Neurocomputing 163 (2015) 115–123

Contents lists available at ScienceDirect

Neurocomputing

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

A multi-view relational fuzzy c-medoid vectors clustering algorithm

Francisco de A.T. de Carvalho [a,*], Filipe M. de Melo [a], Yves Lechevallier [b]

[a] Centro de Informática, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária, CEP 50740-560, Recife (PE), Brazil
[b] INRIA Paris-Rocquencourt, Domaine de Voluceau, BP 105, 78153 Le Chesnay, France

ARTICLE INFO

Article history:
Received 15 January 2014
Received in revised form
7 August 2014
Accepted 1 November 2014
Available online 14 April 2015

Keywords:
Fuzzy c-medoid vectors
Multi-view clustering
Collaborative clustering
Relational clustering
Relevance weights

ABSTRACT

This paper gives a multi-view relational fuzzy c-medoid vectors clustering algorithm that is able to partition objects taking into account simultaneously several dissimilarity matrices. The aim is to obtain a collaborative role of the different dissimilarity matrices in order to obtain a final consensus fuzzy partition. These matrices could have been obtained using different sets of variables and dissimilarity functions. This algorithm is designed to give a fuzzy partition and a vector of medoids for each fuzzy cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an objective function. These relevance weights change at each iteration of the algorithm and are different from one cluster to another. Moreover, various tools for interpreting the fuzzy partition and fuzzy clusters provided by this algorithm are also presented. Several examples illustrate the performance and usefulness of the proposed algorithm.

1. Introduction

Clustering methods seek to organize a set of items into clusters such that items within a given cluster have a high degree of similarity, whereas items belonging to different clusters have a high degree of dissimilarity. These methods have been widely applied in fields such as taxonomy, image processing, information retrieval, and data mining. Clustering techniques may be divided into hierarchical and partitioning methods [1,2].

Hierarchical methods yield complete hierarchy, i.e., a nested sequence of partitions of the input data. Hierarchical methods can be agglomerative and divisive. Agglomerative methods yield a sequence of nested partitions starting with trivial clustering in which each item is in a unique cluster and ending with trivial clustering in which all items are in the same cluster. A divisive method starts with all items in a single cluster and performs a splitting procedure until a stopping criterion is met (usually upon obtaining a partition of singleton clusters).

Partitioning methods seek to obtain a single partition of the input data into a fixed number of clusters. These methods often look for a partition that optimizes (usually locally) an objective function. To improve cluster quality, the algorithm is run multiple times with different starting points and the best configuration obtained from the total runs is used as the output clustering.

Partitioning methods can be divided into hard clustering and fuzzy clustering. Hard clustering provides a hard partition in which each object of the dataset is assigned to one and only one cluster. Fuzzy clustering [3] generates a fuzzy partition that provides a degree of membership of each object in a given cluster. This gives the flexibility to express objects that belong to more than one cluster at the same time [4].

Two usual representations of the objects upon which clustering can be based are 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. When each pair of objects is represented by a relationship, then 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. Another issue is multi-view data, where the objects are represented by several (feature or relational) data matrices. Multi-view data can be found in many domains such as bioinformatics, and marketing [5].

Several clustering models and algorithms have been proposed aiming to cluster feature data [1,2]. However, few clustering models have been proposed for relational data. Ref. [6] observed that several applications, such as content-based image retrieval, would be strongly benefited by clustering methods for relational data.

This paper gives a multi-view relational fuzzy c-medoid vectors clustering algorithm that is able to give a fuzzy partition of the 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 [7] aiming to obtain a final consensus partition [8]. These dissimilarity

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail addresses: fatc@cin.ufpe.br (F.d.A.T. de Carvalho), fmm@cin.ufpe.br (F.M. de Melo), yves.lechevallier@inria.fr (Y. Lechevallier).

http://dx.doi.org/10.1016/j.neucom.2014.11.083
0925-2312/

matrices can be computed using different sets of variables and a fixed dissimilarity function. In this case, the final partition gives a consensus between different views (sets of variables) describing the objects. They can also be computed from a fixed set of variables and different dissimilarity functions (in this case, the final partition gives the consensus between different dissimilarity functions) or even using different sets of variables and dissimilarity functions. Moreover, the influence of the different dissimilarity matrices is not equally important in the definition of the fuzzy clusters in the final fuzzy partition [6]. Thus, in order to obtain a meaningful fuzzy partition from all dissimilarity matrices, it is necessary to learn cluster dependent relevance weights for each dissimilarity matrix.

Frigui et al. [6] proposed CARD, a fuzzy clustering algorithm that is able to partition objects taking into account multiple dissimilarity matrices and that learns a relevance weight for each dissimilarity matrix in each cluster. CARD has two versions: CARD-R, which is based on NERF [9], and CARD-F, which is based on FANNY [4]. NERF and FANNY are well known fuzzy clustering algorithm for relational data.

The relational fuzzy clustering algorithm presented in this paper is a multi-view version of the single-view fuzzy c-medoids algorithm [10]. It is related to the fuzzy c-medoids clustering algorithms based on multiple dissimilarity matrices [11]. It is designed to give a fuzzy partition and a vector of medoids for each fuzzy cluster as well as to learn a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fitting between the fuzzy clusters and their representatives. These relevance weights change at each iteration of the algorithm and are different from one fuzzy cluster to another. The approach to compute the relevance weights of the dissimilarity matrices is 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 [12]. Moreover, it is also presented various tools for interpreting the fuzzy partition and the fuzzy clusters provided by the proposed algorithm.

This paper is organized as follows. Section 2 gives the multi-view relational fuzzy c-medoid vectors clustering algorithm, its convergence properties and time complexity, as well as fuzzy partition and fuzzy cluster interpretation tools. To illustrate the performance and usefulness of this clustering algorithm, experiments with several datasets (available at the UCI Repository [13]) are considered in Section 3. Finally, Section 4 gives the concluding remarks.

2. Multi-view relational fuzzy c-medoid vectors clustering algorithm

This section introduces the multi-view relational fuzzy c-medoid vectors clustering algorithm based on multiple dissimilarity matrices (here denoted MVFCMddV).

2.1. The objective function of the MVFCMddV multi-view clustering algorithm

Let $E=\{e_1,\dots,e_n\}$ be a set of n objects and let p dissimilarity $(n\times n)$ matrices $(\mathbf{D}_1,\dots,\mathbf{D}_j,\dots,\mathbf{D}_p)$, where $\mathbf{D}_j[i,l]=d_j(e_i,e_l)$ gives the dissimilarity between objects e_i and e_l on dissimilarity matrix $\mathbf{D}_j$. Each fuzzy cluster C_k $(k=1,\dots,K)$ has a representative element, called in this paper a medoid vector. Assume that $\mathbf{g}_k=(g_{k1},\dots,g_{kp})$ is the medoid vector of the fuzzy cluster C_k $(k=1,\dots,K)$, where each component belongs to the set of objects E, i.e., $\mathbf{g}_k\in E^p$ $(k=1,\dots,K)$, with $g_{kj}\in E$ $(j=1,\dots,p)$.

This multi-view relational fuzzy clustering algorithm provides a fuzzy partition $\mathcal{P}=(C_1,\dots,C_K)$ of E into K fuzzy clusters represented by a n-dimensional vector of membership degree vectors (one for each object) $\mathbf{U}=(\mathbf{u}_1,\dots,\mathbf{u}_n)$, a corresponding K-dimensional vector of medoid vectors (one for each fuzzy cluster) $\mathbf{G}=(\mathbf{g}_1,\dots,\mathbf{g}_K)$ representing the fuzzy clusters in the fuzzy partition $\mathcal{P}$, and a K-dimensional vector of relevance weight vectors (one for each fuzzy cluster) $\boldsymbol{\Lambda}=(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_K)$, by (locally) minimizing a suitable adequacy criterion (objective function). This criterion measures the adequacy between the clusters and their respective representatives and it is defined as

$$J(\mathbf{G},\boldsymbol{\Lambda},\mathbf{U})=\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m d_{\boldsymbol{\lambda}_k}(e_i,\mathbf{g}_k)=\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m\sum_{j=1}^{p}\lambda_{kj}d_j(e_i,g_{kj}) \quad (1)$$

in which

$$d_{\boldsymbol{\lambda}_k}(e_i,\mathbf{g}_k)=\sum_{j=1}^{p}\lambda_{kj}d_j(e_i,g_{kj}) \quad (2)$$

is the dissimilarity between the object e_i and the medoid vector $\mathbf{g}_k$ of fuzzy cluster C_k parameterized by the relevance weight vector $\boldsymbol{\lambda}_k=(\lambda_{k1},\dots,\lambda_{kp})$, where λ_{kj} $(j=1,\dots,p)$ is the relevance weight of the dissimilarity matrix $\mathbf{D}_j$ on fuzzy cluster C_k, $d_j(e_i,g_{kj})$ is the local dissimilarity between the object e_i and the j-th component g_{kj} of the medoid vector $\mathbf{g}_k$ on dissimilarity $\mathbf{D}_j$, u_{ik} is the membership degree of the object e_i on the 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 $\mathcal{P}$ as the sum of the homogeneities in each fuzzy cluster. This algorithm searches a fuzzy partition $\mathcal{P}^*=\{C_1^*,\dots,C_K^*\}$ of E into K fuzzy clusters represented by a n-dimensional vector of membership degree vectors $\mathbf{U}^*=(\mathbf{u}_1^*,\dots,\mathbf{u}_n^*)$, a corresponding K-dimensional vector of medoid vectors $\mathbf{G}^*=(\mathbf{g}_1^*,\dots,\mathbf{g}_K^*)$ representing the fuzzy clusters in the fuzzy partition $\mathcal{P}$, and a K-dimensional vector of relevance weight vectors $\boldsymbol{\Lambda}^*=(\boldsymbol{\lambda}_1^*,\dots,\boldsymbol{\lambda}_K^*)$, such that

$$J(\mathbf{G}^*,\boldsymbol{\Lambda}^*,\mathbf{U}^*)=\min\left\{J(\mathbf{G},\boldsymbol{\Lambda},\mathbf{U}):\mathbf{G}\in\mathbb{L}^K,\boldsymbol{\Lambda}\in\boldsymbol{\Xi}^K,\mathbf{U}\in\mathbb{U}^n\right\}, \quad (3)$$

where

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

2.2. The optimization steps of the MVFCMddV multi-view clustering algorithm

From an initial configuration $v_{(0)}=(\mathbf{G}^{(0)},\boldsymbol{\Lambda}^{(0)},\mathbf{U}^{(0)})$ and its associated objective function value $u_{(0)}=J(\mathbf{G}^{(0)},\boldsymbol{\Lambda}^{(0)},\mathbf{U}^{(0)})$, the algorithm generates a series $v_{(t)}=(\mathbf{G}^{(t)},\boldsymbol{\Lambda}^{(t)},\mathbf{U}^{(t)})\in\mathbb{L}^K\times\boldsymbol{\Xi}^K\times\mathbb{U}^n$ and $u_{(t)}=J(v_{(t)})=J(\mathbf{G}^{(t)},\boldsymbol{\Lambda}^{(t)},\mathbf{U}^{(t)})$, where $t=0,1,\dots$ is the iteration number. Each iteration has three steps. The algorithm iterates until it converges to a value of J corresponding to a local minimum:

Step 1: Search for the best medoid vectors.

In this step, the algorithm starts from the configuration $v_{(t-1)}=(\mathbf{G}^{(t-1)},\boldsymbol{\Lambda}^{(t-1)},\mathbf{U}^{(t-1)})$ and end with a configuration $(\mathbf{G}^{(t)},\boldsymbol{\Lambda}^{(t-1)},\mathbf{U}^{(t-1)})$, i.e., $\boldsymbol{\Lambda}^{(t-1)}=(\boldsymbol{\lambda}_1^{(t-1)},\dots,\boldsymbol{\lambda}_K^{(t-1)})$ and $\mathbf{U}^{(t-1)}=(\mathbf{u}_1^{(t-1)},\dots,\mathbf{u}_n^{(t-1)})$ are kept fixed while the cluster medoid vectors are updated.

Proposition 2.1. *The medoid vector* $\mathbf{g}_k^{(t)} = \mathbf{g}^* = (g_1^*, \ldots, g_p^*) \in E^p$ *of the fuzzy cluster* $C_k^{(t-1)}$ $(k = 1, \ldots, K)$, *which minimizes the clustering criterion J, is such that* $\sum_{i=1}^n (u_{ik}^{(t-1)})^m d_j(e_i, g_j^*) \longrightarrow \text{Min}$. *The medoid vector contains the components* $g_{kj}^{(t)} = g_j^* = e_l \in E$ *that are obtained using*

$$l = \underset{1 \le h \le n}{\operatorname{argmin}} \sum_{i=1}^{n} (u_{ik}^{(t-1)})^m \, d_j(e_i, e_h) \tag{4}$$

Proof. The proof of Proposition 2.1 is straightforward. □

Step 2: Computation of the best vectors of relevance weights.

This step starts with the configuration $(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t-1)}, \mathbf{U}^{(t-1)})$ and end with a configuration $(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t-1)})$, i.e., $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_K^{(t)})$ and $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \ldots, \mathbf{u}_n^{(t-1)})$ are kept fixed while it is computed new optimal vectors of relevance weights.

Proposition 2.2. *The vector of relevance weights* $\boldsymbol{\lambda}_k^{(t)} = (\lambda_{k1}^{(t)}, \ldots, \lambda_{kp}^{(t)})$ $(k = 1, \ldots, K)$, *under* $\lambda_{kj}^{(t)} > 0$ *and* $\prod_{j=1}^p \lambda_{kj}^{(t)} = 1$, *have their weights* $\lambda_{kj}^{(t)}$ $(j = 1, \ldots, p)$ *calculated according to*

$$\lambda_{kj}^{(t)} = \frac{\left\{ \prod_{h=1}^p \left[\sum_{i=1}^n (u_{ik}^{(t-1)})^m d_h(e_i, g_{kh}^{(t)}) \right] \right\}^{1/p}}{\left[\sum_{i=1}^n (u_{ik}^{(t-1)})^m d_j(e_i, g_{kj}^{(t)}) \right]} \tag{5}$$

Proof. The proof of Proposition 2.2 is given in Appendix A.

Remark. The relevance weight of a dissimilarity matrix $\mathbf{D}_j$ into a fuzzy cluster C_k has a high value when the similarity between the objects of this dissimilarity matrix $\mathbf{D}_j$ and the corresponding component of the medoid vector of this fuzzy cluster C_k is high.

Step 3: Computation of the best fuzzy partition.

This step starts with the configuration $(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t-1)})$ and end with a configuration $v_{(t)} = (\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$, i.e., $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_K^{(t)})$ and $\mathbf{\Lambda}^{(t)} = (\boldsymbol{\lambda}_1^{(t)}, \ldots, \boldsymbol{\lambda}_K^{(t)})$ are kept fixed while it is computed new optimal membership degrees.

Proposition 2.3. *The fuzzy partition represented by* $\mathbf{U}^{(t)} = (\mathbf{u}_1^{(t)}, \ldots, \mathbf{u}_n^{(t)})$, *where* $\mathbf{u}_i^{(t)} = (u_{i1}^{(t)}, \ldots, u_{iK}^{(t)})$, *which minimizes the clustering criterion J, is such that the membership degree* $u_{ik}^{(t)}$ $(i = 1, \ldots, n;\ k = 1, \ldots, K)$ *of each object* e_i *in each fuzzy cluster* C_k, *under* $u_{ik}^{(t)} \in [0, 1]$ *and* $\sum_{k=1}^K u_{ik}^{(t)} = 1$, *is calculated according to the following expression*:

$$u_{ik}^{(t)} = \left[\sum_{h=1}^{K} \left(\frac{d_{\boldsymbol{\lambda}_k^{(t)}}(e_i, \mathbf{g}_k^{(t)})}{d_{\boldsymbol{\lambda}_h^{(t)}}(e_i, \mathbf{g}_h^{(t)})} \right)^{1/(m-1)} \right]^{-1} = \left[\sum_{h=1}^{K} \left(\frac{\sum_{j=1}^p \lambda_{kj}^{(t)} d_j(e_i, g_{kj}^{(t)})}{\sum_{j=1}^p \lambda_{hj}^{(t)} d_j(e_i, g_{hj}^{(t)})} \right)^{1/(m-1)} \right]^{-1} \tag{6}$$

Proof. The proof of Proposition 2.3 follows the same scheme of that developed in the classical fuzzy c-means algorithm [3]. □

2.3. The algorithm

The multi-view relational fuzzy c-medoid vectors clustering algorithm with relevance weight for each dissimilarity matrix is executed according to the following steps.

MVFCMddV clustering algorithm

INPUT

p dissimilarity $(n \times n)$ matrices $(\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_p)$, where $\mathbf{D}_j[i, l] = d_j(e_i, e_l)$ gives the dissimilarity between objects e_i and e_l on dissimilarity matrix $\mathbf{D}_j$;

The number K of clusters $(2 \le K \ll n)$; The parameter m $(1 < m < \infty)$; The parameter T (an iteration limit); The parameter ϵ $(\epsilon > 0$ and $\epsilon \ll 1)$;

OUTPUT

The K-dimensional vector of medoid vectors $\mathbf{G}$, the K-dimensional vector of relevance weight vectors $\mathbf{\Lambda}$ and the n-dimensional vector of membership degree vectors $\mathbf{U}$

INITIALIZATION

Set $t \leftarrow 0$;

Randomly select K distinct medoid vectors $\mathbf{g}_k^{(t)} \in E^p (k = 1, \ldots, K)$ to obtain the vector of medoid vectors $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_K^{(t)})$;

Initially, the dissimilarity matrices have the same relevance weight: set $\boldsymbol{\lambda}_k^{(t)} = (\lambda_{k1}^{(t)}, \ldots, \lambda_{kp}^{(t)}) = (1, \ldots, 1)$ to obtain the vector of relevance weight vectors $\mathbf{\Lambda}^{(t)} = (\boldsymbol{\lambda}_1^{(t)}, \ldots, \boldsymbol{\lambda}_K^{(t)})$;

For each object e_i $(i = 1, \ldots, n)$ compute the component $u_{ik}^{(t)}$ of $\mathbf{u}_i^{(t)} = (u_{i1}^{(t)}, \ldots, u_{iK}^{(t)})$ according to Eq. (6) to obtain the vector of membership degree vectors $\mathbf{U}^{(t)} = (\mathbf{u}_1^{(t)}, \ldots, \mathbf{u}_n^{(t)})$;

From $v_{(t)} = (\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ compute $u_{(t)} = J(v_{(t)}) = J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (1);

REPEAT

Set $t \leftarrow t + 1$;

Step 1: Search for the Best Medoid Vectors

Compute the component $g_{kj}^{(t)}$ of $\mathbf{g}_k^{(t)} = (g_{k1}^{(t)}, \ldots, g_{kp}^{(t)})$ according to Eq. (4) to obtain the vector of medoid vectors $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_K^{(t)})$;

Step 2: Computation of the Best Vectors of Relevance Weights

Compute the component $\lambda_{kj}^{(t)}$ of $\boldsymbol{\lambda}_k^{(t)} = (\lambda_{k1}^{(t)}, \ldots, \lambda_{kp}^{(t)})$ according to Eq. (5) to obtain the vector of relevance weight vectors $\mathbf{\Lambda}^{(t)} = (\boldsymbol{\lambda}_1^{(t)}, \ldots, \boldsymbol{\lambda}_K^{(t)})$;

Step 3: Computation of the Best Fuzzy Partition

Compute the component $u_{ik}^{(t)}$ of $\mathbf{u}_i^{(t)} = (u_{i1}^{(t)}, \ldots, u_{iK}^{(t)})$ according to Eq. (6) to obtain the vector of membership degree vectors $\mathbf{U}^{(t)} = (\mathbf{u}_1^{(t)}, \ldots, \mathbf{u}_n^{(t)})$;

From $v_{(t)} = (\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ compute $u_{(t)} = J(v_{(t)}) = J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (1);

UNTIL $|u_{(t)} - u_{(t-1)}| < \epsilon$ OR $t > T$

2.4. Properties of the MVFCMddV algorithm

This section first shows the convergence properties of the MVFCMddV clustering algorithm, then its complexity is given.

2.4.1. Convergence properties

Following the method used in Diday and Simon [14], the convergence properties of this kind of algorithm can be studied from two series: $v_{(t)} = (\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)}) \in \mathbb{L}^K \times \mathbf{\Xi}^K \times \mathbb{U}^n$ and $u_{(t)} = J(v_{(t)}) = J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$, where $t = 0, 1, \ldots$ is the iteration number. From an initial term $v_{(0)} = (\mathbf{G}^{(0)}, \mathbf{\Lambda}^{(0)}, \mathbf{U}^{(0)})$, 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 2.4. *The series* $u_{(t)} = J(v_{(t)}) = J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ *decreases at each iteration and converges.*

Proof. The proof is given in Appendix B.

Proposition 2.5. *The series* $v_{(t)} = (\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ *converges.*

Proof. The proof is given in Appendix C.

2.4.2. Complexity analysis

The time complexity of MVFCMddV clustering algorithm can be analyzed considering the complexity of each single step. Let n be the number of objects, $K \ll n$ be the number of clusters and p be the number of dissimilarity matrices.

- *Initialization*: In this step, the initialization of the relevance weight vector costs $O(K \times p)$. The random selection of the medoid vector can be done selecting $K \times p$ objects of a random permutation of the n objects in $O(n+K \times p)$. Finally, the initialization of the membership corresponds to the step 3. The time complexity is $O(K \times n \times p)$. Thus, the initialization costs $O(K \times n \times p)$.
- *Step* 1: *Computation of the best medoid vector*. For each fuzzy cluster in a given table of dissimilarity it is needed to test each individual as a candidate for its respective medoid vector. This needs the computation of the distance between all pairs of objects using all p dissimilarity matrices and it costs $O(K \times n^2 \times p)$. The selection of the individuals for each medoid vector for a cluster is linear on the number of objects with cost $O(K \times p \times n)$. Thus, the step 1 costs $O(K \times n^2 \times p)$.
- *Step* 2: *Computation of the best relevance weight vectors*. According to Eq. (5), this step needs the computation of K denominators, the computation of the numerator just once, and to repeat that for each component of the vector of relevance weights. Thus, the step 2 costs $O(K \times n \times p+K \times p)$.
- *Step* 3: *Definition of the best fuzzy partition*. This step needs the computation of the dissimilarity between an individual i $(i=1,\dots,n)$ and the medoid vector of each cluster using the p dissimilarity matrices and it costs $O(K \times n \times p)$.

So, globally these steps cost $O(K \times n^2 \times p)$. Thus, if the clustering process needs t iterations to converge, the total time complexity of this algorithm is $O(K \times n^2 \times p \times t)$.

2.5. Fuzzy partition and fuzzy cluster interpretation

Partition and cluster interpretation tools allow the user to evaluate the overall heterogeneity of the data, the intra-cluster and inter-cluster data heterogeneity, and the contribution of each variable to the cluster formation. Indices for partition and cluster interpretation of quantitative data partitioned by the hard K-means algorithm are based on the decomposition of the overall dispersion into the overall dispersion within clusters plus the overall dispersion between clusters [15–17]. Based on the approach proposed by [18], in this section we give tools that help to interpret the fuzzy clusters and the fuzzy partition by considering corresponding suitable definitions of overall and within fuzzy clusters dispersion measures, as well as their corresponding decompositions according to clusters, views (dissimilarity matrices) and both clusters and views, even when this decomposition is not valid.

Let a fuzzy partition $\mathcal{P}=(C_1,\dots,C_K)$ of the set of objects E into K fuzzy clusters represented by the n-dimensional vector of membership degree vectors $\mathbf{U}=(\mathbf{u}_1,\dots,\mathbf{u}_n)$, a corresponding K-dimensional vector of medoid vectors $\mathbf{G}=(\mathbf{g}_1,\dots,\mathbf{g}_K)$ representing the fuzzy clusters in the fuzzy partition $\mathcal{P}$, and a K-dimensional vector of relevance weight vectors $\mathbf{\Lambda}=(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_K)$, provided by the MVFCMddV clustering algorithm.

The total fuzzy cluster dispersion T of the fuzzy partition $\mathcal{P}=(C_1,\dots,C_K)$ is the within-fuzzy-cluster dispersion J where the medoid vectors of each fuzzy cluster are replaced by a global medoid vector. It is defined as

$$\begin{aligned} T &= \sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m d_{\lambda_k}(e_i,\mathbf{g}) \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m \sum_{j=1}^{p}\lambda_{kj} d_j(e_i,g_j) \end{aligned} \tag{7}$$

where $\mathbf{g}$ is the global medoid vector.

Proposition 2.6. *The global medoid vector* $\mathbf{g}=\mathbf{g}^*=(g_1^*,\dots,g_p^*)\in E^p$, *which minimizes the total fuzzy cluster dispersion* T, *is such that* $\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}d_j(e_i,g_j^*)\longrightarrow \text{Min}$. *The global medoid vector contains the components* $g_j^{(t)}=g_j^*=e_l\in E$ *that are obtained using*

$$l=\underset{1\le h\le n}{\text{argmin}}\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}\, d_j(e_i,e_h) \tag{8}$$

Proof. The proof of Proposition 2.6 is straightforward. □

An important aspect of the proposed algorithm is that the representative of each fuzzy cluster (as well as the global representative) is a vector of objects from E, and not only a single object (as it is in Ref. [11]). Thus, to each view (dissimilarity matrix) and to each fuzzy cluster, an object of E is associated. Thanks to that, the total fuzzy cluster dispersion is decomposed according to

(a) the clusters: $T=\sum_{k=1}^{K}T_k$ with $T_k=\sum_{i=1}^{n}(u_{ik})^m\sum_{j=1}^{p}\lambda_{kj}d_j(e_i,g_j)$;
(b) the views (the dissimilarity matrices): $T=\sum_{j=1}^{p}T_j$ with $T_j=\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}d_j(e_i,g_j)$;
(c) the clusters and the views simultaneously: $T=\sum_{k=1}^{K}\sum_{j=1}^{p}T_{kj}$ with $T_{kj}=\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}d_j(e_i,g_j)$.

The within-fuzzy-cluster dispersion decomposes also according to

(a) the clusters: $J=\sum_{k=1}^{K}J_k$ with $J_k=\sum_{i=1}^{n}(u_{ik})^m\sum_{j=1}^{p}\lambda_{kj}d_j(e_i,g_{kj})$;
(b) the views: $J=\sum_{j=1}^{p}J_j$ with $J_j=\sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}d_j(e_i,g_{kj})$;
(c) the clusters and the views simultaneously: $J=\sum_{k=1}^{K}\sum_{j=1}^{p}J_{kj}$ with $J_{kj}=\sum_{i=1}^{n}(u_{ik})^m\lambda_{kj}d_j(e_i,g_{kj})$.

One can easily show that:

- $T\ge J$; $T_k\ge J_k$ $(k=1,\dots,K)$; $T_j\ge J_j$ $(j=1,\dots,p)$
- $T_{kj}\ge J_{kj}$ $(k=1,\dots,K; j=1,\dots,p)$.

Given the total fuzzy cluster dispersion, the within-fuzzy-cluster dispersion and their respective decompositions, the indices for the interpretation of the clusters and the partition introduced by [18] can be easily adapted to the MVFCMddV clustering algorithm.

3. Experimental evaluation

This section discusses the performance and usefulness of the MVFCMddV clustering algorithm in comparison with CARD-R [6] and MFCMdd-RWL-P [11] multi-view relational fuzzy clustering algorithms.

3.1. Performance of the clustering algorithms

To evaluate the performance of the MVFCMddV clustering algorithm in comparison with CARD-R and MFCMdd-RWL-P clustering algorithms, applications with several datasets selected from the UCI machine learning Repository [13] were considered.

The single-view relational fuzzy clustering algorithms FCMdd [10] and NERF [9] were also included in this study.

Eight benchmark datasets obtained from the UCI Machine Learning Repository, namely Abalone, Ecoli, Glass Identification, Image Segmentation, Iris Plant, Seeds, Thyroid Gland, and Wine, were considered in this study. Each dataset is described by a data matrix of "objects $\times$ real-valued variables". The real-valued variables of these datasets were previously standardized to have a mean of zero and a standard deviation of one. Table 1 (in which n represents the number of objects, p represents the number of variables and K represents the number of a priori clusters) briefly describes the data matrices considered.

Several dissimilarity matrices are obtained from each data matrix of Table 1. One of these dissimilarity matrices has the cells which are the dissimilarities between pairs of objects computed taking into account simultaneously all the real-valued variables. All the others dissimilarity matrices have the cells which are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute. In this paper, the dissimilarity between pairs of objects was computed according to the standard Euclidean distance.

All these dissimilarity matrices were then normalized to have the same overall dispersion of one [19]. This means that each dissimilarity $d(x_i, x_i')$ in a given dissimilarity matrix has been normalized into $d(x_i, x_i')/\Theta$, where $\Theta = \sum_{i=1}^{n} d(e_i, g)$ is the overall dispersion and $g = e_l \in E = \{e_1, \ldots, e_n\}$ is the overall medoid, which is computed according to $l = \text{argmin}_{1 \le h \le n} \sum_{i=1}^{n} d(e_i, e_h)$.

For each data matrix of Table 1, FCMdd and NERF single-view algorithms were performed on the dissimilarity matrix which has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously all the real-valued attributes. CARD-R, MFCMdd-RWL-P and MVFCMddV multi-view algorithms were performed simultaneously on all dissimilarity matrices which have the cells that are the dissimilarities between pairs of objects computed taking into account only a single real-valued attribute.

Each 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 and ϵ were set, respectively, equal to 2 and 10^{-10}. The maximum number of iterations N_{iter} was 350. For each dataset, the number of clusters was set equal to the number of a priori clusters.

The fuzzy partition given by each algorithm on each dataset was compared with the a priori partition. The comparison criteria used were the fuzzy Rand of Hullermeier and Rifqi index (FRHR) [20,21]. The FRHR index takes its values from the interval [0, 1], in which the value 1 indicates perfect agreement between fuzzy partitions, whereas values near 0 correspond to cluster agreement found by chance. The FRHR index was calculated for the best result.

A hard partition $Q = (Q_1, \ldots, Q_c)$ is obtained from each fuzzy partition by defining the hard cluster Q_i $(i = 1, \ldots, c)$ as $Q_i = \{e_k : u_{ik} \ge u_{mk} \; \forall m \in \{1, \ldots, c\}\}$. The hard cluster partitions obtained from these fuzzy clustering algorithms were compared with the known a priori class partition. The comparison criteria used were the corrected Rand index (CR) [22], as well as the F-measure [23] and the overall error rate of classification (OERC) [24]. These indices were calculated for the best result.

The CR index takes its values from the interval $[-1, 1]$, in which the value 1 indicates perfect agreement between hard partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance [25]. The F-measure index takes its values from the interval [0, 1], in which the value 1 indicates perfect agreement between hard partitions. The OERC index aims to measure the ability of a clustering algorithm to find out a priori classes present in a dataset and takes its values from the interval [0, 1] in which lower OERC values indicate better clustering results.

Tables 2 and 3 show the FRHR index, the CR index, the F-measure and the OERC computed to the fuzzy and hard partitions provided by the algorithms in comparison with the a priori partitions. In addition, it is shown (in parenthesis) the performance rank of each algorithm according to the indices and datasets considered.

Table 4 shows the average performance rank of the clustering algorithms according to the indices considered computed from Tables 2 and 3. It is also shown in Table 4 the performance rank of the clustering algorithms (in parenthesis) according to the average performance rank.

It can be observed that algorithm MVFCMddV presented the best average performance rank, concerning the fuzzy partitions in comparison with the a priori partition, according to the FRHR index. FCMdd was the worst. Moreover, the average performance rank of the MVFCMddV algorithm is the best or the second best whatever the index considered (CR, F-measure or OERC) concerning the comparison between the hard partitions and the a priori partition. For this comparison, CARD-R was the worst.

Table 1
Summary of the data matrices.

Data matrices	n	p	K
Abalone	4177	8	3
Ecoli	336	7	8
Glass Identification	214	9	6
Image Segmentation	2310	16	7
Iris Plant	150	4	3
Seeds	210	7	3
Thyroid gland	215	5	3
Wine	178	13	3

Table 2
Performance of the algorithms on the benchmark datasets considered.

Algorithms	Abalone FRHR index	CR index	F-measure	OERC
FCMdd	0.566 (5)	0.134 (3)	0.514 (5)	0.481 (3)
NERF	0.567 (4)	0.132 (4)	0.517 (3)	0.486 (4)
CARD-R	0.570 (3)	0.130 (5)	0.533 (1)	0.510 (5)
MFCMdd-RWL-P	0.571 (2)	0.137 (2)	0.516 (4)	0.480 (2)
MVFCMddV	0.571 (2)	0.137 (2)	0.517 (3)	0.480 (2)
	Ecoli			
FCMdd	0.606 (4)	0.348 (5)	0.575 (5)	0.220 (4)
NERF	0.623 (3)	0.393 (4)	0.639 (4)	0.163 (1)
CARD-R	0.384 (5)	0.469 (3)	0.643 (3)	0.339 (5)
MFCMdd-RWL-P	0.685 (1)	0.532 (1)	0.738 (1)	0.184 (2)
MVFCMddV	0.670 (2)	0.486 (2)	0.670 (2)	0.214 (3)
	Glass			
FCMdd	0.527 (3)	0.215 (1)	0.532 (1)	0.359 (1)
NERF	0.519 (5)	0.155 (3)	0.446 (4)	0.453 (3)
CARD-R	0.524 (4)	0.007 (5)	0.375 (5)	0.612 (5)
MFCMdd-RWL-P	0.543 (2)	0.135 (4)	0.510 (2)	0.518 (4)
MVFCMddV	0.554 (1)	0.195 (2)	0.475 (3)	0.443 (2)
	Image Segmentation			
FCMdd	0.576 (4)	0.478 (4)	0.642 (4)	0.364 (2)
NERF	0.617 (3)	0.538 (1)	0.711 (1)	0.288 (1)
CARD-R	0.341 (5)	0.004 (5)	0.245 (5)	0.810 (5)
MFCMdd-RWL-P	0.779 (1)	0.493 (2)	0.650 (3)	0.380 (3)
MVFCMddV	0.776 (2)	0.489 (3)	0.653 (2)	0.388 (4)

Table 3
Performance of the algorithms on the benchmark datasets considered.

Algorithms	Iris Plant FRHR index	CR index	*F*-measure	OERC
FCMdd	0.743 (4)	0.641 (4)	0.846 (4)	0.153 (4)
NERF	0.738 (5)	0.630 (5)	0.839 (5)	0.160 (5)
CARD-R	0.885 (1)	0.885 (1)	0.959 (1)	0.040 (1)
MFCMdd-RWL-P	0.838 (2)	0.850 (2)	0.946 (2)	0.053 (2)
MVFCMddV	0.829 (3)	0.757 (3)	0.905 (3)	0.093 (3)
	Seeds			
FCMdd	0.718 (5)	0.746 (2)	0.910 (2)	0.090 (2)
NERF	0.721 (4)	0.772 (1)	0.919 (1)	0.081 (1)
CARD-R	0.766 (1)	0.637 (5)	0.856 (5)	0.142 (5)
MFCMdd-RWL-P	0.744 (3)	0.641 (4)	0.861 (4)	0.138 (4)
MVFCMddV	0.754 (2)	0.666 (3)	0.870 (3)	0.128 (3)
	Thyroid Gland			
FCMdd	0.736 (4)	0.597 (4)	0.866 (4)	0.120 (4)
NERF	0.741 (3)	0.659 (3)	0.888 (3)	0.102 (3)
CARD-R	0.621 (5)	0.229 (5)	0.716 (5)	0.218 (5)
MFCMdd-RWL-P	0.806 (2)	0.863 (2)	0.956 (2)	0.041 (2)
MVFCMddV	0.815 (1)	0.877 (1)	0.961 (1)	0.037 (1)
	Wine			
FCMdd	0.571 (5)	0.755 (3)	0.914 (3)	0.084 (3)
NERF	0.601 (4)	0.897 (1)	0.966 (1)	0.033 (1)
CARD-R	0.657 (1)	0.380 (5)	0.722 (5)	0.269 (5)
MFCMdd-RWL-P	0.607 (3)	0.728 (4)	0.903 (4)	0.095 (4)
MVFCMddV	0.646 (2)	0.756 (2)	0.914 (3)	0.084 (3)

Table 4
Average performance ranking.

Algorithms	FRHR	CR index	*F*-measure	OERC
FCMdd	4.25 (5)	3.25 (4)	3.50 (4)	2.87 (4)
NERF	3.87 (4)	2.75 (3)	2.75 (3)	2.37 (1)
CARD-R	3.12 (3)	4.25 (5)	3.75 (5)	4.50 (5)
MFCMdd-RWL-P	2.00 (2)	2.62 (2)	2.75 (3)	2.87 (4)
MVFCMddV	1.87 (1)	2.25 (1)	2.50 (1)	2.62 (2)

3.2. Application: the thyroid gland dataset

To introduce the partition and cluster interpretation indices, we consider the previous results provided by the MVFCMddV algorithm applied to the thyroid gland dataset. Additional information as the confusion matrix and the relevance weights of the views (dissimilarity matrices) on each cluster are also provided in this section.

This dataset consists of three a priori clusters concerning the state of the thyroid gland: normal (1), hyperthyroidism (2) and hypothyroidism (3). The a priori clusters (1, 2 and 3) have, respectively, 150, 35 and 30 instances. Each object is described by five real-valued attributes: (1) T3-resin uptake test, (2) Total serum thyroxin, (3) Total serum triiodo thyronine, (4) Basal thyroid stimulating hormone (TSH) and (5) Maximal absolute difference in TSH value.

Table 5 shows the confusion matrix provided by the MVFCMddV algorithm. The clusters 1, 2 and 3 correspond, respectively, to the a priori clusters 1 (normal), 2 (hyperthyroidism) and 3 (hypothyroidism). It can be observed the very good performance of the MVFCMddV algorithm on this dataset.

Table 6 shows the relevance weights of the views (dissimilarity matrices) on each cluster. It can be observed that the relevance weights of the dissimilarity matrices "4-Basal thyroid stimulating hormone (TSH)" and "5-Maximal absolute difference in TSH value" were very high in cluster 2. Thus, this cluster is very homogeneous on these dissimilarity matrices, the objects with high membership degrees on this cluster have a high degree of similarity with the component of the medoid vector of this cluster.

Table 5
Confusion matrix for the hard partition provided by the MVFCMddV algorithm.

Clusters	A priori clusters		
	1	2	3
1	148	0	6
2	2	35	0
3	0	0	24

Table 6
The relevance weights of the views (dissimilarity matrices) provided by the algorithm.

Views	Clusters		
	1	2	3
1	0.24631	0.07987	1.74974
2	0.44456	0.15365	4.71427
3	0.85773	0.05816	4.99824
4	9.48280	33.53862	0.15207
5	1.12277	41.77008	0.15949

3.2.1. Partition interpretation

The overall heterogeneity index is given by $Q(\mathcal{P}) = (T-J)/T = 1-J/T$. This index takes its values between 0 and 1 and a value of $Q(\mathcal{P})$ close to 1 means more homogeneous clusters and a better representation of the objects of a fuzzy cluster by its medoid vector. The fuzzy partition provided by the MVFCMddV algorithm achieved a quality index $Q(\mathcal{P})$ equal to 56.44%. This means that the set of dissimilarity matrices have a moderate discriminant power, the fuzzy clusters medoid vectors are quite similar to the overall medoid vector.

The overall heterogeneity index concerning the j-th dissimilarity matrix is measured by $Q_j(\mathcal{P}) = (T_j - J_j)/T_j = 1 - J_j/T_j$, $j = 1, \ldots, p$. This index also takes its values between 0 and 1 and a value of $Q_j(\mathcal{P})$ close to 1 denotes better quality of a fuzzy partition concerning the j-th dissimilarity matrix. This means that the j-th dissimilarity matrix has a high discriminant power, the dissimilarity between the j-th component of the overall medoid vector and the j-th component of the fuzzy clusters medoid vectors is very high.

From Table 7 we can observe, by comparing the values of $Q_j(\mathcal{P})$ with the value of $Q(\mathcal{P})$, that the discriminant power of the dissimilarity matrix "2-Total serum thyroxin" is above the average discriminant power of the set of dissimilarity matrices.

3.2.2. Cluster interpretation

The relative contribution of the fuzzy cluster C_k to the overall within-fuzzy-cluster dispersion is given by $J(k) = J_k/J$, $k = 1, \ldots, K$. Note that $0 \le J(k) \le 1$ and $\sum_{k=1}^{K} J(k) = 1$. A relatively large value of $J(k)$ indicates that the fuzzy cluster C_k is relatively heterogeneous in comparison with the other fuzzy clusters.

The quality of a fuzzy cluster C_k ($k = 1, \ldots, K$) is measured by $Q(C_k) = (T_k - J_k)/T_k = 1 - J_k/T_k$, $k = 1, \ldots, K$. This index measures the gain in homogeneity of the fuzzy cluster C_k obtained when replacing the overall medoid vector by the cluster-specific medoid vector. A value of $Q(C_k)$ close to 0 means that the fuzzy cluster medoid vector is very similar to the overall medoid vector and

Table 7
Quality of the partition concerning the single dissimilarity matrices ($Q_j(\mathcal{P})$) for the thyroid gland dataset (%).

Relevance weight of the dissimilarity matrices	$Q_j(\mathcal{P})$
1-T3-resin uptake test	46.13
2-Total serum thyroxin	74.05
3-Total serum triiodo thyronine	43.18
4-Basal thyroid stimulating hormone (TSH)	43.41
5-Maximal absolute difference in TSH value	55.40

Table 8
Cluster heterogeneity indices for the thyroid gland dataset provided by the MVFCMddV algorithm (%).

Cluster	$J(k)$	$Q(C_k)$
1	38.96	49.10
2	16.34	66.12
3	44.69	57.35

Table 9
Quality of clusters in the dissimilarity matrices (%).

Dissimilarity matrices	Clusters		
	1	2	3
1-T3-resin uptake test	38.29	58.43	46.27
2-Total serum thyroxin	71.36	86.03	66.22
3-Total serum triiodo thyronine	31.51	60.77	42.29
4-Basal thyroid stimulating hormone (TSH)	4.77	16.30	61.56
5-Maximal absolute difference in TSH value	54.57	31.04	61.05

denotes a fuzzy cluster of poor quality. A value of $Q(C_k)$ close to 1 indicates a fuzzy cluster of high quality. From Table 8, we can see that fuzzy cluster 2 is the most homogeneous and has the best quality index.

The quality of a fuzzy cluster C_k $(k=1,\ldots,K)$ concerning the j-th dissimilarity matrix is measured by $Q_j(C_k)=(T_{kj}-J_{kj})/T_{kj}=1-J_{kj}/T_{kj},\ k=1,\ldots,K;\ j=1,\ldots,p$. This index also takes its values between 0 and 1. This index measures the gain in homogeneity of the fuzzy cluster C_k for the j-th dissimilarity matrix obtained when replacing the j-th component of the overall medoid vector by the j-th component of the cluster-specific medoid vector. A value of $Q_j(C_k)$ close to 1 denotes a high quality of the fuzzy cluster C_k regarding the j-th dissimilarity matrix. Moreover, this index helps the user to find the dissimilarity matrices that characterize the cluster C_k. The comparison with the index $Q(C_k)$ helps the interpretation of the index $Q_j(C_k)$: the j-th dissimilarity matrix characterizes the cluster P_k if $Q_j(C_k)>Q(C_k)$.

Table 9 shows the cluster heterogeneity index concerning the single dissimilarity matrices. Comparing the values of $Q_j(C_k)$ with the values of $Q(C_k)$, $k=1,\ldots,K; j=1,\ldots,p$ (see Table 8), we can see that dissimilarity matrices “2-Total serum thyroxin”, “4-Basal thyroid stimulating hormone (TSH)” and “5-Maximal absolute difference in TSH value” characterize fuzzy cluster 3 (hypothyroidism). Moreover, dissimilarity matrices “2-Total serum thyroxin” and “5-Maximal absolute difference in TSH value” characterize fuzzy cluster 1 (normal), whereas fuzzy cluster 2 (hyperthyroidism) is characterized by dissimilarity matrix “2-Total serum thyroxin”.

4. Concluding remarks

In this paper, the MVFCMddV fuzzy c-medoid vectors clustering algorithm for relational data that is able to partition objects taking into account simultaneously their relational description given by multiple dissimilarity matrices was presented. These dissimilarity matrices were computed using different sets of variables and dissimilarity functions. The aim was to obtain a final fuzzy partition giving a consensus between different views (dissimilarity matrices) describing the objects.

This multi-view clustering algorithm gives a fuzzy partition of the input data, a corresponding medoid vector for each cluster and a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fitting between the fuzzy clusters and their representatives (medoid vectors). These relevance weights change at each iteration of the algorithm and are different from one fuzzy cluster to another. They are computed in such a way that they have a high value if the homogeneity of the objects of a given dissimilarity matrix in a given fuzzy cluster is high. Moreover, various suitable tools for interpreting the fuzzy partition and the fuzzy clusters provided by this algorithm were also presented.

The proposed clustering algorithm was run on several real-valued datasets coming from the UCI machine learning repository with different number of objects, variables and a priori clusters, in comparison with useful single-view and multi-view relational fuzzy clustering algorithms. The real-valued variables of these datasets were previously standardized to have a mean of zero and a standard deviation of one. The dissimilarity matrices were computed from these datasets according to the classical Euclidean distance, but the computation with other dissimilarity functions, such as the City-Block and Mahalanobis distances, can also be considered.

In this study, the single-view algorithms were run on a single dissimilarity matrix where the dissimilarity between the objects was computed taking into account simultaneously all the real-valued variables of a given dataset. The multi-view algorithms were run simultaneously on several dissimilarity matrices where the dissimilarity between the objects was computed taking into account only a single real-valued attribute of a given dataset.

Concerning the results, the MVFCMddV algorithm presented the best average performance rank on these datasets regarding the FRHR index. This index compares the fuzzy partition provided by the MVFCMddV with the a priori partition of the dataset. Moreover, its average performance rank was the best or the second best whatever the index considered (CR, *F*-measure or OERC) regarding the comparison between the hard partitions and the a priori partition. Finally, the example with the thyroid gland dataset illustrated the usefulness and the merit of the proposed indices to interpreting the fuzzy partition and the fuzzy clusters provided by this algorithm.

Acknowledgments

The authors are grateful to the anonymous referees for their careful revision, valuable suggestions, and comments which improved this paper. The authors would like to thank FACEPE (Research Agency from the State of Pernambuco, Brazil) and CNPq (National Council for Scientific and Technological Development, Brazil) for their financial support.

Appendix A. Proof of Proposition 2.2

The vectors of relevance weights $\boldsymbol{\lambda}_k^{(t)}=(\lambda_{k1}^{(t)},\ldots,\lambda_{kp}^{(t)})$ $(k=1,\ldots,K)$, under $\lambda_{kj}^{(t)}>0$ and $\prod_{j=1}^{p}\lambda_{kj}^{(t)}=1$, have their weights $\lambda_{kj}^{(t)}$ $(j=1,\ldots,p)$

calculated according to

$$\lambda_{kj}^{(t)} = \frac{\left\{ \prod_{h=1}^{p} \left[\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_h(e_i, g_{kh}^{(t)}) \right] \right\}^{1/p}}{\left[\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_j(e_i, g_{kj}^{(t)}) \right]}.$$

Proof. As the fuzzy partition represented by $\mathbf{U}^{(t-1)} = (\mathbf{u}_1^{(t-1)}, \dots, \mathbf{u}_n^{(t-1)})$ and the vector of medoid vectors $\mathbf{G}^{(t)} = (\mathbf{g}_1^{(t)}, \dots, \mathbf{g}_K^{(t)})$ are kept fixed, we can rewrite the criterion J as $J(\mathbf{\Lambda}^{(t)}) = J(\lambda_1^{(t)}, \dots, \lambda_K^{(t)}) = \sum_{k=1}^{K} J_k(\lambda_k^{(t)})$ with $J_k(\lambda_k^{(t)}) = J_k(\lambda_{k1}^{(t)}, \dots, \lambda_{kp}^{(t)}) = \sum_{j=1}^{p} \lambda_{kj}^{(t)} J_{kj}$, where $J_{kj} = \sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_j(e_i, g_{kj}^{(t)})$.

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

$$\lambda_{kj}^{(t)} = \frac{(\Pi_{h=1}^{p} J_{kh})^{1/p}}{J_{kj}} = \frac{\left\{ \prod_{h=1}^{p} \left(\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_h(e_i, g_{kh}^{(t)}) \right) \right\}^{1/p}}{\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_j(e_i, g_{kj}^{(t)})}.$$

Thus, an extreme value of J_k is reached when $J_k(\lambda_{k1}^{(t)}, \dots, \lambda_{kp}^{(t)}) = p\{J_{k1} \times \dots \times J_{kp}\}^{1/p}$. 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., $(1/p)\left(J_{k1} + \dots + J_{kp}\right) > \{J_{k1} \times \dots \times J_{kp}\}^{1/p}$ (the equality holds only if $J_{k1} = \dots = J_{kp}$), we conclude that this extreme is a minimum value. □

Appendix B. Proof of Proposition 2.4

The series $u_{(t)} = J(v_{(t)}) = J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)})$ decreases at each iteration and converges.

Proof. We will first show that the inequalities (I), (II) and (III) below hold (i.e., the series decreases at each iteration)

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

The inequality (I) holds because $J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik}^{(t)})^m d_{\lambda_k^{(t)}}(e_i, \mathbf{g}_k^{(t)})$, and $J(\mathbf{G}^{(t+1)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik}^{(t)})^m d_{\lambda_k^{(t)}}(e_i, \mathbf{g}_k^{(t+1)})$, and according to Proposition 2.1,

$$\mathbf{G}^{(t+1)} = (\mathbf{g}_1^{(t+1)}, \dots, \mathbf{g}_K^{(t+1)}) = \underset{\mathbf{G} = (\mathbf{g}_1, \dots, \mathbf{g}_K) \in \mathbb{L}^K}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik}^{(t)})^m d_{\lambda_k^{(t)}}(e_i, \mathbf{g}_k).$$

Moreover, inequality (II) holds because $J(\mathbf{G}^{(t+1)}, \mathbf{\Lambda}^{(t+1)}, \mathbf{U}^{(t)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik}^{(t)})^m d_{\lambda_k^{(t+1)}}(e_i, \mathbf{g}_k^{(t+1)})$, and according to Proposition 2.2,

$$\mathbf{\Lambda}^{(t+1)} = (\lambda_1^{(t+1)}, \dots, \lambda_K^{(t+1)}) = \underset{\mathbf{\Lambda} = (\lambda_1, \dots, \lambda_K) \in \Xi^K}{\arg\min} \sum_{i=1}^{n} (u_{ik}^{(t)})^m d_{\lambda_k}(e_i, \mathbf{g}_k^{(t+1)}).$$

The inequality (III) also holds because $J(\mathbf{G}^{(t+1)}, \mathbf{\Lambda}^{(t+1)}, \mathbf{U}^{(t+1)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik}^{(t+1)})^m d_{\lambda_k^{(t+1)}}(e_i, \mathbf{g}_k^{(t+1)})$, and according to Proposition 2.3,

$$\mathbf{U}^{(t+1)} = (\mathbf{u}_1^{(t+1)}, \dots, \mathbf{u}_n^{(t+1)}) = \underset{\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)}{\operatorname{argmin}} = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik})^m d_{\lambda_k^{(t+1)}}(e_i, \mathbf{g}_k^{(t+1)}).$$

Finally, because the series $u_{(t)}$ decreases and it is bounded ($J(v_{(t)}) \geq 0$), it converges. □

Appendix C. Proof of Proposition 2.5

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 that $J(\mathbf{G}^{(t)}, \mathbf{\Lambda}^{(t)}, \mathbf{U}^{(t)}) = J(\mathbf{G}^{(t+1)}, \mathbf{\Lambda}^{(t+1)}, \mathbf{U}^{(t+1)})$ and this equality, according to Proposition 2.4, can be rewritten as the equalities (I)–(III):

$$\underbrace{J(\mathbf{G}^{(T)}, \mathbf{\Lambda}^{(T)}, \mathbf{U}^{(T)})}_{u_{(T)}} \overbrace{=}^{\text{(I)}} J(\mathbf{G}^{(T+1)}, \mathbf{\Lambda}^{(T)}, \mathbf{U}^{(T)}) \overbrace{=}^{\text{(II)}} J(\mathbf{G}^{(T+1)}, \mathbf{\Lambda}^{(T+1)}, \mathbf{U}^{(T)})$$
$$\overbrace{=}^{\text{(III)}} \underbrace{J(\mathbf{G}^{(T+1)}, \mathbf{\Lambda}^{(T+1)}, \mathbf{U}^{(T+1)})}_{u_{(T+1)}}$$

From the first equality (I), we have that $\mathbf{G}^{(T)} = \mathbf{G}^{(T+1)}$ because $\mathbf{G}$ is unique minimizing J when the fuzzy partition represented by $\mathbf{U}^{(T)}$ and the vector of relevance weight vectors $\mathbf{\Lambda}^{(T)}$ are kept fixed. From the second equality (II), we have that $\mathbf{\Lambda}^{(T)} = \mathbf{\Lambda}^{(T+1)}$ because $\mathbf{\Lambda}$ is unique minimizing J when the fuzzy partition represented by $\mathbf{U}^{(T)}$ and the vector of medoid vectors $\mathbf{G}^{(T+1)}$ are kept fixed. Furthermore, from the third equality (III), we have that $\mathbf{U}^{(T)} = \mathbf{U}^{(T+1)}$ because $\mathbf{U}$ is unique minimizing J when the vector of medoid vectors $\mathbf{G}^{(T+1)}$ and the vector of relevance weight vectors $\mathbf{\Lambda}^{(T+1)}$ are kept fixed.

Therefore 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. □

References

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

[2] R. Xu, D.I.I. Wunusch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.

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

[4] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, Hoboken, NJ, 2005.

[5] G. Cleuziou, M. Exbrayat, L. Martin, J.-H. Sublemontier, Cofkm: a centralized method for multiple-view clustering, in: Proceedings of 2009 Ninth IEEE International Conference on Data Mining, 2009, pp. 752–757.

[6] H. Frigui, C. Hwang, F. Chung-Hoon Rhee, Clustering and aggregation of relational data with applications to image database categorization, Pattern Recognit. 40 (2007) 3053–3068.

[7] W. Pedrycz, Collaborative fuzzy clustering, Pattern Recognit. Lett. 23 (2002) 675–686.

[8] B. Leclerc, G. Cucumel, Concensus en classification: une revue bibliographic, Math. Sci. Hum. 100 (1987) 109–128.

[9] R. Hathaway, J. Bezdek, Nerf c-means: non-Euclidean relational fuzzy clustering, Pattern Recognit. 27 (1994) 429–437.

[10] R. Krishnapuram, A. Joshi, L. 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.

[11] F.A.T. de Carvalho, Y. Lechevallier, F.M. de Melo, Relational partitioning fuzzy clustering based on multiple dissimilarity matrices, Fuzzy Sets Syst. 215 (2013) 1–28.

[12] E. Diday, G. Govaert, Classification automatique avec distances adaptatives, R.A.I.R.O. Inf. Comput. Sci. 11 (1977) 329–349.

[13] A. Frank, A. Asuncion, UCI machine learning repository, 2010. URL: ⟨http://archive.ics.uci.edu/ml⟩.

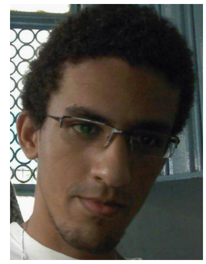
[14] E. Diday, J.C. Simon, Digital Pattern Classification, Springer, Berlin (1976) 47–94.

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

[16] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, NJ, 1988.

[17] B. Mirkin, Mathematical Classification and Clustering, Kluwer Academic Publishers, Dordrecht, Boston, London, 1996.
[18] 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 (2010) 2978–2999.
[19] M. Chavent, J. Saracco, On central tendency and dispersion measures for intervals and hypercubes, Commun. Stat. Theory Methods 37 (2008) 1471–1482.
[20] D.T. Anderson, J. Bezdek, M. Popescu, J.M. Keller, Comparing fuzzy, probabilistic, and possibilistic partitions, IEEE Trans. Fuzzy Syst. 18 (2010) 906–918.
[21] E. Hullermeier, M. Rifqi, A fuzzy variant of the rand index for comparing clustering structures, in: Proceedings of the Joint 2009 IFSA World Congress and 2009 EUSFLAT Conference, 2009, pp. 1–5.
[22] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[23] R.A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
[24] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, Chapman & Hall CRC, Boca Raton, 1984.
[25] G.W. Milligan, Clustering and Classification, World Scientific, Singapore (1996) 341–375.

Francisco de A.T. de Carvalho received the Ph.D. degree in Computer Science in 1992 from Institut National de Recherche en Informatique et en Automatique (INRIA) and Université Paris-IX Dauphine, France. From 1992 to 1998, he was a lecturer at Statistical Department at Universidade Federal de Pernambuco, Brazil. He joined the Center of Informatics at Universidade Federal de Pernambuco in 1999, where he is currently a full professor. He held visiting posts in several leading universities and research centers in Europe. With main research interests in symbolic data analysis, clustering analysis and statistical pattern recognition he has authored over 200 technical papers in international journals and conferences. He has served as Coordinator (2005–2009) of the post-graduate program of computer science of the CIn/UFPE. He has been involved in program committees of many Brazilian and international conferences. He has also served as a reviewer of many international journals and conferences. He was elected for the council (2009–2013) of the International Association for Statistical Computing (IASC).



Filipe M. de Melo currently is a master student in Computer Science at Universidade Federal de Pernambuco, Brazil.

Yves Lechevallier joined the INRIA in 1976 where he was engaged in the project of Clustering and Pattern Recognition. Since 1988 he has been teaching Clustering, Neural Network and Data Mining at the University of PARIS IX, CNAM and ENSAE. He specializes in Mathematical Statistics, Applied Statistics, Data Analysis and Classification. Current Research Interests: Clustering algorithms (Dynamic Clustering Method, Kohonen Maps, Divisive Clustering Method); Discrimination Problems and Decision Tree Methods; Build an efficient Neural Network by Classification Tree.