. Comput. Sci.* 11 (4) (1977) 329–349.
- [17] F. Exposito, D. Malerba, V. Tamma, Dissimilarity measures for symbolic objects, in: H.-H. Bock, E. Diday (Eds.), *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*, Springer, Berlin, Heidelberg, 2000, pp. 165–185.
- [18] A. Irpino, R. Verde, Dynamic clustering of interval data using a Wasserstein-based distance, *Pattern Recognition Lett.* 29 (11) (2007) 1648–1658.
- [19] E. Diday, J.C. Simon, Clustering analysis, in: K.S. Fu (Ed.), *Digital Pattern Classification*, Springer, Berlin, 1976, pp. 47–94.
- [20] L. Hubert, P. Arabie, Comparing partitions, *J. Classification* 2 (1985) 193–218.
- [21] G.W. Milligan, Clustering validation: results and implications for applied analysis, in: P. Arabie, L. Hubert, G. De Soete (Eds.), *Clustering and Classification*, Word Scientific, Singapore, 1996, pp. 341–375.
- [22] M. Chavent, Normalized k-means clustering of hyper-rectangles, in: *Proceedings of the XIth International Symposium of Applied Stochastic Models and Data Analysis (ASMDA 2005)*, Brest, France, 2005, pp. 670–677.
- [23] M. Ichino, H. Yaguchi, Generalized Minkowski metrics for mixed feature type data analysis, *IEEE Trans. Syst. Man Cybern.* 24 (4) (1994) 698–708.
- [24] H. Barcelar-Nicolau, The affinity coefficient, in: H.-H. Bock, E. Diday (Eds.), *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*, Springer, Berlin, Heidelberg, 2000, pp. 160–165.
- [25] F.A.T. De Carvalho, C.P. Tenorio, Fuzzy K-means clustering algorithms for interval-valued data based on adaptive quadratic distances, *Fuzzy Sets Syst.* 161 (23) (2010) 2978–2999.
- [26] F.A.T. De Carvalho, Fuzzy c-means clustering methods for symbolic interval data, *Pattern Recognition Lett.* 28 (4) (2007) 423–437.
- [27] F.A.T. De Carvalho, R.M.C.R. De Souza, Unsupervised pattern recognition models for mixed feature-type symbolic data, *Pattern Recognition Lett.* 31 (5) (2010) 430–443.