Partitioning Hard Clustering Algorithms Based On Multiple Dissimilarity Matrices

Francisco de A.T. de Carvalho [a,*], Yves Lechevallier [b] and 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*

Abstract

This paper introduces hard clustering algorithms that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. These matrices have been generated using different sets of variables and dissimilarity functions. These methods are designed to furnish a 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 fitting between the clusters and their representatives. These relevance weights change at each algorithm iteration and can either be the same for all clusters or different from one cluster to another. Experiments with data sets (synthetic and from UCI machine learning repository) described by real-valued variables as well as with time trajectory data sets show the usefulness of the proposed algorithms.

Key words: Partitioning Clustering Algorithms, Relational Data, Relevance Weight, Multiple Dissimilarity Matrices.

[*] Corresponding Author. tel.: +55-81-21268430 ; fax:+55-81-21268438
Email addresses: fatc@cin.ufpe.br (Francisco de A.T. de Carvalho), Yves.Lechevallier@inria.fr (Yves Lechevallier), fmm@cin.ufpe.br (Filipe M. de Melo).

[1] Acknowledgements. The authors are grateful to anonymous referees for their careful revision, valuable suggestion, and comments that have improved this paper. This research was partially supported by grants from CNPq, FACEPE (Brazilian Agencies) and from a conjoint research project FACEPE and INRIA (France).

1 Introduction

Clustering methods organize a set of items into clusters such that items within a given cluster have a high degree of similarity, whereas those of 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. The most popular clustering techniques are 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 [3–7] or divisive [8–12]. Agglomerative methods yield a sequence of nested partitions starting with trivial clustering in which each item is in a unique cluster and ending with 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 [13–17] and fuzzy clustering [18–22]. Hard clustering furnishes a partition in which each object of the data set is assigned to one and only one cluster. Fuzzy clustering generates a fuzzy partition that furnishes a degree of membership of each pattern in a given cluster. This gives the flexibility to express that objects belong to more than one cluster at the same time.

There are two common representations of the objects upon which clustering can be based : 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*. Alternatively, when each pair of objects is represented by a relationship, then it is called *relational data*. The most common case of relational data is when one has (a matrix of) dissimilarity data, say $R = [r_{kl}]$, where r_{kl} is the pairwise dissimilarity (often a distance) between objects k and l. Clustering of relational data is very useful when the objects cannot be described by a vector of feature values, when the distance measure does not have a closed form, etc [10], [23–26]. Recently, Frigui et al [27] proposed CARD, a relational 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 is mainly based on the well-known fuzzy clustering algorithms for relational data NERF

[26] and FANNY [10]. As remarked by [27], 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.

This paper extends the dynamic hard clustering algorithm for relational data [23], [24], into hard clustering algorithms that are able 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 [28] to obtain a final partition. These dissimilarity matrices could have been generated using different sets of variables and a fixed dissimilarity function (in this case, the final partition is given according to different views (i.e., different sets of variables) describing the objects), or using a fixed set of variables and different dissimilarity functions (in this case, the final partition is given according to different dissimilarity functions) or using different sets of variables and dissimilarity functions. As pointed out by [27], the influence of the different dissimilarity matrices can not be equally important in the definition of the clusters in the final partition. Thus, to obtain a meaningful partition from all dissimilarity matrices, the relational hard clustering algorithms given in this paper are designed to give a 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 fitting between the 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.

This paper is organized as follows. Section 2 first reviews a partitioning dynamic hard clustering algorithm based on a single dissimilarity matrix (section 2.1) and then introduces partitioning dynamic hard clustering algorithms based on multiple dissimilarity matrices with relevance weight for each dissimilarity matrix either estimated locally (section 2.2.1) or estimated globally (section 2.2.2). Section 3 gives empirical results to show the usefulness of these relational clustering algorithms. Finally, section 4 gives final remarks and comments.

2 Partitioning Hard Clustering Algorithms based on Multiple Dissimilarity Matrices

This section introduces partitioning dynamic hard clustering algorithm for relational data that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices.

2.1 *Dynamic Hard Clustering Algorithm based on a Single Dissimilarity Matrix*

There are several relational clustering algorithms based on a single dissimilarity matrix in the literature like SAHN (sequential agglomerative hierarchical non-overlapping) [1] and PAM (partitioning around medoids) [10] but the paper starts with a brief description of the partitioning dynamic hard clustering algorithm for relational data based on a single dissimilarity matrix [23,24] (denote here $SRDCA$) because the algorithms here are based on it.

Let $E = \{e_1, \ldots, e_n\}$ 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 $e_l \, (i, l = 1, \ldots, n)$. A particularity of this method is that it assumes that the prototype G_k of cluster C_k is a subset of fixed cardinality $1 \leq q << n$ of the set of objects E (even if, for a matter of simplicity, very often $q = 1$), i.e., $G_k \in E^{(q)} = \{A \subset E : |A| = q\}$. It looks for a partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the corresponding prototypes $G_1, \ldots, G_K$ representing the clusters in P such that it is (locally) optimized an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes.

The adequacy criterion measures the homogeneity of the partition P as the sum of the homogeneities in each cluster. It is defined as

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} D(e_i, G_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{e \in G_k} d(e_i, e) \qquad (1)$$

where $J_k = \sum_{e_i \in C_k} D(e_i, G_k)$ is the homogeneity in cluster $C_k \, (k = 1, \ldots, K)$ and

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

measures the matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$.

The $SRDCA$ relational clustering algorithm sets an initial partition and alternates two steps until convergence, when the criterion J reaches a stationary value representing a local minimum. This algorithm is summarized as follows.

Dynamic Hard Clustering Algorithm for Relational Data

(1) *Initialization.*
Fix the number K of clusters;

```
    Fix the cardinality $1 \leq q << n$ of the prototypes $G_k$ $(k = 1, \ldots, K)$;
    Set $t = 0$;
    Randomly select $K$ distinct prototypes $G_k^{(0)} \in E^{(q)}$ $(k = 1, \ldots, K)$;
    Assign each object $e_i$ to the closest prototype to obtain the partition $P^{(0)} = (C_1^{(0)}, \ldots, C_K^{(0)})$ with $C_k^{(0)} = \{e_i \in E : D(e_i, G_k^{(0)}) \leq D(e_i, G_h^{(0)}), (h = 1, \ldots, K)\}$.
(2) Step 1: computation of the best prototypes.
    Set $t = t + 1$;
    The partition $P^{(t-1)} = (C_1^{(t-1)}, \ldots, C_K^{(t-1)})$ is fixed.
    Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of cluster $C_k^{(t-1)}$ $( k = 1, \ldots, K)$ according to: $G^* = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} D(e_i, G) = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} \sum_{e \in G} d(e_i, e)$
(3) Step 2: definition of the best partition.
    The prototypes $G_k^{(t)} \in E^{(q)}$ $(k = 1, \ldots, K)$ are fixed.
    $test \leftarrow 0$
    $P^{(t)} \leftarrow P^{(t-1)}$
    for $i = 1$ to $n$ do
        find the cluster $C_m^{(t)}$ to which $e_i$ belongs
        find the winning cluster $C_k^{(t)}$ such that
            $k = argmin_{1 \leq h \leq K} D(e_i, G_h^{(t)}) = argmin_{1 \leq h \leq K} \sum_{e \in G_h} d(e_i, e)$
        if $k \neq m$
            $test \leftarrow 1$
            $C_k^{(t)} \leftarrow C_k^{(t)} \cup \{e_i\}$
            $C_m^{(t)} \leftarrow C_m^{(t)} \setminus \{e_i\}$

(4) Stopping criterion. If $test = 0$ then STOP; otherwise go to 2 (Step 1).
```

Let $E = \{e_1, \ldots, e_n\}$ be the set of n objects and let p dissimilarity matrices $\mathbf{D}_j = [d_j(e_i, e_l)]$ $(j = 1, \ldots, p)$, where $d_j(e_i, e_l)$ gives the dissimilarity between objects e_i and e_l $(i, l = 1, \ldots, n)$ on dissimilarity matrix $\mathbf{D}_j$.

The $SRDCA$ relational clustering algorithm can be changed into the "dynamic hard clustering algorithm based on multiple dissimilarity matrices" (denoted here $MRDCA$) to take into account simultaneously these p dissimilarity matrices $\mathbf{D}_j$. For that, the adequacy criterion of the $SRDCA$ relational clustering algorithm is modified into

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} D(e_i, G_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} D_j(e_i, G_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \sum_{e \in G_k} d_j(e_i, e) \tag{3}$$

in which

$$D(e_i, G_k) = \sum_{j=1}^{p} D_j(e_i, G_k) = \sum_{j=1}^{p} \sum_{e \in G_k} d_j(e_i, e) \qquad (4)$$

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 $\mathbf{D}_j$ $(j = 1, \ldots, p)$.

In this case, this algorithm is modified so as in the *Step 1* the prototype $G_k \in E^{(q)}$ of cluster C_k $(k = 1, \ldots, K)$ is computed according to $G^* = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} D_j(e_i, G) = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} \sum_{e \in G} d_j(e_i, e)$ whereas in the *Step 2* the winning cluster C_k is such that $k = argmin_{1 \leq h \leq K} \sum_{j=1}^{p} D_j(e_i, G_h) = argmin_{1 \leq h \leq K} \sum_{j=1}^{p} \sum_{e \in G_h} d_j(e_i, e)$.

This approach is equivalent to cluster 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 $d(e_i, e_l) = \sum_{j=1}^{p} d_j(e_i, e_l)$ $(i, l = 1, \ldots, n)$, that gives the same weight to the p partial dissimilarity matrices. However, as pointed out by [27], this approach may not be effective as the influence of each partial dissimilarity matrices may be not equally important to define the cluster to which similar objects belong.

2.2 Dynamic Hard Clustering Algorithms with Relevance Weight for each Dissimilarity Matrix

This section presents dynamic hard clustering algorithms based on multiple dissimilarity matrices. These algorithms extend the dynamic hard clustering algorithm for relational data [23,24]. The computation of the relevance weight of each dissimilarity matrix in these algorithms is inspired from the approach used to compute a relevance weight for each variable in each cluster in the dynamic clustering algorithm based on adaptive distances [29].

2.2.1 Dynamic Hard Clustering Algorithm with Relevance Weight for each Dissimilarity Matrix Estimated Locally

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

The dynamic hard clustering algorithm with relevance weight for each dissimilarity matrix estimated locally (denoted here $MRDCA - RWL$) looks for a partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the corresponding prototypes $G_1, \ldots, G_K$ representing the clusters in the partition P such that it (locally) optimizes an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes. The adequacy criterion is defined as

$$
\begin{aligned}
J &= \sum_{k=1}^{K} \sum_{e_i \in C_k} D_{\boldsymbol{\lambda}_k}(e_i, G_k) \qquad (5)\\
&= \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} \, D_j(e_i, G_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e)
\end{aligned}
$$

in which

$$
D_{\boldsymbol{\lambda}_k}(e_i, G_k) = \sum_{j=1}^{p} \lambda_{kj} \, D_j(e_i, G_k) = \sum_{j=1}^{p} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) \qquad (6)
$$

is the global matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$, parameterized by the relevance weight vector $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kp})$ of the dissimilarity matrices $\mathbf{D}_j$ into cluster C_k $(k = 1, \ldots, K)$, and $D_j(e_i, G_k)$ is the local dissimilarity between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$ on dissimilarity matrix $\mathbf{D}_j$ $(j = 1, \ldots, p)$.

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 $\boldsymbol{\lambda}_k$ $(k = 1, \ldots, K)$ are estimated locally, change at each iteration, i.e., they are not determined absolutely, and are different from one cluster to another.

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

Step 1: Computation of the Best Prototypes

In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the relevance weight vectors $\boldsymbol{\lambda}_k$ $(k = 1, \ldots, K)$ are fixed.

Proposition 2.1 *The prototype $G_k = G^* \in E^{(q)}$ of cluster C_k $(k = 1, \ldots, K)$, which minimizes the clustering criterion J, is computed according to:*

$$
\begin{aligned}
G^* &= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} \, D_j(e_i, G) \qquad (7)\\
&= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_{kj} \sum_{e \in G} d_j(e_i, e)
\end{aligned}
$$

Step 2: Computation of the Best Relevance Weight Vector

In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the prototypes $G_1, \ldots, G_K$ are fixed.

Proposition 2.2 *The vectors of relevance weights $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kp})\,(k = 1, \ldots, K)$, which minimizes the clustering criterion J under $\lambda_{kj} > 0$ and $\prod_{j=1}^{p} \lambda_{kj} = 1$, have their relevance weights $\lambda_{kj}\,(j = 1, \ldots, p)$ calculated according to the following expression:*

$$\lambda_{kj} = \frac{\left\{\prod_{h=1}^{p} \left[\sum_{e_i \in C_k} D_h(e_i, G_k)\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k} D_j(e_i, G_k)\right]} = \frac{\left\{\prod_{h=1}^{p} \left[\sum_{e_i \in C_k} \sum_{e \in G_k} d_h(e_i, e)\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k} \sum_{e \in G_k} d_j(e_i, e)\right]} \quad (8)$$

Proof. As the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the prototypes $G_1, \ldots, G_K$ are fixed, one can rewrite the criterion J as:

$$J(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K) = \sum_{k=1}^{K} J_k(\boldsymbol{\lambda}_k)$$

with

$$J_k(\boldsymbol{\lambda}_k) = J_k(\lambda_{k1}, \ldots, \lambda_{kp}) = \sum_{j=1}^{p} \lambda_{kj}\, J_{kj} \text{ where } J_{kj} = \sum_{i \in C_k} D_j(e_i, G_k)$$

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

$$\lambda_{kj} = \frac{\left(\prod_{h=1}^{p} J_{kh}\right)^{1/p}}{J_{kj}} = \frac{\left\{\prod_{h=1}^{p} \left(\sum_{e_i \in C_k} D_h(e_i, G_k)\right)\right\}^{\frac{1}{p}}}{\sum_{e_i \in C_k} D_j(e_i, G_k)}$$

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

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

Step 3: Definition of the Best Partition

In this step, the prototypes $G_1, \ldots, G_K$ and the relevance weight vectors $\boldsymbol{\lambda}_1, \ldots, \ \boldsymbol{\lambda}_k$ are fixed.

Proposition 2.3 *The clusters $C_k\,(k = 1, \ldots, K)$, which minimize the criterion J, are updated according to the following allocation rule:*

$$
\begin{aligned}
C_k = \{ & e_i \in E : D_{\boldsymbol{\lambda}_k}(e_i, G_k) = \sum_{j=1}^{p} \lambda_{kj}\, D_j(e_i, G_k) = \sum_{j=1}^{p} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) \\
& \leq D_{\boldsymbol{\lambda}_h}(e_i, G_h) = \sum_{j=1}^{p} \lambda_{hj}\, D_j(e_i, G_h) = \sum_{j=1}^{p} \lambda_{hj} \sum_{e \in G_h} d_j(e_i, e) \qquad (9) \\
& \quad \textit{and when } D_{\boldsymbol{\lambda}_k}(e_i, G_k) = D_{\boldsymbol{\lambda}_h}(e_i, G_h) \textit{ then } i \in C_k \textit{ if } k < h, \\
& \quad \forall h \neq k\,(h = 1, \ldots, K) \}
\end{aligned}
$$

Proof. The proof of Proposition 2.3 is straightforward.

Algorithm

The $MRDCA\!-\!RWL$ relational clustering algorithm is summarized as follows.

Dynamic Hard Clustering Algorithm with Relevance Weight for each Dissimilarity Matrix Estimated Locally

(1) *Initialization.*
Fix the number K of clusters;
Fix the cardinality $1 \leq q << n$ of the prototypes $G_k\,(k = 1, \ldots, K)$;
Set $t = 0$;
Set $\boldsymbol{\lambda}_k^{(0)} = (\lambda_{k1}^{(0)}, \ldots, \lambda_{kp}^{(0)}) = (1, \ldots, 1)\,(k = 1, \ldots, K)$;
Randomly select K distinct prototypes $G_k^{(0)} \in E^{(q)}\,(k = 1, \ldots, K)$;
Assign each object e_i to the closest prototype to obtain the partition $P^{(0)} = (C_1^{(0)}, \ldots, C_K^{(0)})$ with $C_k^{(0)} = \{e_i \in E : \sum_{j=1}^{p} \lambda_{kj}^{(0)}\, D_j(e_i, G_k^{(0)}) \leq \sum_{j=1}^{p} \lambda_{hj}^{(0)}\, D_j(e_i, G_h^{(0)}), (h = 1, \ldots, K)\}$.

(2) *Step 1: computation of the best prototypes.*
Set $t = t + 1$;
The partition $P^{(t-1)} = (C_1^{(t-1)}, \ldots, C_K^{(t-1)})$ and the relevance weight vectors $\boldsymbol{\lambda}_k^{(t-1)} = (\lambda_{k1}^{(t-1)}, \ldots, \lambda_{kp}^{(t-1)})$, $k = 1, \ldots, K$ are fixed.
Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of cluster $C_k^{(t-1)}$ $(\,k = 1, \ldots, K)$ according to: $G^* = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} \sum_{j=1}^{p} \lambda_{kj}^{(t-1)}\, D_j(e_i, G)$
$= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} \sum_{j=1}^{p} \lambda_{kj}^{(t-1)} \sum_{e \in G} d_j(e_i, e)$

(3) *Step 2: computation of the best relevance weight vector.*
The prototypes $G_k^{(t)} \in E^{(q)}\,(k = 1, \ldots, K)$ and the partition $P^{(t-1)} =$

$(C_1^{(t-1)}, \dots, C_K^{(t-1)})$ are fixed.
Compute the components $\lambda_{kj}^{(t)}$ $(j = 1, \dots, p)$ of the relevance weight vector $\boldsymbol{\lambda}_k^{(t)}$ $(k = 1 \dots, K)$ according to

$$\lambda_{kj}^{(t)} = \frac{\left\{\prod_{h=1}^{p}\left[\sum_{e_i \in C_k^{(t-1)}} D_h(e_i, G_k^{(t)})\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k^{(t-1)}} D_j(e_i, G_k^{(t)})\right]} = \frac{\left\{\prod_{h=1}^{p}\left[\sum_{e_i \in C_k^{(t-1)}} \sum_{e \in G_k^{(t)}} d_h(e_i, e)\right]\right\}^{\frac{1}{p}}}{\left[\sum_{e_i \in C_k^{(t-1)}} \sum_{e \in G_k^{(t)}} d_j(e_i, e)\right]}$$

(4) *Step 3: definition of the best partition.*
The prototypes $G_k^{(t)} \in E^{(q)}$ $(k = 1, \dots, K)$ and the relevance weight vectors $\boldsymbol{\lambda}_k^{(t)} = (\lambda_{k1}^{(t)}, \dots, \lambda_{kp}^{(t)})$, $k = 1, \dots, K$, are fixed.

```
$test \leftarrow 0$
$P^{(t)} \leftarrow P^{(t-1)}$
for $i = 1$ to $n$ do
    find the cluster $C_m^{(t)}$ to which $e_i$ belongs
    find the winning cluster $C_k^{(t)}$ such that
        $k = argmin_{1 \le h \le K} \sum_{j=1}^{p} \lambda_{hj}^{(t)} D_j(e_i, G_h^{(t)})$
        $= argmin_{1 \le h \le K} \sum_{j=1}^{p} \lambda_{hj}^{(t)} \sum_{e \in G_h^{(t)}} d_j(e_i, e)$
    if $k \neq m$
        $test \leftarrow 1$
        $C_k^{(t)} \leftarrow C_k^{(t)} \cup \{e_i\}$
        $C_m^{(t)} \leftarrow C_m^{(t)} \setminus \{e_i\}$
```

(5) *Stopping criterion.* If $test = 0$ then STOP; otherwise go to 2 (Step 1).

2.2.2 Dynamic Hard Clustering Algorithm with Relevance Weight of each Dissimilarity Matrix Estimated Globally

The clustering algorithm presented in section 2.2.1 presents numerical instabilities (division by zero) in the computation of the relevance weight of each dissimilarity matrix in each cluster when the algorithm produces single clusters or clusters with some objects that have dissimilarity zero between each other. To decreases significantly the probability of this kind of numerical instability, the authors present in this section an algorithm designed to give a partition and a prototype for each 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 clusters.

The dynamic hard clustering algorithm with relevance weight for each dissimilarity matrix estimated globally (denoted here $MRDCA - RWG$) looks for a partition $P = (C_1, \dots, C_K)$ of E into K clusters and the corresponding prototypes $G_1, \dots, G_K$ representing the clusters in the partition P such that it (locally) optimizes an adequacy criterion (objective function) measuring the fit between the clusters and their prototypes. The adequacy criterion is defined as

$$\begin{aligned} J &= \sum_{k=1}^{K} \sum_{e_i \in C_k} D_{\boldsymbol{\lambda}}(e_i, G_k) \qquad (10) \\ &= \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_j D_j(e_i, G_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_j \sum_{e \in G_k} d_j(e_i, e) \end{aligned}$$

in which

$$D_{\boldsymbol{\lambda}}(e_i, G_k) = \sum_{j=1}^{p} \lambda_j D_j(e_i, G_k) = \sum_{j=1}^{p} \lambda_j \sum_{e \in G_k} d_j(e_i, e) \qquad (11)$$

is the global matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$ parameterized by the relevance weight vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ of the dissimilarity matrices $\mathbf{D}_j$ and $D_j(e_i, G_k)$ is the local matching between an example $e_i \in C_k$ and the cluster prototype $G_k \in E^{(q)}$ on dissimilarity matrix $\mathbf{D}_j$ $(j = 1, \ldots, p)$.

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 $\boldsymbol{\lambda}$ is estimated globally, changes at each iteration but is the same for all clusters.

From an initial partition, this clustering algorithm alternates three steps and stops when the criterion J reaches a stationary value representing a local minimum.

Step 1: Computation of the Best Prototypes

In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the relevance weight vector $\boldsymbol{\lambda}$ are fixed.

Proposition 2.4 *The prototype $G_k = G^* \in E^{(q)}$ of cluster C_k $(k = 1, \ldots, K)$, which minimizes the clustering criterion J, is computed according to:*

$$\begin{aligned} G^* &= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_j \, D_j(e_i, G) \qquad (12) \\ &= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_j \sum_{e \in G} d_j(e_i, e) \end{aligned}$$

Step 2: Computation of the Best Relevance Weight Vector

In this step, the partition $P = (C_1, \ldots, C_K)$ of E into K clusters and the prototypes $G_1, \ldots, G_K$ are fixed.

Proposition 2.5 *The vector of relevance weights* $\boldsymbol{\lambda}= (\lambda_1, \ldots, \lambda_p)$*, which minimizes the clustering criterion* J *under* $\lambda_j > 0$ *and* $\prod_{j=1}^{p} \lambda_j = 1$*, has its relevance weights* λ_j $(j = 1, \ldots, p)$ *calculated according to the following expression:*

$$\begin{aligned}\lambda_j &= \frac{\left\{\prod_{h=1}^{p}\left(\sum_{k=1}^{K}\left[\sum_{e_i \in C_k} D_h(e_i, G_k)\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K}\left[\sum_{e_i \in C_k} D_j(e_i, G_k)\right]} \\ &= \frac{\left\{\prod_{h=1}^{p}\left(\sum_{k=1}^{K}\left[\sum_{e_i \in C_k}\sum_{e \in G_k} d_h(e_i, e)\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K}\left[\sum_{e_i \in C_k}\sum_{e \in G_k} d_j(e_i, e)\right]}\end{aligned} \tag{13}$$

Proof. The Proof proceeds in a similar way as presented in Proposition 2.2.

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

Step 3: Definition of the Best Partition

In this step, the prototypes $G_1, \ldots, G_K$ and the relevance weight vector $\boldsymbol{\lambda}$ are fixed.

Proposition 2.6 *The clusters* C_k $(k = 1, \ldots, K)$*, which minimize the criterion* J*, are updated according to the following allocation rule:*

$$\begin{aligned} C_k = \{e_i \in E : D_{\boldsymbol{\lambda}}(e_i, G_k) &= \sum_{j=1}^{p} \lambda_j \, D_j(e_i, G_k) = \sum_{j=1}^{p} \lambda_j \sum_{e \in G_k} d_j(e_i, e) \\ \leq D_{\boldsymbol{\lambda}}(e_i, G_h) &= \sum_{j=1}^{p} \lambda_j \, D_j(e_i, G_h) = \sum_{j=1}^{p} \lambda_j \sum_{e \in G_h} d_j(e_i, e) \\ &\textit{and when } D_{\boldsymbol{\lambda}}(e_i, G_k) = D_{\boldsymbol{\lambda}}(e_i, G_h) \textit{ then } i \in C_k \textit{ if } k < h, \\ &\forall h \neq k \, (h = 1, \ldots, K)\} \end{aligned} \tag{14}$$

Proof. The proof of Proposition 2.6 is straightforward.

Algorithm

The $MRDCA - RWG$ relational clustering algorithm is summarized as follows.

Dynamic Hard Clustering Algorithm with Relevance Weight for each Dissimilarity Matrix Estimated Globally

(1) *Initialization.*
Fix the number K of clusters;
Fix the cardinality $1 \leq q << n$ of the prototypes G_k $(k = 1, \ldots, K)$;
Set $t = 0$;
Set $\boldsymbol{\lambda}^{(0)} = (\lambda_1^{(0)}, \ldots, \lambda_p^{(0)}) = (1, \ldots, 1)$;
Randomly select K distinct prototypes $G_k^{(0)} \in E^{(q)}$ $(k = 1, \ldots, K)$;
Assign each object e_i to the closest prototype to obtain the partition $P^{(0)} = (C_1^{(0)}, \ldots, C_K^{(0)})$ with $C_k^{(0)} = \{e_i \in E : \sum_{j=1}^{p} \lambda_j^{(0)} D_j(e_i, G_k^{(0)}) \leq \sum_{j=1}^{p} \lambda_j D_j(e_i, G_h^{(0)}), (h = 1, \ldots, K)\}$.

(2) *Step 1: computation of the best prototypes.*
Set $t = t + 1$;
The partition $P^{(t-1)} = (C_1^{(t-1)}, \ldots, C_K^{(t-1)})$ and the relevance weight vector $\boldsymbol{\lambda}^{(t-1)} = (\lambda_1^{(t-1)}, \ldots, \lambda_p^{(t-1)})$ are fixed.
Compute the prototype $G_k^{(t)} = G^* \in E^{(q)}$ of cluster $C_k^{(t-1)}$ $(k = 1, \ldots, K)$ according to: $G^* = argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} \sum_{j=1}^{p} \lambda_j^{(t-1)} D_j(e_i, G)$
$= argmin_{G \in E^{(q)}} \sum_{e_i \in C_k^{(t-1)}} \sum_{j=1}^{p} \lambda_j^{(t-1)} \sum_{e \in G} d_j(e_i, e)$

(3) *Step 2: computation of the best relevance weight vector.*
The prototypes $G_k^{(t)} \in E^{(q)}$ $(k = 1, \ldots, K)$ and the partition $P^{(t-1)} = (C_1^{(t-1)}, \ldots, C_K^{(t-1)})$ are fixed.
Compute the relevance weight vector according to

$$\lambda_j^{(t)} = \frac{\left\{\prod_{h=1}^{p}\left(\sum_{k=1}^{K}\left[\sum_{e_i \in C_k^{(t-1)}} D_h(e_i, G_k^{(t)})\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K}\left[\sum_{e_i \in C_k^{(t-1)}} D_j(e_i, G_k^{(t)})\right]} = \frac{\left\{\prod_{h=1}^{p}\left(\sum_{k=1}^{K}\left[\sum_{e_i \in C_k^{(t-1)}} \sum_{e \in G_k^{(t)}} d_h(e_i, e)\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K}\left[\sum_{e_i \in C_k^{(t-1)}} \sum_{e \in G_k^{(t)}} d_j(e_i, e)\right]}$$

(4) *Step 3: definition of the best partition.*
The prototypes $G_k^{(t)} \in E^{(q)}$ $(k = 1, \ldots, K)$ and the relevance weight vector $\boldsymbol{\lambda}^{(t)} = (\lambda_1^{(t)}, \ldots, \lambda_p^{(t)})$ are fixed.

```
$test \leftarrow 0$
$P^{(t)} \leftarrow P^{(t-1)}$
for $i = 1$ to $n$ do
    find the cluster $C_m^{(t)}$ to which $e_i$ belongs
    find the winning cluster $C_k^{(t)}$ such that
        $k = argmin_{1 \leq h \leq K} \sum_{j=1}^{p} \lambda_j^{(t)} D_j(e_i, G_h^{(t)})$
        $= argmin_{1 \leq h \leq K} \sum_{j=1}^{p} \lambda_j^{(t)} \sum_{e \in G_h^{(t)}} d_j(e_i, e)$
    if $k \neq m$
        $test \leftarrow 1$
        $C_k^{(t)} \leftarrow C_k^{(t)} \cup \{e_i\}$
        $C_m^{(t)} \leftarrow C_m^{(t)} \setminus \{e_i\}$
```

(5) *Stopping criterion.* If *test* $= 0$ then STOP, otherwise go to 2 (Step 1).

This section illustrates the convergence properties of the presented algorithms by giving the proof of the convergence of the $MRDCA-RWL$ clustering algorithm introduced in section 2.2.1. Then, the complexity of both $MRDCA-RWL$ and $MRDCA-RWG$ clustering algorithms are given.

According to the general schema of the dynamic clustering algorithm [30], this clustering method looks for the partition $P^* = \{C_1^*, \dots, C_K^*\}$ of E into K clusters, the corresponding K prototypes $\mathbf{G}^* = (G_1^*, \dots, G_K^*)$ representing the clusters in P^* and K squared adaptive Euclidean distances parameterized by K vectors of weights $\mathbf{D}^* = (\boldsymbol{\lambda}_1^*, \dots, \boldsymbol{\lambda}_K^*)$ such that

$$W(\mathbf{G}^*, \mathbf{D}^*, P^*) = min\Big\{W(\mathbf{G}, \mathbf{D}, P) : \mathbf{G} \in \mathbb{L}^K, \mathbf{D} \in \boldsymbol{\Lambda}^K, P \in \mathbb{P}_K\Big\} \qquad (15)$$

where

- $\mathbb{P}_K$ is the set of all the possible partitions of E in K classes such that $C_k \in \mathbf{P}(E)$ (the set of subsets of E) and $P \in \mathbb{P}_K$;
- $\mathbb{L}$ is the representation space of prototypes such that $G_k \in \mathbb{L}\,(k = 1, \dots, K)$ and $\mathbf{G} \in \mathbb{L}^K = \mathbb{L} \times \ldots \times \mathbb{L}$. In this paper, $\mathbb{L} = E^{(q)} = \{A \subset E : |A| = q\}$.
- $\boldsymbol{\Lambda}$ is the space of vectors of weights that parameterize the adaptive Euclidean distances such that $\boldsymbol{\lambda}_k \in \boldsymbol{\Lambda}(k = 1, \dots, K)$. Here, $\boldsymbol{\Lambda}= \{\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p : \lambda_j > 0$ and $\prod_{j=1}^p \lambda_j = 1\}$ and $\mathbf{D} \in\boldsymbol{\Lambda}^K = \boldsymbol{\Lambda}\times \ldots \times \boldsymbol{\Lambda}$.

According to [30], the properties of convergence of this kind of algorithm can be studied from two series: $v_t = (\mathbf{G}^t, \mathbf{D}^t, P^t) \in \mathbb{L}^K\times\ \boldsymbol{\Lambda}^K \times \mathbb{P}_K$ and $u_t = J(v_t) = J(\mathbf{G}^t, \mathbf{D}^t, P^t),\ t = 0, 1, \ldots$. From an initial term $v_0 = (\mathbf{G}^0, \mathbf{D}^0, P^0)$, 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 2.7 *The series $u_t = J(v_t)$ decreases at each iteration and converges.*

Proof.

Following [30], first the authors of this study show that the inequalities (I), (II) and (III)

$$\underbrace{J(\mathbf{G}^t, \mathbf{D}^t, P^t)}_{u_t} \overbrace{\geq}^{(I)} J(\mathbf{G}^{t+1}, \mathbf{D}^t, P^t) \overbrace{\geq}^{(II)} J(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^t) \overbrace{\geq}^{(III)} \underbrace{J(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^{t+1})}_{u_{t+1}}$$

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

The inequality (I) holds because

$$J(\mathbf{G}^t, \mathbf{D}^t, P^t) = \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k^{(t)}} D_{\boldsymbol{\lambda}_k^{(t)}}(e_i, G_k^{(t)}),$$

$$J(\mathbf{G}^{t+1}, \mathbf{D}^t, P^t) = \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k^{(t)}} D_{\boldsymbol{\lambda}_k^{(t)}}(e_i, G_k^{(t+1)}),$$

and according to proposition (2.1),

$$\mathbf{G}^{(t+1)} = (G_1^{(t+1)}, \ldots, G_K^{(t+1)}) = \underbrace{argmin}_{\mathbf{G}=(G_1,\ldots,G_K)\in\mathbb{L}^K} \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k^{(t)}} D_{\boldsymbol{\lambda}_k^{(t)}}(e_i, G_k).$$

Moreover, inequality (II) also holds because

$$J(\mathbf{G}^{t+1}, \mathbf{D}^{(t+1)}, P^t) = \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k^{(t)}} D_{\boldsymbol{\lambda}_k^{(t+1)}}(e_i, G_k^{(t+1)}),$$

and according to proposition (2.2),

$$\mathbf{D}^{t+1} = (\boldsymbol{\lambda}_1^{(t+1)}, \ldots, \boldsymbol{\lambda}_K^{(t+1)}) = \underbrace{argmin}_{\mathbf{D}=(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_K)\in\boldsymbol{\Lambda}^K} \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k^{(t)}} D_{\boldsymbol{\lambda}_k}(e_i, G_k^{(t+1)})$$

The inequality (III) also holds because

$$J(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^{t+1}) = \textstyle\sum_{e_i \in C_k^{(t+1)}} D_{\boldsymbol{\lambda}_k^{(t+1)}}(e_i, G_k^{(t+1)}),$$

and according to proposition (2.3),

$$P^{t+1} = \{C_1^{t+1}, \ldots, C_K^{t+1}\} = \underbrace{argmin}_{P=\{C_1,\ldots,C_K\}\in\mathbb{P}_K} \textstyle\sum_{k=1}^{K} \sum_{e_i \in C_k} D_{\boldsymbol{\lambda}_k^{(t+1)}}(e_i, G_k^{(t+1)}).$$

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

Proposition 2.8 *The series $v_t = (\boldsymbol{G}^t, \boldsymbol{D}^t, P^t)$ converges.*

Proof. Assume that the stationarity of the series u_t is achieved in the iteration $t = T$. Then, it is seen that $u_T = u_{T+1}$ and then $J(v_T) = J(v_{T+1})$.

From $J(v_T) = J(v_{T+1})$, one has $J(\mathbf{G}^t, \mathbf{D}^t, P^t) = J(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^{T+1})$ and this equality, according to proposition 2.7, can be rewritten as equalities (I), (II) and (III):

$$J(\mathbf{G}^t, \mathbf{D}^t, P^t) \overbrace{=}^{I} J(\mathbf{G}^{T+1}, \mathbf{D}^T, P^T) \overbrace{=}^{II} J(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^T) \overbrace{=}^{III}$$
$$J(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^{T+1})$$

From the first equality (I), one can understand that $\mathbf{G}^T = \mathbf{G}^{T+1}$ because $\mathbf{G}$ is unique in minimizing J when the partition P^T and the vector of vectors of weights $\mathbf{D}^T$ are fixed. From the second equality (II), $\mathbf{D}^T = \mathbf{D}^{T+1}$ because $\mathbf{D}$ is unique in minimizing J when the partition P^T and the vector of prototypes $\mathbf{G}^{T+1}$ are fixed. Moreover, from the third equality (III), $P^T = P^{T+1}$ because P is unique in minimizing J (because if the minimum is not unique, e_i is assigned to the cluster having the smallest index) when the vector of prototypes $\mathbf{G}^{T+1}$ and the vector vectors of weights $\mathbf{D}^T$ are fixed.

Finally, one can 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.

The time complexity of $MRDCA - RWL$ can be analyzed considering the complexity of each single step. Let n be the number of objects, $K << n$ be the number of clusters, $q << n$ be the cardinality of each prototype 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 K distinct prototypes (i.e., the selection of $K \times q$ distinct objects) can be done using random functions and a red-black tree to check for repetitions. The time complexity is then $O(K \times q \times log(K \times q))$. The assignment of each object to the closest prototype corresponds to the step 3. The complexity is $O(n \times K \times q \times p)$. Thus, the initialization costs $O(n \times K \times q \times p)$.
- *Step1: computation of the best prototypes.* For each cluster using each table of dissimilarity the authors test each individual as a candidate prototype. This needs the computation of the distance between an individual $i\,(i = 1, ...n)$ and all elements of each cluster using all p dissimilarity matrices and it costs $O(n^2 * p)$. The selection of the prototype of cardinality q for each cluster needs to sort the vector of individual for each cluster (it costs $O(K \times n \times log\, n)$) and to select the best q individuals as the prototype (it costs $O(K \times q)$). Thus, the step 1 costs $O(n^2 * p)$.
- *Step 2: computation of the best relevance weight vectors.* According to equation (8), 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 $O(n \times q \times p + K \times p)$.
- *Step 3: definition of the best partition.* This step needs the computation of the dissimilarity between an individual $i\,(i = 1, \ldots, n)$ and the prototype of cardinality q of each cluster using the p dissimilarity matrices and it costs $O(n \times q \times K \times p)$.

So, globally these steps cost $O(n^2 * p)$. Thus, if the clustering process needs t iterations to converge, the total time complexity of this algorithm is $O(n^2 \times p \times t)$. Following a similar reasoning, one can conclude that the total time complexity of $MRDCA - RWG$ is also $O(n^2 \times p \times t)$.

3 Empirical results

To evaluate the performance of these partitioning relational hard clustering algorithms in comparison with $NERF$ and $SRDCA$ (relational clustering algorithms that perform on a single dissimilarity matrix) as well as $MRDCA$ (relational hard clustering algorithm that performs on multiple dissimilarity matrices) and $CARD-R$ (relational fuzzy clustering algorithm that performs on multiple dissimilarity matrices and learns a relevance weight for each dissimilarity matrix in each cluster), applications with synthetic and real data sets (available at the UCI Repository http://www.ics.uci.edu/mlearn/ ML-Repository.html) described by real-valued variables as well as time trajectories data sets (available at http://www.math.univ-toulouse.fr/staph/npfda/ npfda-datasets.html) are considered.

The relational hard clustering algorithms $SRDCA$, $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ will be applied to these data sets to obtain a partition $Q = (Q_1, ..., Q_K)$. $NERF$ and $CARD - R$ will be also applied to these data sets to obtain first a fuzzy partition into K fuzzy clusters. Then, a hard partition $Q = (Q_1, \ldots, Q_K)$ is obtained from this fuzzy partition by defining the hard cluster $Q_k(k = 1, \ldots, K)$ as: $Q_k = \{e_i : u_{ik} \geq u_{im} \forall m \in \{1, \ldots, K\}\}$. The quantity u_{ik} is the membership degree of object $e_i \, (i = 1, \ldots, n)$ in fuzzy cluster $k \, (k = 1, \ldots, K)$.

To compare the clustering results furnished by the clustering methods, an external index – the corrected Rand index (CR) [31] – as well as the $F-measure$ [32] and the overall error rate of classification ($OERC$) [33] will be considered.

Let $P = \{P_1, \ldots, P_i, \ldots, P_m\}$ be the *a priori* partition into m classes and $Q = \{Q_1, \ldots, Q_j, \ldots, Q_K\}$ be the hard partition into K clusters given by a clustering algorithm. Let the confusion matrix be as below:

The corrected Rand index is:

$$CR = \frac{\sum_{i=1}^{m} \sum_{j=1}^{K} \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^{m} \binom{n_{i\bullet}}{2} \sum_{j=1}^{K} \binom{n_{\bullet j}}{2}}{\frac{1}{2}[\sum_{i=1}^{m} \binom{n_{i\bullet}}{2} + \sum_{j=1}^{K} \binom{n_{\bullet j}}{2}] - \binom{n}{2}^{-1} \sum_{i=1}^{m} \binom{n_{i\bullet}}{2} \sum_{j=1}^{K} \binom{n_{\bullet j}}{2}} \tag{16}$$

Table 1
Confusion matrix

Classes	Clusters					
	Q_1	$\dots$	Q_j	$\dots$	Q_K	$\sum$
P_1	n_{11}	$\dots$	n_{1j}	$\dots$	n_{1K}	$n_{1\bullet} = \sum_{j=1}^{K} n_{1j}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
P_i	n_{i1}	$\dots$	n_{ij}	$\dots$	n_{iK}	$n_{i\bullet} = \sum_{j=1}^{K} n_{ij}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	
P_m	n_{m1}	$\dots$	n_{mj}	$\dots$	n_{mK}	$n_{m\bullet} = \sum_{j=1}^{K} n_{mj}$
$\sum$	$n_{\bullet 1} = \sum_{i=1}^{m} n_{i1}$	$\dots$	$n_{\bullet j} = \sum_{i=1}^{m} n_{ij}$	$\dots$	$n_{\bullet K} = \sum_{i=1}^{m} n_{iK}$	$n = \sum_{i=1}^{m} \sum_{j=1}^{K} n_{ij}$

where $\binom{n}{2} = \frac{n(n-1)}{2}$ and n_{ij} represents the number of objects that are in class P_i and cluster Q_j; $n_{i\bullet}$ indicates the number of objects in class P_i; $n_{\bullet j}$ indicates the number of objects in cluster Q_j; and n is the total number of objects in the data set.

CR index 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. Finally, *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 [34].

The traditional $F - measure$ between class $P_i \, (i = 1, \dots, m)$ and cluster $Q_j \, (j = 1, \dots, K)$ is the harmonic mean of precision and recall:

$$F - measure(P_i, Q_j) = 2 \, \frac{Precision(P_i, Q_j) \, Recall(P_i, Q_j)}{Precision(P_i, Q_j) + Recall(P_i, Q_j)} \tag{17}$$

The *Precision* between class $P_i \, (i = 1, \dots, m)$ and cluster $Q_j \, (j = 1, \dots, K)$ is defined as the ratio between the number of objects that are in class P_i and cluster Q_j and the number of objects in cluster Q_j:

$$Precision(P_i, Q_j) = \frac{n_{ij}}{n_{\bullet j}} = \frac{n_{ij}}{\sum_{i=1}^{m} n_{ij}} \tag{18}$$

The *Recall* between class $P_i \, (i = 1, \dots, m)$ and cluster $Q_j \, (j = 1, \dots, K)$ is defined as the ratio between the number of objects that are in class P_i and cluster Q_j and the number of objects in class P_i:

$$Recall(P_i, Q_j) = \frac{n_{ij}}{n_{i\bullet}} = \frac{n_{ij}}{\sum_{j=1}^{K} n_{ij}} \tag{19}$$

The $F - measure$ between the *a priori* partition $P = \{P_1, \ldots, P_i, \ldots, P_m\}$ and the hard partition $Q = \{Q_1, \ldots, Q_j, \ldots, Q_K\}$ given by a cluster algorithm is defined as:

$$F - measure(P, Q) = \frac{1}{n} \sum_{i=1}^{m} n_{i\bullet} \ \max{}_{1 \leq j \leq K} \ F - measure(P_i, Q_j) \qquad (20)$$

The $F - measure$ index takes its values from the interval [0,1], in which the value 1 indicates perfect agreement between partitions.

In classification problems, each cluster Q_j is assigned to an *a priori* class P_i and this assignment must be interpreted as if the true *a priori* class is P_i. Once this decision is taken, for a given object of the cluster Q_j the decision is correct if the *a priori* class of this object is P_i and is an error if the *a priori* class is not P_i. To have a minimum error rate of classification ERC, one needs to seek a decision rule that minimizes the probability of error.

Let $p(P_i/Q_j)$ be the posterior probability that an object belongs to the class P_i when it is assigned to the cluster Q_j. Let $p(Q_j)$ the probability that the object belongs to the cluster Q_j. The function p is known as the likelihood function.

The maximum a posteriori probability (MAP) estimate is the mode of the posteriori probability $p(P_i/Q_j)$ and the index of the *a priori* class associated to this mode is given by:

$$MAP(Q_j) = \arg \max_{1 \leq i \leq m} p(P_i/Q_j) \qquad (21)$$

The Bayes decision rule to minimize the average probability of error is to select the *a priori* class that maximizes the posterior probability. The error rate of classification $ERC(Q_j)$ of the cluster Q_j is equal to $1 - p(P_{MAP(Q_j)}/Q_j)$ and the overall error rate of classification $OERC$ is equal to :

$$OERC = \sum_{j=1}^{K} p(Q_j)(1 - p(P_{MAP(Q_j)}/Q_j)) \qquad (22)$$

For a sample,

$$p(P_{MAP(Q_j)}/Q_j) = \max_{1 \leq i \leq m} \frac{n_{ij}}{n_{\bullet j}}. \qquad (23)$$

The $OERC$ index aims to measure the ability of a clustering algorithm to find out the *a priori* classes present in a data set and it is computed by :

$$OERC = \sum_{j=1}^{K} \frac{n_{\bullet j}}{n} (1 - \max_{1 \leq i \leq m} n_{ij}/n_{\bullet j}) = 1 - \frac{\sum_{j=1}^{K} \max_{1 \leq i \leq m} n_{ij}}{n} \qquad (24)$$

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 (see Table 2), but different covariance matrices (see Table 3): 1) the variance are different between the variables and from one class to another (synthetic data set 1); 2) the variance are different between the variables but they are almost the same from one class to another (synthetic data set 2); 3) the variance are almost the same between the variables and different from one class to another (synthetic data set 3). 4) the variance are almost the same between the variables and from one class to another (synthetic data set 3).

Table 2
Configurations of quantitative data sets: mean vectors of the bivariate normal distributions of the classes.

$\boldsymbol{\mu}$	Class 1	Class 2	Class 3	Class 4
μ_1	45	70	45	42
μ_2	30	38	35	20

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 others 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 were computed according to the Euclidean (L_2) distance.

For these data sets, $NERF$ and $SRDCA$ are performed on the dissimilarity

Table 3
Configurations of quantitative data sets: covariance matrices of the bivariate normal distributions of the classes.

$\mathbf{\Sigma}$	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
$\mathbf{\Sigma}$	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_1	16	10	2	6	8	8	8	8
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

matrix that has the cells that are the dissimilarities between pairs of objects computed taking into account simultaneously the two real-valued attributes. $CARD-R$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ are 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.

All dissimilarity matrices were normalized according to their overall dispersion [37] to have the same dynamic range. This means that each dissimilarity $d(e_k, e_{k'})$ in a given dissimilarity matrix has been normalized as $\frac{d(e_k,e_{k'})}{T}$ where $T = \sum_{k=1}^{n} d(e_k, g)$ is the overall dispersion and $g = e_l \in E = \{e_1, \ldots, e_n\}$is the overall prototype, which is computed according to $l = argmin_{1 \leq h \leq n} \sum_{k=1}^{n}$ $d(e_k, e_h)$.

The relational fuzzy clustering algorithms $NERF$ and $CARD-R$ were applied to the dissimilarity matrices obtained from this data set to obtain a four-cluster fuzzy partition. The hard clustering algorithms $SRDCA$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ were applied to the dissimilarity matrices obtained from this data set to obtain a four-cluster hard partition. The hard cluster partitions (obtained from the fuzzy partitions given by $NERF$ or $CARD-R$ or obtained directly from $SRDCA$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$) were compared with the known a priori class partition. For the synthetic data sets, the CR, $F-measure$ and $OERC$ indexes were estimated in the framework of a Monte Carlo simulation with 100 replications. The average and the standard deviation of these indexes 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 CR, $F-measure$ and $OERC$ indexes were calculated for the best result.

Table 4 shows the performance of the $NERF$ and the $CARD-R$ algorithms as well as the performance of $SRDCA$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ algorithms (with prototypes of cardinality $|G_k|=1$, $k=1,\ldots,4$) on the synthetic data sets according to the average and the standard deviation of the CR, $F-measure$ and $OERC$ indexes. Table 5 shows the 95% confidence interval for the average of the CR, $F-measure$ and $OERC$ indexes.

The performance of the algorithms $MRDCA-RWL$ and $CARD-R$ 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. $MRDCA-RWL$, $CARD-R$ and $MRDCA-RWG$ algorithms were also superior when the variance was different between the variables but almost the same from one class to another (synthetic data set 2) especially in comparison with the algorithms that perform on a single dissimilarity matrix ($NERF$ and $SRDCA$). Moreover, except $CARD-R$, which presented its worst performance, all the other algorithms had a similar performance when the variance was almost the same between the variables and different from one class to another (synthetic data set 3). Finally, the algorithms $NERF$ and $SRDCA$ were superior in comparison with all the other algorithms when the variance was almost the same between the variables and from one class to another (synthetic data set 4). In conclusion, in comparison with the algorithms that perform on a single dissimilarity matrix, $MRDCA-RWL$ was clearly superior in the synthetic data sets where the variance was different between the variables whereas $MRDCA-RWG$ was clearly superior only in the synthetic data sets where the variance was different between the variables but almost the same from one class to another.

3.2 UCI machine learning repository data sets

This paper considers abalone, image, iris plants, thyroid gland, and wine data sets. These data sets are found in http://www.ics.uci.edu/mlearn/ MLRepository.html.

All these data sets are described by a data matrix of “objects × real-valued attributes”. Several dissimilarity matrices are 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 which

Table 4
Performance of the algorithms on the synthetic data sets: average and standard deviation (in parenthesis) of the CR, $F-measure$, and $OERC$ indexes

Algorithms	Synthetic data set 1		
	CR	$F-measure$	$OERC$
$NERF$	0.1334 (0.0206)	0.4942 (0.0187)	42.98% (1.88%)
$SRDCA$	0.1118 (0.0221)	0.4861 (0.0207)	44.57% (2,19%)
$MRDCA$	0.1008 (0.0236)	0.4645 (0.0257)	46.14% (2.50%)
$MRDCA-RWG$	0.1137 (0.0250)	0.4790 (0.0239)	44.98% (2.44%)
$MRDCA-RWL$	0.5327 (0.0281)	0.7080 (0.0262)	23.76% (2.54%)
$CARD-R$	0.4810 (0.0296)	0.6947 (0.0217)	25.69% (2.35%)

Algorithms	Synthetic data set 2		
	CR	$F-measure$	$OERC$
$NERF$	0.1416 (0.0173)	0.4730 (0.0186)	46.26% (1.72%)
$SRDCA$	0.1415 (0.0178)	0.4728 (0.0212)	46.13% (1.79%)
$MRDCA$	0.2066 (0.0264)	0.5486 (0.0291)	39.82% (2.87%)
$MRDCA-RWG$	0.2384 (0.0327)	0.5763 (0.0309)	37.54% (2.90%)
$MRDCA-RWL$	0.2343 (0.0414)	0.5672 (0.0423)	38.26% (3.96%)
$CARD-R$	0.2571 (0.0214)	0.5828 (0.0212)	36.88% (1.89%)

Algorithms	Synthetic data set 3		
	CR	$F-measure$	$OERC$
$NERF$	0.2381 (0.0279)	0.5294 (0.0244)	41.75% (2.41%)
$SRDCA$	0.2172 (0.0446)	0.5189 (0.0383)	43.20% (3.36%)
$MRDCA$	0.2353 (0.0439)	0.5448 (0.0284)	41.93% (3.20%)
$MRDCA-RWG$	0.2133 (0.0368)	0.5123 (0.0312)	43.07% (3.15%)
$MRDCA-RWL$	0.2208 (0.0296)	0,5180 (0.0301)	43.44% (2.66%)
$CARD-R$	0.1285 (0.0130)	0.4395 (0.0200)	51.73% (1.76%)

Algorithms	Synthetic data set 4		
	CR	$F-measure$	$OERC$
$NERF$	0.2942 (0.0285)	0.6013 (0.0267)	34.79% (2.48%)
$SRDCA$	0.3014 (0.0307)	0.6034 (0.0214)	34.54% (2.49%)
$MRDCA$	0.2741 (0.0312)	0,5910 (0.0250)	35.83% (2.79%)
$MRDCA-RWG$	0.2888 (0.0276)	0.5885 (0.0244)	35.55% (2.08%)
$MRDCA-RWL$	0.2873 (0.0313)	0.5826 (0.0316)	35.96% (2.91%)
$CARD-R$	0.1625 (0.0190)	0.5026 (0.0207)	43.56% (2.24%)

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 were computed according to the Euclidean (L_2) distance.

For these data sets, $NERF$ and $SRDCA$ 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. $CARD-R$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ were performed simultaneously on all dissimilarity matrices which have the cells that

Table 5
Performance of the algorithms on the synthetic data sets: 95% confidence interval for the average of the CR, $F-measure$, and $OERC$ indexes

Algorithms	Synthetic data set 1		
	CR	$F-measure$	$OERC$
$NERF$	0.1293—0.1374	0.4905—0.4978	42.61%—43.35%
$SRDCA$	0.1074—0.1161	0.4820—0.4901	44.14%—45.00%
$MRDCA$	0.0961—0.1054	0.4594—0.4695	45.65%—46.63%
$MRDCA-RWG$	0.1088—0.1186	0.4743—0.4836	44,50%—45.45%
$MRDCA-RWL$	0.5271—0.5382	0.7028—0.7131	23.26%—24.25%
$CARD-R$	0.4751—0.4868	0.6904—0.6989	25.23%—26.15%

Algorithms	Synthetic data set 2		
	CR	$F-measure$	$OERC$
$NERF$	0.1382—0.1449	0.4693—0.4766	45.92%—46.60%
$SRDCA$	0.1380—0.1449	0.4686—0.4769	45.78%—46.48%
$MRDCA$	0.2014—0.2117	0.5428—0.5543	39.26%—40.39%
$MRDCA-RWG$	0.2319—0.2448	0.5702—0.5823	36.97%—38.11%
$MRDCA-RWL$	0.2261—0.2424	0.5589—0.5754	37.48%—39.04%
$CARD-R$	0.2529—0.2612	0.5786—0.5869	36.51%—37.25%

Algorithms	Synthetic data set 3		
	CR	$F-measure$	$OERC$
$NERF$	0.2326—0.2435	0.5246—0.5341	41.27%—42.22%
$SRDCA$	0.2084—0.2259	0.5113—0.5264	42.54%—43.86%
$MRDCA$	0.2266—0.2439	0.5392—0.5503	41.30%—42.56%
$MRDCA-RWG$	0.2060—0.2205	0.5061—0.5184	42.45%—43.69%
$MRDCA-RWL$	0.2149—0.2266	0.5121—0.5238	42.92%—43.96%
$CARD-R$	0.1259—0.1310	0.4355—0.4434	51.38%—52.07%

Algorithms	Synthetic data set 4		
	CR	$F-measure$	$OERC$
$NERF$	0.2886—0.2997	0.5960—0.6065	34.30%—35.27%
$SRDCA$	0.2953—0.3074	0.5992—0.6075	34.05%—35.03%
$MRDCA$	0.2679—0.2802	0.5861—0.5959	35.28%—36.38%
$MRDCA-RWG$	0.2833—0.2942	0.5837—0.5932	35.14%—35.96%
$MRDCA-RWL$	0.2811—0.2934	0.5764—0.5887	35.38%—36.53%
$CARD-R$	0.1587—0.1662	0.4985—0.5066	43.12%—44.00%

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 [37] to have the same dynamic range.

Each 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 hard cluster partitions (obtained from the fuzzy partitions given by $NERF$ or $CARD-R$ or obtained directly from $SRDCA$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$)

were compared with the known a priori class partition. The comparison criteria used were the corrected Rand index (CR), the $F - measure$ and the overall error rate of classification ($OERC$). The CR, $F - measure$, and $OERC$ indexes were calculated for the best result.

3.2.1 Abalone data set

This data set consists of 4177 abalones described by 8 real-valued attributes and 1 nominal attribute. In this application, the 8 real-valued attributes were considered for clustering purposes. They are: (1) Length, (2) Diameter, (3) Height, (4) Whole weight, (5) Shucked weight, (6) Viscera weight, (7) Shell weight and (8) Rings. The nominal attribute "Sex" with three classes (Male, Female and Infant) was used as an *a priori* classification. The classes (Male, Female and Infant) have 1528, 1307 and 1342 instances, respectively.

The fuzzy clustering algorithms $NERF$ and $CARD - R$ were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster fuzzy partition. The three-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori three-class partition. $NERF$ had 0.0851, 0.4566 and 51.98% for CR, $F - measure$, and $OERC$ indexes, respectively, whereas $CARD - R$ had 0.0935, 0.5021 and 52.09%, respectively, for these indexes.

The hard clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster hard partition. Table 6 shows the performance of the $SRDCA$, $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ algorithms on the abalone data set according to CR, $F - measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k = 1, 2, 3$).

For this data set, globally, the best performance was presented by $MRDCA - RWG$, $MRDCA - RWL$, $MRDCA$ and $CARD - R$, in this order. The worst performance was presented by $NERF$ and $SRDCA$, in this order. Note that the performance was stable for $SRDCA$ and worsened for $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$, with the increase of the cardinality of the prototypes.

Table 7 gives the vector of relevance weights, globally for all dissimilarity matrices (according to the best result given by $MRDCA - RWG$ algorithm with prototypes of cardinality 1) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA - RWL$ algorithm with prototypes of cardinality 1). Table 8 gives the confusion matrix of the three-cluster hard partition given by the $MRDCA - RWL$ algorithm with prototypes of cardinality 1.

Table 6
Abalone data set: CR, $F - measure$, and $OERC$ indexes

Indexes	$\|G_k\|$	$SRDCA$	$MRDCA$	$MRDCA - RWL$	$MRDCA - RWG$
CR	1	0.0855	0.1440	0.1555	0.1847
	2	0.0853	0.1438	0.1531	0.1809
	3	0.0855	0.1436	0.1531	0.1827
	5	0.0855	0.1419	0.1535	0.1809
	10	0.0855	0.1409	0.1531	0.1799
$F - measure$	1	0.4572	0.5398	0.5503	0.6025
	2	0.4570	0.5416	0.5500	0.6005
	3	0.4572	0.5422	0.5500	0.6018
	5	0.4572	0.5402	0.5502	0.6005
	10	0.4572	0.5397	0.5498	0.6010
$OERC$	1	51.92%	46.89%	46.58%	47.11%
	2	51.92%	46.82%	46.66%	47.28%
	3	51.92%	46.89%	46.66%	47.16%
	5	51.92%	47.04%	46.66%	47.33%
	10	51.92%	47.11%	46.68%	47.28%

Table 7
Abalone data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$		
		Cluster 1	Cluster 2	Cluster 3
Length	**1.0915**	0.2281	**5.8765**	**1.7714**
Diameter	**1.1227**	0.2361	**5.4532**	**1.6567**
Height	0.7615	0.1584	**1.6357**	0.7500
Whole weight	**5.8800**	**84.1655**	0.1887	0.2504
Shucked weight	0.3895	**12.7859**	0.1137	**13.6994**
Viscera weight	0.9774	0.7795	1.1872	0.8345
Shell weight	0.9745	0.6773	0.9422	0.8894
Rings	0.4910	0.2061	0.7943	0.1783

Table 8
Abalone data set: confusion matrix

	Classes		
Clusters	1-Male	2-Female	3-Infant
1	372	256	1068
2	339	346	24
3	817	705	250

Concerning the three-cluster partition given by $MRDCA - RWG$, dissimilarity matrices were computed taking into account only " (4) Whole weight" and "(5) Shucked weight" attributes which had the highest (5.8800) and the lowest (0.3895) relevance weight in the definition of the clusters, respectively.

For the three-cluster hard partition given by the $MRDCA - RWL$ algorithm, Table 7 shows (in bold) the dissimilarity matrices of the most relevance weights in the definition of each cluster. For example, dissimilarity matrices computed taking into account only "(5) Shucked weight," "(1) Length" and "(2) Diameter" (in this order) are the most relevant in the definition of cluster 3 (Infant).

3.2.2 Image data set

This data set consists of images that were drawn randomly from a database of seven outdoor images. The images were segmented by hand to create the seven class labels: sky, cement, window, brick, grass, foliage and path. Each class has 330 instances. Each object is described by 16 real-valued attributes. These attributes are: (1) region-centroidcol; (2) region-centroid-row; (3) vedge-mean; (4) vegde-sd; (5) hedge-mean; (6) hedge-sd; (7) intensity-mean; (8) rawred-mean; (9) rawblue-mean; (10) rawgreen-mean; (11) exred-mean; (12) exblue-mean; (13) exgreen-mean; (14) value-mean; (15) saturation-mean and (16) hue-mean.

The fuzzy clustering algorithms $NERF$ and $CARD - R$ were applied to the dissimilarity matrices obtained from this data set to obtain a seven-cluster fuzzy partition. The seven-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori seven-class partition. $NERF$ had 0.2822, 0.5014 and 47.09% for CR, $F - measure$, and $OERC$ indexes, respectively, whereas $CARD - R$ had 0.0528, 0.3051, and 71.47% for these indexes, respectively.

The hard clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a seven-cluster hard partition. Table 9 shows the performance of the $SRDCA$, $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ algorithms on the image data set according to CR, $F - measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k = 1, \ldots, 7$).

For this data set, globally, the best performance was presented by $MRDCA - RWL$, $MRDCA$, $MRDCA - RWG$ and $SRDCA$, in this order. The worst performance was presented by $CARD - R$ and $NERF$, in this order. In particular, $MDCA - RWL$ with prototypes of cardinality 3 had the best and $CARD - R$ had the worst performance, concerning these indexes. Note that the performance was improved for $SRDCA$ and worsened for $MRDCA$,

$MRDCA-RWL$ and $MRDCA-RWG$, with the increase of the cardinality of the prototypes.

Table 9
Image data set: CR, $F-measure$, and $OERC$ indexes

Indexes	$\lvert G_k\rvert$	$SRDCA$	$MRDCA$	$MRDCA-RWL$	$MRDCA-RWG$
CR	1	0.3116	0.4756	0.4962	0.4382
	2	0.3909	0.4698	0.4947	0.4397
	3	0.3919	0.4603	0.4974	0.4371
	5	0.3223	0.4587	0.4948	0.4123
	10	0.4100	0.4568	0.4949	0.4128
$F-measure$	1	0.5310	0.6342	0.6490	0.6187
	2	0.6116	0.6300	0.6496	0.6101
	3	0.5869	0.6253	0.6527	0.6097
	5	0.5469	0.6237	0.6533	0.5817
	10	0.6193	0.6225	0.6528	0.5841
$OERC$	1	49.48%	38.70%	38.00%	40.12%
	2	44.80%	38.96%	38.05%	37.09%
	3	44.80%	39.61%	37.96%	37.14%
	5	50.04%	39.61%	38.81%	39.69%
	10	41.21%	39.61%	38.26%	39.69%

Table 10 gives the vector of relevance weights, globally for all dissimilarity matrices (according to the best result given by $MRDCA-RWG$ algorithm with prototypes of cardinality 3) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA-RWL$ algorithm with prototypes of cardinality 3). Table 11 gives the confusion matrix of the seven-cluster hard partition given by the $MRDCA-RWL$ algorithm with prototypes of cardinality 3.

Concerning the seven-cluster partition given by $MRDCA-RWG$, dissimilarity matrices computed taking into account only " (16) hue-mean" and "(1) region-centroidcol" attributes, had the highest (6.3530) and the lowest (0.1392) relevance weight in the definition of the clusters, respectively.

For the seven-cluster hard partition given by $MRDCA-RWL$ algorithm, Table 10 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. For example, dissimilarity matrices computed taking into account only "(4) vegde-sd," "(6) hegde-sd," "(16) hue-mean," and "(3) vegde-mean" (in this order) are the most relevant in the definition

Table 10
Image data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$						
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
region-centroidcol	0.1392	0.0329	0.0433	0.5251	0.0449	0.0764	0.0381	0.1138
region-centroid-row	0.2895	0.1435	0.0309	**5.9588**	0.2166	0.5125	0.1667	0.1974
vedge-mean	0.2570	0.4840	0.1364	0.0867	**2.9086**	0.0699	0.4695	0.0756
vegde-sd	0.6346	**22.1196**	**48.4827**	0.0124	**565.5961**	0.0847	**135.0916**	**15.1192**
hedge-mean	0.2146	0.3846	0.1625	0.0289	0.9601	0.0884	**1.0664**	0.0424
hedge-sd	0.2884	**21.7621**	**69.8531**	0.0067	**208.6723**	0.1796	**207.1898**	0.2895
intensity-mean	**3.9574**	**2.1687**	**1.3030**	**3.8936**	0.3634	**7.1007**	**1.7675**	**3.4010**
rawred-mean	**3.0212**	**2.5154**	**1.3864**	**3.5434**	0.2303	**8.3460**	1.5910	**2.8607**
rawblue-mean	**4.9499**	**2.5363**	**1.0718**	**4.1286**	0.8258	**5.7401**	**1.3864**	**4.0582**
rawgreen-mean	**3.4086**	**1.4650**	**1.4573**	**3.8519**	0.2640	**7.2421**	**2.4677**	**2.9034**
exred-mean	0.6573	0.3574	0.2632	**2.3650**	0.0949	**1.0620**	0.2845	0.5664
exblue-mean	**1.1291**	**1.1280**	0.3342	**3.9761**	0.1189	**1.7735**	0.4035	**1.4148**
exgreen-mean	**1.2551**	0.3515	0.7094	**1.7040**	0.1769	**2.4896**	0.3493	**1.5726**
value-mean	**4.7500**	**2.0473**	**1.0422**	**4.0145**	0.8030	**5.5291**	**1.3557**	**3.9385**
saturation-mean	0.4329	0.2982	**2.1785**	**3.7801**	0.6078	0.2884	0.0501	0.8251
hue-mean	**6.3530**	**1.3434**	**24.8247**	**28.2962**	**17.4598**	**14.6969**	0.4270	6.7460

Table 11
Image data set: confusion matrix

	Classes						
Clusters	1-sky	2-cement	3-window	4-brick	5-grass	6-foliage	7-path
1	0	0	22	37	1	63	0
2	0	0	1	215	0	252	0
3	146	0	200	13	198	0	1
4	0	330	0	0	0	0	0
5	184	0	30	64	103	13	2
6	0	0	77	1	28	2	0
7	0	0	0	0	0	0	327

of cluster 4 (cement).

3.2.3 Iris plant data set

This data set consists of three types (classes) of iris plants: iris setosa, iris versicolour 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 $NERF$ and $CARD - R$ were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster fuzzy partition. The three-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori three-class partition. $NERF$ had 0.7294, 0.8922 and 10.67% for CR, $F - measure$, and $OERC$ indexes, respectively, whereas $CARD - R$ had 0.8856, 0.9599 and 4.00% for these indexes, respectively.

The hard clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster hard partition. Table 12 shows the performance of the $SRDCA$, $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ algorithms on the iris data set according to CR, $F-measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k=1,2,3$).

For this data set, globally, the best performance was presented by $CARD-R$, $MRDCA-RWG$, $MRDCA-RWL$ and $SRDCA$, in this order. The worst performance was presented by $MRDCA$ and $NERF$, in this order. Note that the performance was improved for $MRDCA$ and worsened for $SRDCA$ and $MRDCA-RWL$, with the increase of the cardinality of the prototypes. The performance of $MRDCA-RWL$ was not affected by the cardinality of the prototypes.

Table 12
Iris data set: CR, $F-measure$, and $OERC$ indexes

Indexes	$\|G_k\|$	$SRDCA$	$MRDCA$	$MRDCA-RWL$	$MRDCA-RWG$
CR	1	0.7455	0.6412	0.8680	0.8856
	2	0.7037	0.6412	0.8680	0.8856
	3	0.7302	0.6575	0.8507	0.8856
	5	0.7294	0.6575	0.8342	0.8856
	10	0.7436	0.6451	0.8681	0.8856
$F-measure$	1	0.8976	0.8465	0.9533	0.9599
	2	0.8782	0.8465	0.9533	0.9599
	3	0.8917	0.8600	0.9466	0.9599
	5	0.8922	0.8600	0.9398	0.9599
	10	0.8987	0.8535	0.9532	0.9599
$OERC$	1	10.00%	15.33%	4.67%	4.00%
	2	12.00%	15.33%	4.67%	4.00%
	3	10.67%	14.00%	5.33%	4.00%
	5	10.67%	14.00%	6.00%	4.00%
	10	10.00%	14.67%	4.67%	4.00%

Table 13 gives the vector of relevance weights, globally for all dissimilarity matrices (according to the best result given by $MRDCA-RWG$ algorithm with prototypes of cardinality 1) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA-RWL$ algorithm with prototypes of cardinality 1). Table 14 gives the confusion matrix of the three-cluster hard partition given by the $MRDCA - RWL$ algorithm with prototypes of cardinality 1.

Table 13
Iris data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$		
		Cluster 1	Cluster 2	Cluster 3
Sepal length	0.5523	0.4215	0.4423	0.4145
Sepal width	0.2971	0.5146	0.3555	0.0994
Petal length	**2.9820**	**2.3212**	**2.0378**	**7.3868**
Petal width	**2.0428**	**1.9861**	**3.1202**	**3.2822**

Table 14
Iris data set: confusion matrix

	Classes		
Clusters	1-Iris setosa	2-Iris versicolour	3-Iris virginica
1	50	0	0
2	0	3	46
3	0	47	4

Concerning the three-cluster partition given by $MRDCA - RWG$, dissimilarity matrices computed taking into account only "(3) petal length" or only "(4) petal width" attributes have the highest relevant weight. Thus the objects described by these dissimilarity matrices are closer to the prototypes of the clusters than are those described by dissimilarity matrices computed taking into account only "(1) sepal length" or "(2) sepal width" attributes.

For the three-cluster hard partition given by $MRDCA - RWL$ algorithm, Table 13 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. For example, dissimilarity matrices computed taking into account only "(3) Petal length" and "(4) Petal width" (in this order) are the most relevant in the definition of cluster 3 (Iris setosa), whereas dissimilarity matrices computed taking into account only "(4) Petal width" and "(3) Petal length" are the most relevant in the definition of cluster 2 (Iris versicolour).

3.2.4 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 thyroxin, (3) total serum triiodothyronine, (4) basal thyroid-stimulating hormone (TSH) and (5) maximal absolute difference in TSH value.

The fuzzy clustering algorithms $NERF$ and $CARD - R$ were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster fuzzy partition. The three-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori three-class partition. $NERF$ had 0.4413, 0.7993 and 20.93% for CR, $F - measure$, and $OERC$ indexes, respectively, whereas $CARD - R$ had 0.2297, 0.7160 and 21.86% for these indexes, respectively.

The hard clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster hard partition. Table 15 shows the performance of the $SRDCA$, $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ algorithms on the thyroid data set according to CR, $F - measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k = 1, 2, 3$).

For this data set, globally, the best performance was presented by $MRDCA - RWL$, $MRDCA - RWG$, and $MRDCA$, in this order. The worst performance was presented by $CARD - R$, $NERF$ and $SRDCA$, in this order. In particular, $MRDCA - RWL$ with prototypes of cardinality 1 had the best and $SRDCA$ with prototypes of cardinality 10 had the worst performance, concerning these indexes. Note that the performance was stable for $MRDCA - RWG$ and was worsened for $MRDCA - RWL$, with the increase of the cardinality of the prototypes. Performance of $SRDCA$ was better with prototypes of cardinality 2, 3 and 5 and worst with prototypes of cardinality 10. Finally, the performance of $MRDCA$ was worst with prototypes of cardinality 2 and better with prototypes of cardinality 3, 5 and 10.

Table 16 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by $MRDCA - RWG$ algorithm with prototypes of cardinality 1) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA - RWL$ algorithm with prototypes of cardinality 1). Table 17 gives the confusion matrix of the three-cluster hard partition given by the $MRDCA - RWL$ algorithm with prototypes of cardinality 1.

Concerning the three-cluster partition given by $MRDCA - RWG$, dissimilarity matrices computed taking into account only “ (2) Total serum thyroxin” and “(1) T3-resin uptake test” attributes, had the highest (1.3982) and the lowest (0.6546) relevance weight in the definition of the clusters, respectively.

For the three-cluster hard partition given by $MRDCA - RWL$ algorithm, Table 16 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. For example, dissimilarity matrices computed taking into account only “(5) Maximal absolute difference in TSH value” and“(4) Basal thyroid-stimulating hormone (TSH)” (in this order) are the most relevant in the definition of cluster 2 (Hyperthyroidism), whereas dis-

Table 15
Thyroid data set: CR, $F - measure$, and $OERC$ indexes

Indexes	$\lvert G_k \rvert$	$SRDCA$	$MRDCA$	$MRDCA - RWL$	$MRDCA - RWG$
CR	1	0.3577	0.5665	0.8776	0.5809
	2	0.5217	0.5484	0.8475	0.5484
	3	0.5217	0.5809	0.8328	0.5809
	5	0.6285	0.5974	0.8182	0.5809
	10	0.2014	0.5831	0.8185	0.5831
$F - measure$	1	0.7709	0.8551	0.9616	0.8614
	2	0.8408	0.8508	0.9525	0.8508
	3	0.8408	0.8614	0.9481	0.8614
	5	0.8820	0.8665	0.9437	0.8614
	10	0.6764	0.8602	0.9429	0.8602
$OERC$	1	24.65%	13.02%	3.72%	12.55%
	2	15.81%	13.48%	4.65%	13.48%
	3	15.81%	12.55%	5.11%	12.55%
	5	11.62%	12.09%	5.58%	12.55%
	10	35.34%	12.55%	5.58%	12.55%

Table 16
Thyroid data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$		
		Cluster 1	Cluster 2	Cluster 3
T3-resin uptake test	0.6546	0.2437	0.0599	**1.7284**
Total serum thyroxin	**1.3982**	0.4086	0.0933	**4.9804**
Total serum triiodothyronine	0.9716	0.8272	0.0488	**5.2643**
Basal thyroidstimulating hormone (TSH)	**1.1822**	**12.93**	**29.3203**	0.1350
Maximal absolute difference in TSH value	0.9509	0.9136	**124.6778**	0.1633

Table 17
Thyroid data set: confusion matrix

	Classes		
Clusters	1-Normal	2-Hyperthyroidism	3-Hypothyroidism
1	148	0	6
2	2	35	0
3	0	0	24

similarity matrices computed taking into account only "(3) Total serum triiodothyronine," "(2) Total serum thyroxin," and "(1) T3-resin uptake test" are the most relevant in the definition of cluster 3 (Hypothyroidism).

3.2.5 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 $NERF$ and $CARD - R$ were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster fuzzy partition. The three-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori three-class partition. $NERF$ had 0.3539, 0.6986 and 31.46% for CR, $F - measure$, and $OERC$ indexes, respectively, whereas $CARD - R$ had 0.3808, 0.7227 and 26.97% for these indexes, respectively.

The hard clustering algorithms were applied to the dissimilarity matrices obtained from this data set to obtain a three-cluster hard partition. Table 18 shows the performance of the $SRDCA$, $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ algorithms on the wine data set according to CR, $F - measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k = 1, 2, 3$).

For this data set, globally, the best performance was presented by $MRDCA$, $MRDCA - RWG$, $MRDCA - RWL$, and $CARD - R$, in this order. The worst performance was presented by $NERF$ and $SRDCA$, in this order. In particular, $MRDCA$ with prototypes of cardinality 5 or 10 had the best and $NERF$ had the worst performances, concerning these indexes. Note that the performance was worsened for $SRDCA$ and was improved for $MRDCA$, $MRDCA-RWL$, and $MRDCA-RWG$, with the increase of the cardinality of the prototypes.

Table 19 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by $MRDCA - RWG$ algorithm with prototypes of cardinality 10) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA - RWL$ algorithm with prototypes of cardinality 5). Table 20 gives the confusion matrix of the three-cluster hard partition given by the $MRDCA - RWL$ algorithm with prototypes of cardinality 5.

Concerning the three-cluster hard partition given by $MRDCA - RWG$, dissimilarity matrices computed taking into account only “ (7) Flavonoids” and “(3) Ash” attributes, had the highest and the lowest relevance weight in the

Table 18
Wine data set: CR, $F - measure$, and $OERC$ indexes

Indexes	$\|G_k\|$	$SRDCA$	$MRDCA$	$MRDCA - RWL$	$MRDCA - RWG$
CR	1	0.3749	0.7263	0.7407	0.7548
	2	0.3711	0.8297	0.7702	0.8150
	3	0.3749	0.8319	0.7553	0.8319
	5	0.3711	0.8804	0.7712	0.8185
	10	0.3711	0.8804	0.7702	0.8348
$F - measure$	1	0.7204	0.9024	0.9077	0.9138
	2	0.7147	0.9435	0.9195	0.9372
	3	0.7204	0.9429	0.9136	0.9429
	5	0.7147	0.9603	0.9194	0.9371
	10	0.7147	0.9603	0.9195	0.9430
$OERC$	1	29.21%	9.55%	8.98%	8.42%
	2	29.77%	5.61%	7.86%	6.17%
	3	29.21%	5.61%	8.42%	5.61%
	5	29.77%	3.93%	7.86%	6.17%
	10	29.77%	3.93%	7.86%	5.61%

Table 19
Wine data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$		
		Cluster 1	Cluster 2	Cluster 3
Alcohol	**1.1425**	0.8661	0.6262	**1.4453**
Malic acid	0.7764	0.4632	**1.6446**	0.5761
Ash	0.5881	0.8849	0.4594	0.4902
Alkalinity of ash	0.6648	0.6879	0.5142	0.5693
Magnesium	0.5914	0.6026	0.4912	0.4559
Total phenols	**1.2453**	**1.0469**	**1.5799**	**1.0290**
Flavonoids	**2.5725**	**4.4077**	**2.5278**	**2.5297**
Non-flavonoid phenols	0.7232	0.5121	**1.6356**	0.6200
Proanthocyanins	0.7954	0.8521	0.8685	0.6673
Color intensity	0.9707	0.2827	**1.5606**	**4.6254**
Hue	0.9462	**1.3060**	**1.2557**	0.5543
OD280/OD315 of diluted wines	**1.7677**	**3.3920**	**1.4217**	0.9359
Proline	**1.6284**	**2.6922**	0.5292	**3.6505**

definition of the clusters, respectively.

For the three-cluster hard partition given by $MRDCA - RWL$ algorithm,

Table 20
Wine data set: confusion matrix

Clusters	Classes		
	Wine type 1	Wine type 2	Wine type 3
1	59	8	0
2	0	57	0
3	0	6	48

Table 19 shows (in bold) the dissimilarity matrices of most relevance weights in the definition of each cluster. For example, dissimilarity matrices computed taking into account only "(7) Flavonoids," "(12) OD280/OD315 of diluted wines," "(13) Proline," "(11) Hue," and "(6) Total phenols" (in this order) are the most relevant in the definition of cluster 1 (Wine type 1).

In conclusion, for these UCI machine learning data sets, the best performance was presented by $MRDCA - RWL$, $MRDCA - RWG$, $MRDCA$, and $CARD - R$, in this order, according to CR, $F - measure$, and $OERC$ indexes. The worst performance was presented by $NERF$ and $SRDCA$, in this order. Moreover, when the cardinality of the prototypes was increased, in the majority of the data sets, the performance was worsened for $MRDCA - RWL$, was worsened or stable for $MRDCA - RWG$ and $SRDCA$, and was improved for $MRDCA$.

3.3 Time trajectories data sets

The authors consider phoneme and satellite time trajectories data sets. These data sets are available at http://www.math.univ-toulouse.fr/staph/npfda/npfda-datasets.html. To compare time trajectories, a "cross sectional-longitudinal" dissimilarity function proposed by D'Urso and Vichi was considered [35] [36]. The authors propose a compromise dissimilarity that is a combination of a cross-sectional dissimilarity, which compares the instantaneous position (trend) of each pair of trajectories, and two longitudinal dissimilarities, based on the concepts of velocity and acceleration of a time trajectory.

Let $\mathbf{x}_i = (x_i(t_1), \ldots, x_i(t_p))$ $(i = 1, \ldots, n)$ the i-th time trajectory. The velocity of the i-th time trajectory is defined as $\mathbf{v}_i = (v_i(t_2), \ldots, v_i(t_p))$ $(i = 1, \ldots, n)$, where $v_i(t_j) = \frac{x_i(t_j) - x_i(t_{j-1})}{t_j - t_{j-1}}$ $(j = 2 \ldots, p)$ is the velocity in the interval $[t_{j-1}, t_j)$ which measures the variation of the i-th time trajectory in $[t_{j-1}, t_j)$. The acceleration of the i-th time trajectory is defined as $\mathbf{a}_i = (a_i(t_3), \ldots, a_i(t_p))$ $(i = 1, \ldots, n)$, where $a_i(t_j) = \frac{v_i(t_j) - v_i(t_{j-1})}{t_j - t_{j-2}}$ $(j = 3 \ldots, p)$ is the acceleration in the interval $[t_{j-2}, t_j)$.

The compromise dissimilarity between the i-th and the l-th time trajectories is defined as

$$d^2(i,l) = \alpha_1 \|\mathbf{x}_i - \mathbf{x}_l\|^2 + \alpha_2 \|\mathbf{v}_i - \mathbf{v}_l\|^2 + \alpha_3 \|\mathbf{a}_i - \mathbf{a}_l\|^2 \tag{25}$$

where $\|\mathbf{x}_i - \mathbf{x}_l\| = \sum_{j=1}^{p}(x_i(t_j) - x_l(t_j))^2$, $\|\mathbf{v}_i - \mathbf{v}_l\| = \sum_{j=2}^{p}(v_i(t_j) - v_l(t_j))^2$ and $\|\mathbf{a}_i - \mathbf{a}_l\| = \sum_{j=3}^{p}(a_i(t_j) - a_l(t_j))^2$.

In [35], the weights α of each dissimilarity-component are determined by considering a global objective criterion based on the maximization of the variance of the compromise dissimilarity. In this paper, they will be determined according to the relational clustering algorithm presented in sections 2.2.1 and 2.2.2.

Note that because they are based on a single dissimilarity matrix, neither $NERF$ nor $SRDCA$ can be used to cluster time trajectories data sets compared to the "cross sectional-longitudinal" dissimilarity function proposed by D'Urso and Vichi [35] [36].

3.4 Phoneme data set

This data set is a part of the original one that can be found at http://www-stat.stanford.edu/ tibs/ElemStatLearn/. It consists of five phonemes (classes): "sh," "iy," "dcl," "aa," and "ao". The five classes each have 400 instances (objects). Each object (time trajectory) is described as $(\mathbf{x}_i, y_i)$ $(i = 1 \ldots, n)$, where y_i gives the class membership (phonemes) whereas $\mathbf{x}_i = (x_i(t_1), \ldots, x_i(t_{150}))$ is the i-th discretized functional data corresponding to the discretized log-periodograms.

From the original phoneme data set the authors had obtained initially two additional data sets corresponding to the velocity and acceleration of the discretized log-periodograms. Then, three relational data tables are obtained from these three satellite data sets (position, velocity and acceleration of the discretized log-periodograms) through the application of the squared Euclidean distance. All dissimilarity matrices were normalized according to their overall dispersion [37] to have the same dynamic range.

The fuzzy clustering algorithm $CARD - R$ was performed simultaneously on these 3 relational data tables (position , velocity, and acceleration of the discretized log-periodograms) to obtain a five-cluster fuzzy partition. The five-cluster hard partitions obtained from the fuzzy partition were compared with the known a priori five-class partition. $CARD - R$ had 0.1922, 0.4853 and 60.40% for the CR, $F - measure$, and $OERC$ indexes, respectively.

The hard clustering algorithms $MRDCA$, $MRDCA - RWL$ and $MRDCA - RWG$ were applied simultaneously on these 3 relational data tables to obtain a five-cluster hard partition. Table 21 shows the performance of the $MRDCA$,

$MRDCA-RWL$ and $MRDCA-RWG$ algorithms on the phoneme data set according to the CR, $F-measure$ and $OERC$ indexes, considering prototypes of cardinality $|G_k|$ =1, 2, 3, 5 and 10 ($k = 1, \ldots, 5$).

For this data set, globally, the best performance was presented by $MRDCA-RWL$, $MRDCA-RWG$ and $MRDCA$, in this order. The worst performance was presented by $CARD-R$. In particular, $MRDCA-RWG$ with prototypes of cardinality 10 had the best performance, concerning these indexes. Note that the performance was improved for $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$, with the increase of the cardinality of the prototypes.

Table 21
Phoneme data set: CR, $F-measure$, and $OERC$ indexes

Indexes	$\|G_k\|$	$MRDCA$	$MRDCA-RWL$	$MRDCA-RWG$
CR	1	0.4366	0.5418	0.5216
	2	0.4284	0.5835	0.5964
	3	0.5317	0.6972	0.6937
	5	0.4812	0.7270	0.7225
	10	0.4698	0.7264	0.7277
$F-measure$	1	0.6496	0.7435	0.7441
	2	0.6714	0.7675	0.7708
	3	0.7331	0.8448	0.8433
	5	0.6501	0.8550	0.8525
	10	0.6484	0.8495	0.8492
$OERC$	1	38.65%	27.10 %	28.55%
	2	34.50%	26.30%	25.45%
	3	30.15 %	15.70%	16.00%
	5	36.25%	14.70%	14.95%
	10	39.15%	15.10%	15.15%

Table 22 gives the vector of relevance weights globally for all dissimilarity matrices (according to the best result given by $MRDCA-RWG$ algorithm with prototypes of cardinality 10) and locally for each cluster and dissimilarity matrix (according to the best result given by $MRDCA-RWL$ algorithm with prototypes of cardinality 5). Table 23 gives the confusion matrix of the five-cluster hard partition given by the $MRDCA-RWL$ algorithm with prototypes of cardinality 5.

Concerning the five-cluster hard partition given by $MRDCA-RWG$, dissimilarity matrices computed taking into account only " (1) Position" attribute

Table 22
Phoneme data set: vectors of relevance weights

Data Matrix	$MRDCA-RWG$	$MRDCA-RWL$				
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Position	**2.1888**	**2.5936**	**1.5062**	**2.3235**	**2.1424**	**2.0930**
Velocity	0.6900	0.6458	0.8203	0.6451	0.7102	0.7091
Acceleration	0.6621	0.5969	0.8093	0.6670	0.6571	0.6736

Table 23
Phoneme data set: confusion matrix

	Classes				
Clusters	1-sh	2-iy	3-dcl	4-aa	5-ao
1	0	1	387	0	1
2	396	9	1	0	0
3	0	0	0	271	115
4	0	21	9	129	283
5	4	369	3	0	1

had the highest relevance weight in the definition of the clusters. Thus, the objects described by this dissimilarity matrix are closer to the prototypes of the clusters than are those described by velocity or acceleration dissimilarity matrix.

For the five-cluster hard partition given by $MRDCA-RWL$ algorithm, Table 19 shows (in bold) the dissimilarity matrices of the most relevance weights in the definition of each cluster. For all clusters, position dissimilarity matrix has the highest relevant weight, thus the objects described by this dissimilarity matrix are closer to the respective prototypes of these clusters than are those described by velocity or acceleration dissimilarity matrices.

3.5 Satellite data set

This data set concerns $n = 472$ radar waveforms. The data were registered by the Topex/Poseidon satellite upon the Amazon River. Each object (time trajectory) is represented by its discretized wave version $\mathbf{x}_i = (x_i(t_1), \ldots, x_i(t_{70}))$ $(i = 1, \ldots, 472)$. Each wave is linked with the kind of ground treated by the satellite, and the aim is to use these waveforms for altimetric and hydrological purpose on the Amazonian basin.

From the original satellite data set, the authors obtained initially 2 additional data sets corresponding to the velocity and acceleration of the radar waveforms. Then, 3 relational data tables are obtained from these 3 satellite data sets (position, velocity and acceleration of the radar waveforms) through the application of the squared Euclidean distance. All dissimilarity matrices were normalized according to their overall dispersion [37] to have the same dynamic range.

The clustering algorithm has been performed simultaneously on these 3 relational data tables (position, velocity and acceleration of the radar waveforms) to obtain a partition in $K = \{1, \dots, 10\}$. For a fixed number of clusters K, the clustering algorithm is run 100 times and the best result according to the adequacy criterion is selected.

To determine the number of cluster, the authors used the approach described by [38], which consists of the choice of the peaks on the graph of the "second-order differences" of the clustering criterion (equation (5)): $J^{(K-1)} + J^{(K+1)} - 2J^{(K)}$, $K = 2, \dots, 9$. According to this approach, the number of clusters was fixed as 7. Algorithm $MRDCA-RWG$ gives 7 clusters with cardinality of 49, 55, 45, 79, 32, 149 and 63, while algorithm $MRDCA - RWL$ gives 7 clusters with cardinality of 38, 84, 61, 97, 62, 92 and 38. For both algorithms, the prototypes have cardinality 5.

Table 24 gives the vector of relevance weights globally for each dissimilarity matrix (according to the algorithm $MRDCA - RWG$) and locally for each cluster and dissimilarity matrix (according to the algorithm $MRDCA-RWL$). Concerning the seven-cluster partition given by $MRDCA - RWG$, position dissimilarity matrix has the highest relevant weight.

Table 24
Satellite data set: vectors of relevance weights

Data Matrix	$MRDCA - RWG$	$MRDCA - RWL$						
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Position	**1.4309**	**3.4372**	0.7862	**1.6756**	**2.3028**	**2.6052**	**2.1688**	0.7543
Velocity	0.8447	0.5974	**1.1387**	0.7634	0.6757	0.6460	0.6997	**1.0608**
Acceleration	0.8272	0.4869	**1.1168**	0.7817	0.6425	0.5940	0.6588	**1.2496**

For clusters 1, 3, 4, 5 and 6 of the seven-cluster partition given by $MRDCA-RWL$, position dissimilarity matrix had the highest relevant weight, while for cluster 2, velocity and acceleration dissimilarity matrices, in this order, had the highest relevant weights. Finally, for cluster 7, acceleration and velocity dissimilarity matrices, in this order, had the highest relevant weights.

Figure 1 shows selected curves from the original satellite data set (position) belonging to each of the seven clusters. The 5 curves in the prototype of each cluster (1 and 7) are drawn in bold. This figure shows clearly the difference between the clusters.

Figures 2-4 show selected curves from the original satellite data set (position) as well as from the additional data sets (velocity and acceleration) belonging to clusters 1 (where position dissimilarity matrix had the highest relevant weight among the seven clusters) and 7 (where acceleration and velocity dissimilarity matrices were more relevant than position dissimilarity matrix). These figures

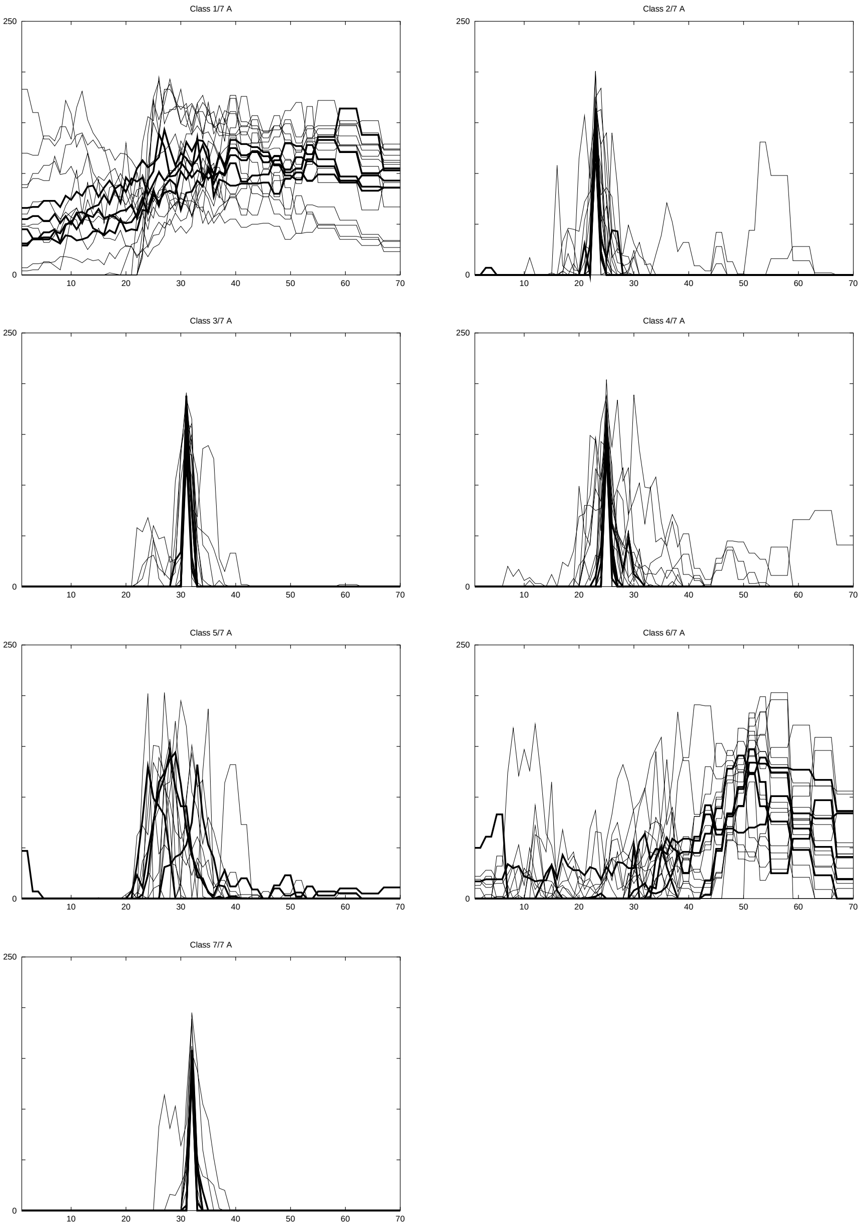

Fig. 1. Selected curves of the clusters: original satellite data set (position)

illustrate clearly why position is the most relevant dissimilarity matrix for cluster 1 whereas acceleration and velocity are the most relevant dissimilarity matrix for cluster 7. In these figures, the 5 curves in the prototype of each cluster (1 and 7) were also drawn in bold.

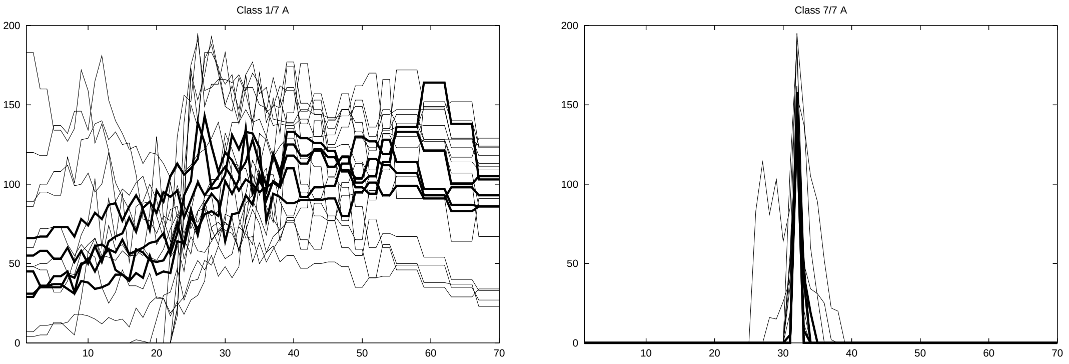

Fig. 2. Selected curves of Clusters 1 and 7: Position

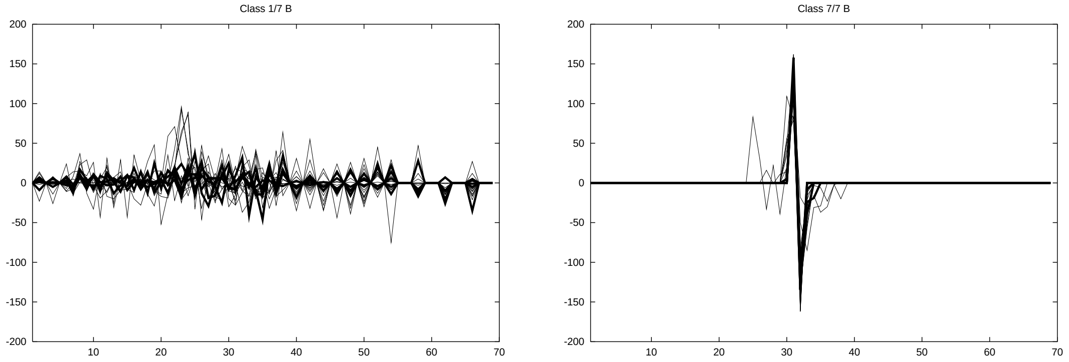

Fig. 3. Selected curves of Clusters 1 and 7: Velocity

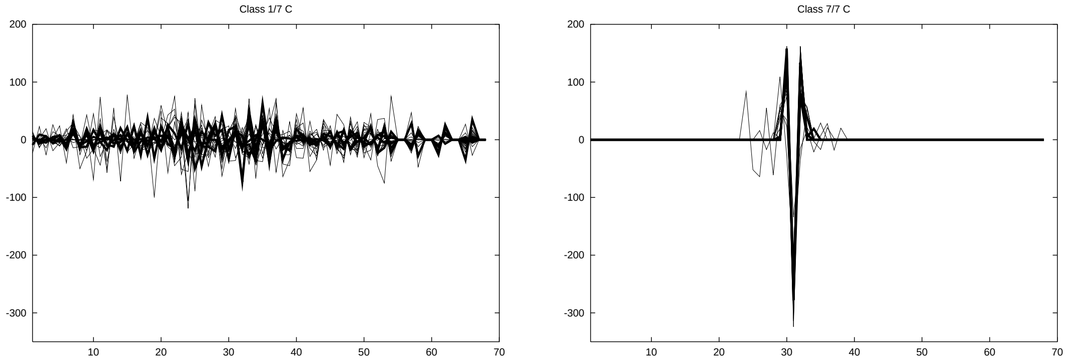

Fig. 4. Selected curves of Clusters 1 and 7: Acceleration

4 Concluding remarks

This paper extended the dynamic clustering algorithm for relational data ($SRDCA$) into hard clustering algorithms ($MRDCA-RWL$ and $MRDCA-RWG$) that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. These matrices have been generated using different sets of variables and dissimilarity functions. These algorithms are designed to furnish a partition and a prototype for each cluster as well as a relevance weight for each dissimilarity matrix by optimizing an adequacy criterion that measures the fitting between clusters and their representatives. As a particularity of these clustering algorithms, they assume that the prototype of each 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 cluster, the best relevance weight of each dissimilarity matrix as well as the best partition, according to the clustering criterion. Moreover, the time complexity and the convergence properties of $MRDCA - RWL$ and $MRDCA - RWG$ are also presented. Concerning the relevance weights, they 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 to the prototype the objects of a given dissimilarity matrix of a given cluster are, the higher is the relevance weight of this dissimilarity matrix on this cluster.

The usefulness of these partitioning relational hard clustering algorithms was shown with data sets (synthetic and from UCI machine learning repository) described by real-valued variables as well as with time trajectory data sets. The accuracy of the results furnished by $MRDCA - RWL$ and $MRDCA - RWG$ algorithms on these data sets was assessed by the corrected Rand index, the F-measure and the overall error rate of classification.

Concerning the synthetic data sets, the performance of $MRDCA - RWL$ and $MRDCA - RWG$ depends on the dispersion of the variables that describes the objects. In comparison with the algorithms $NERF$ and $SRDCA$, which perform on a single dissimilarity matrix, $MRDCA - RWL$ was clearly superior in the synthetic data sets where the variance was different between the variables whereas $MRDCA - RWG$ was clearly superior only in the synthetic data sets where the variance was different between the variables but almost the same from one class to another.

Moreover, for the UCI machine learning data sets, the best performance was presented by $MRDCA - RWL$, $MRDCA - RWG$, $MRDCA$ and $CARD - R$, in this order. The worst performance was presented by $NERF$ and $SRDCA$ (algorithms that perfoms on a single dissimilarity matrix). Moreover, when the cardinality of the prototypes was increased, in the majority of these data sets, the performance was worsened for $MRDCA - RWL$, was worsened or stable for $MRDCA - RWG$ and $SRDCA$ and was improved for $MRDCA$.

Phoneme and satellite time trajectory data sets compared through a “cross sectional-longitudinal” dissimilarity function also have been considered. Because this dissimilarity function, when applied to a data set, produces three dissimilarity matrices corresponding to the comparison of the trajectories according to the their trend, velocity and acceleration, only relational clustering algorithms that are able to manage multiple dissimilarity matrices can be considered. Thus, for the phoneme time trajectory data set, the best performance was presented by $MRDCA-RWL$, $MRDCA-RWG$ and $MRDCA$, in this order. The worst performance was presented by $CARD-R$. Moreover, the performance of $MRDCA$, $MRDCA-RWL$ and $MRDCA-RWG$ was improved with the increase of the cardinality of the prototypes. Finally, the usefulness of the algorithms $MRDCA-RWL$ and $MRDCA-RWG$ have also been illustrated with the study of the satellite time trajectory data set.

References

[1] A. K. Jain, M.N. Murty, P.J. Flynn, Data Clustering: A Review, ACM Computing Surveys 31 (3) (1999) 264–323

[2] R. Xu, D. Wunsch, Survey of Clustering Algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678

[3] P.H. Sneath, R.R. Sokal, Numerical Taxonomy. Freeman, San Francisco, 1973

[4] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: An efficient data clustering method for very large databases, in Proc. ACM SIGMOD Conf. Management of Data, 1996, pp. 103114.

[5] S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for large databases, in Proc. ACM SIGMOD Int. Conf. Management of Data, 1998, pp. 7384.

[6] G. Karypis, E. Han, and V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, IEEE Computer 32 (8) (1999) 68–75

[7] S. Guha, R. Rastogi, and K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Information Systems 25 (5) (2000) 345–366

[8] G.N. Lance, W.T. Williams, Note on a new information statistic classification program, The Computer Journal 11 (1968) 195–197

[9] K.C. Gowda, G. Krishna, Disaggregative clustering using the concept of mutual nearest neighborhood, IEEE Transactions on Systems, Man, and Cybernetics 8 (1978) 888–895

[10] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data, Wiley, New York, 1990

[11] A. Guenoche, P. Hansen, B. Jaumard, Efficient algorithms for divisive hierarchical clustering, Journal of Classification 8 (1991) 5–30.

[12] M. Chavent, A monotetic clustering method, Pattern Recognition Letters 19 (1998) 989–996

[13] E. Forgy, Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768–780

[14] Z. Huang, Extensions to the K-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (1998) 283–304

[15] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, An efficient K-means clustering algorithm: Analysis and implementation, IEEE Transactions in Pattern Analysis Machine Intelligence 24 (7) (2000) 881–892

[16] P. Hansen and N. Mladenoviae, J-means: A new local search heuristic for minimum sum of squares clustering, Pattern Recognition 34 (2001) 405–413

[17] M. Su and C. Chou, A modified version of the K-means algorithm with a distance based on cluster symmetry, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 674–680

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

[19] F. Hoeppner, F. Klawonn, and R. Kruse, Fuzzy Cluster Analysis: Methods for Classification, Data Analysis, and Image Recognition. Wiley, New York, 1999.

[20] R. Hathaway, J. Bezdek, and Y. Hu, Generalized fuzzy c-means clustering strategies using L_p norm distances, IEEE Transactions on Fuzzy Systems 8 (5) (2000) 576–582

[21] M. Hung and D. Yang, An efficient fuzzy c-means clustering algorithm, in Proc. IEEE Int. Conf. Data Mining, 2001, 225–232.

[22] J. Kolen and T. Hutcheson, Reducing the time complexity of the fuzzy c-means algorithm, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 263–267

[23] 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

[24] F.A.T. De Carvalho, M.Csernel, Y. Lechevallier, Pattern Recognition Letters 30 (2009) 10371045

[25] J.W. Davenport, R.J. Hathaway, J.C. Bezdek, Relational duals of the c-means algorithms, Pattern Recognition 22 (1989) 205–212

[26] R.J. Hathaway, J.C. Bezdek, Nerf c-means: non-Euclidean relational fuzzy clustering, Pattern Recognition 27 (3) (1994) 429437

[27] H. Frigui, C. Hwanga, F. C.-H. Rhee, Clustering and aggregation of relational data with applications to image database categorization, Pattern Recognition, 40 (11) (2007) 3053 3068

[28] W. Pedrycz, Collaborative fuzzy clustering, Pattern Recognition Letters, 23, (2002) 675–686

[29] E. Diday, G. Govaert, Classification Automatique avec Distances Adaptatives, R.A.I.R.O. Informatique Computer Science 11 (4) (1977) 329–349.

[30] E. Diday, J.C. Simon, Clustering analysis, in K.S. Fu (ed), Digital Pattern Classification, Springer, Berlin, 1976, 47–94.

[31] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1985) 193–218

[32] C.J. van Rijisbergen, Information retrieval, Butterworth-Heinemann, London, 1979.

[33] L. Breiman, J. Friedman, C.J. Stone,R.A. Olshen, Classification and Regression Trees, Chapman and Hall/CRC, Boca Raton, 1984

[34] 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, 341–375, 1996

[35] P. D'Urso and M. Vichi, Dissimilarities between trajectories of a three-way longitudinal data set, in A. Rizzi, M. Vichi, H.-H. Bock, Advances in Data Science and Classification, Springer, Berlin, 585–592, 1998

[36] P. D'Urso, Dissimilarity measures for time trajectories, Journal of Italian Statistical Society 1 (3) (2000) 53–83

[37] M. Chavent, Normalized k-means clustering of hyper-rectangles, in: Proceedings of the XI International Symposium of Applied Stochastic Models and Data Analysis (ASMDA 2005), Brest, France, 2005, pp. 670–677

[38] A. Da Silva, Analyse de données évolutives: application aux données d'usage Web, Thèse de Doctorat, Université Paris-IX Dauphine, 2009.