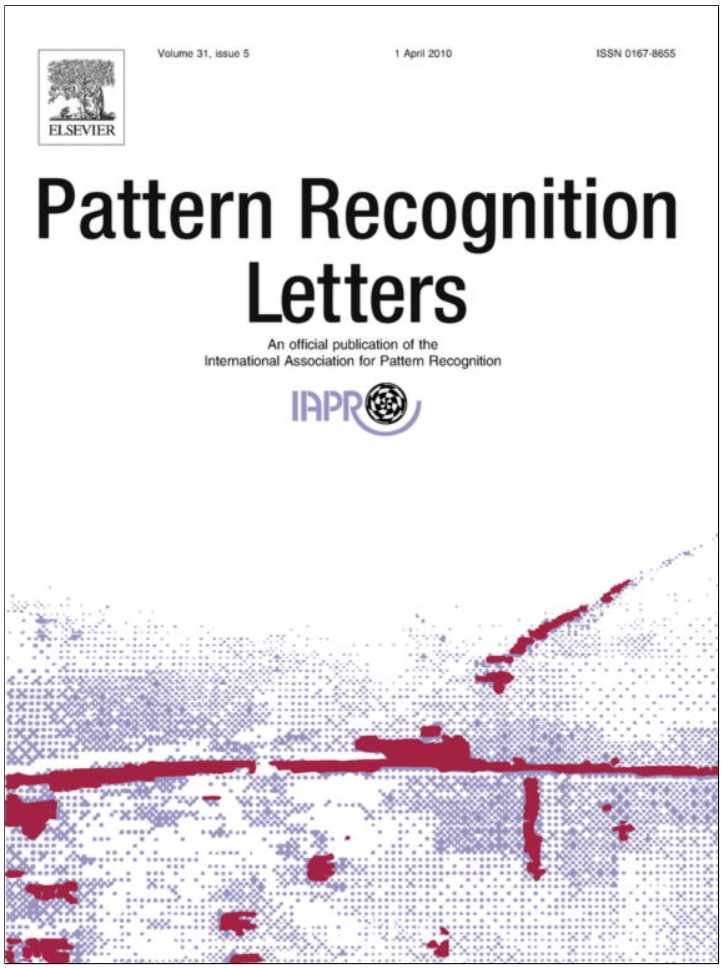

Pattern Recognition Letters 31 (2010) 430–443

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Unsupervised pattern recognition models for mixed feature-type symbolic data

Francisco de A.T. de Carvalho *, Renata M.C.R. de Souza

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

ARTICLE INFO

Article history:
Received 30 October 2008
Received in revised form 6 June 2009
Available online 13 November 2009
Communicated by R.C. Guido

Keywords:
Symbolic data analysis
Clustering analysis
Mixed feature-type symbolic data
Adaptive distances
Partition interpretation indices
Cluster interpretation indices

ABSTRACT

Unsupervised pattern recognition methods for mixed feature-type symbolic data based on dynamical clustering methodology with adaptive distances are presented. These distances change at each algorithm's iteration and can either be the same for all clusters or different from one cluster to another. Moreover, the methods need a previous pre-processing step in order to obtain a suitable homogenization of the mixed feature-type symbolic data into histogram-valued symbolic data. The presented dynamic clustering algorithms have then as input a set of vectors of histogram-valued symbolic data and they furnish a partition and a prototype to each cluster by optimizing an adequacy criterion based on suitable adaptive squared Euclidean distances. To show the usefulness of these methods, examples with synthetic symbolic data sets as well as applications with real symbolic data sets are considered. Moreover, various tools suitable for interpreting the partition and the clusters given by these algorithms are also presented.

1. Introduction

Clustering methods seek to organize a set of items (usually represented as a vector of quantitative values in a multidimensional space) into clusters such that items within a given cluster have a high degree of similarity, whereas items belonging to different clusters have a high degree of dissimilarity. These methods have been widely applied in various areas such as taxonomy, image processing, information retrieval, data mining, etc. and they may be divided into hierarchical and partitioning methods (Jain et al., 1999; Gordon, 1999): hierarchical methods yield complete hierarchy, i.e., a nested sequence of partitions of the input data, whereas partitioning methods seek to obtain a single partition of the input data in a fixed number of clusters, usually by optimizing an objective function.

The partitioning dynamical cluster algorithms (Diday, 1971; Diday and Simon, 1976) are iterative two-step relocation algorithms involving the construction of clusters at each iteration and the identification of a suitable representation or prototype (means, axes, probability laws, groups of elements, etc.) for each cluster by locally optimizing an adequacy criterion between the clusters and their corresponding representations. The adaptive dynamic clustering algorithm (Diday and Govaert, 1977) also optimize a criterion based on a measure of fitting between the clusters and their prototypes, but there are distances to compare clusters and their prototypes that change at each iteration. These distances are not determined once and for all, and moreover, they can be different from one cluster to another. The advantage of these adaptive distances is that the clustering algorithm is able to recognize clusters of different shapes and sizes.

In classical clustering analysis, the patterns to be grouped are usually represented as a vector of quantitative or qualitative measurements where each column represents a variable. Each particular pattern takes a single value for each variable. In practice, however, this model is too restrictive to represent complex data. In order to take into account variability and/or uncertainty inherent to the data, variables must assume sets (or ordered lists) of categories or intervals, possibly even with frequencies or weights. Symbolic Data Analysis (SDA), a domain in the area of knowledge discovery and data management related to multivariate analysis, pattern recognition and artificial intelligence, has provided suitable methods (clustering, factorial techniques, decision trees, etc.) for managing aggregated data described by multi-valued variables, where the cells of the data table contain sets (or ordered lists) of categories, intervals, or weight (probability) distributions (Bock and Diday, 2000; Billard and Diday, 2007; Diday and Noirhome-Fraiture, 2008).

In SDA the clustering methods for symbolic data differ in the type of the considered symbolic data, in their cluster structures and/or in the considered clustering criteria. With hierarchical methods, Gowda and Diday (1991) introduced an agglomerative approach that forms composite symbolic objects using a join operator whenever mutual pairs of symbolic objects are selected for agglomeration based on minimum dissimilarity. Ichino and

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail addresses: fatc@cin.ufpe.br (F.A.T. de Carvalho), rmcrs@cin.ufpe.br (R.M.C.R. de Souza).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.11.007

Yaguchi (1994) defined generalized Minkowski metrics for mixed feature variables and presents dendrograms obtained from the application of standard linkage methods for data sets containing numeric and symbolic feature values. Gowda and Ravi (1995a,b), respectively, presented divisive and agglomerative algorithms for symbolic data based on the combined usage of similarity and dissimilarity measures. These proximity (similarity or dissimilarity) measures are defined on the basis of the position, span and content of symbolic objects. Chavent (2000) proposed a divisive clustering method for symbolic data that simultaneously furnishes a hierarchy of the symbolic data set and a monothetic characterization of each cluster in the hierarchy. Gowda and Ravi (1999) presented a hierarchical clustering algorithm for symbolic objects based on the gravitational approach, which is inspired on the movement of particles in space due to their mutual gravitational attraction. Guru et al. (2004) and Guru and Kiranagi (2005) introduced agglomerative clustering algorithms based, respectively, on similarity and dissimilarity functions that are multi-valued and non-symmetric.

A number of authors have addressed the problem of non-hierarchical clustering for symbolic data. Diday and Brito (1989) used a transfer algorithm to partition a set of symbolic objects into clusters described by weight distribution vectors. Ralambondrainy (1995) extended the classical k-means clustering method in order to manage data characterized by numerical and categorical variables, and complemented this method with a characterization algorithm to provide a conceptual interpretation of the resulting clusters. Gordon et al. (2000) presented an iterative relocation algorithm to partition a set of symbolic objects into classes so as to minimize the sum of the description potentials of the classes. Verde et al. (2001) introduced a dynamic clustering algorithm for symbolic data considering context-dependent proximity functions, where the cluster representatives are weight distribution vectors. Bock (2003) has proposed several clustering algorithms for symbolic data described by interval variables, based on a clustering criterion and has thereby generalized similar approaches in classical data analysis.

Concerning partitional dynamic clustering algorithms for symbolic data, Chavent and Lechevallier (2002) proposed a dynamic clustering algorithm for interval data where the cluster representatives are defined by an optimality criterion based on a modified Hausdorff distance. Souza and De Carvalho (2004) proposed partitioning clustering methods for interval data based on city-block distances, also considering adaptive distances. De Carvalho et al. (2006a) proposed an algorithm using an adequacy criterion based on adaptive Hausdorff distances and De Carvalho et al. (2006b) presented dynamical clustering algorithms based on non-adaptive Euclidean distances for interval data. More recently, De Carvalho and Lechevallier, 2009 presented dynamic clustering algorithms based on single adaptive (city-block and Hausdorff) distances that change at each iteration, but are the same for all clusters. However, none of these former dynamic clustering models are able to manage mixed feature-type symbolic data.

In this paper, we introduce dynamic clustering methods for mixed feature-type symbolic data based on suitable adaptive squared Euclidean distances used for compare clusters and their respective prototypes that change at each iteration: adaptive distances for each cluster, which are different from one cluster to another, and single adaptive distances, which are the same for all clusters. To be able to manage mixed feature-type symbolic data, these methods assume a previous pre-processing step the aim of which is to obtain a suitable homogenization of mixed feature-type symbolic data into histogram-valued symbolic data.

This paper is organized as follows. Section 2 first describes mixed feature-type symbolic data. and then introduces dynamical clustering algorithm for mixed feature-type symbolic data which uses, respectively, a single adaptive squared Euclidean distance and an adaptive squared Euclidean distance for each class. Section 3 presents various tools for cluster interpretation according to these adaptive clustering models: indices for evaluating the quality of a partition, the homogeneity and eccentricity of the individual clusters and the role played by the different variables in the cluster formation process. To show the usefulness of these clustering algorithms and the merit of these cluster interpretation tools, experiments with simulated data in a framework of a Monte Carlo schema as well as applications with real symbolic interval-valued data sets are considered in Section 4. Finally, Section 5 gives the concluding remarks.

2. Dynamic clustering algorithms for mixed feature-type symbolic data

In classical data analysis, an individual is described by a row of a data matrix whose columns are single-valued variables, i.e., variables that assumes only one value from their domain. According to its domain, a variable may be quantitative (discrete or continuous) or qualitative (ordinal or nominal). However, this type of data is too restrictive to represent complex data which may comprehend, for instance, variability and/or uncertainty. It is why different types of symbolic variables and symbolic data have been introduced in symbolic data analysis.

A symbolic variable (Bock and Diday, 2000) X_j is set-valued if, given an item $i, X_j(i) = x_i^j \subseteq A_j$ where $A_j = \{t_1^j, \ldots, t_{H_j}^j\}$ is a set of categories. A symbolic variable X_j is ordered list-valued if, given an item $i, X_j(i) = x_i^j$, where x_i^j is a sub-list of a ordered list of categories $A_j = [t_1^j, \ldots, t_{H_j}^j]$. A symbolic variable X_j is an interval-valued variable when, given an item $i, X_j(i) = x_i^j = [a_i^j, b_i^j] \in [a, b]$, where $[a, b] \in \Im$ and $\Im$ is the set of closed intervals defined from $\Re$. Finally, a symbolic variable X_j is histogram-valued variable if, given an item $i, X_j(i) = x_i^j = (S^j(i), \mathbf{q}^j(i))$ where $\mathbf{q}^j(i) = (q_{i1}^j, \ldots, q_{iH_{ij}}^j)$ is a vector of weights defined in $S^j(i)$ such that a weight $q(m)$ corresponds to each category $m \in S^j(i)$. $S^j(i)$ is the support of the measure $\mathbf{q}^j(i)$.

Table 1 shows a mixed feature-type symbolic data table describing four cities. Symbolic variables X_1 (number of inhabitants in thousands), X_2 (spectrum of political parties: Democrats, Conservatives, Socialists, Nationalists), X_3 (Consulates: France, Italy, Spain, Great-Britain, Germany, Belgium) are, respectively, interval-valued, histogram-valued and set-valued.

Let a generic data table representing the values of p symbolic variables $X_1, \ldots, X_p$ on a set $\Omega = \{1, \ldots, n\}$ of n objects each one represented as a vector of mixed feature-type symbolic data $\mathbf{x}_i = (x_i^1, \ldots, x_i^p)$ $(i = 1, \ldots, n)$. This means that $x_i^j = X_j(i)$ can be a set or a (ordered) list of categories, an interval or a weight distribution according to the type of the corresponding symbolic variable. Table 2 shows a mixed feature-type symbolic data table where X_1 is an interval-valued variable, X_j is a set-valued variable and X_p is a histogram-valued variable.

The standard dynamical clustering algorithm (Diday and Simon, 1976) aims to provide a partition $P = (C_1, \ldots, C_K)$ of Ω in a fixed number K of clusters and their corresponding prototypes $L = (L_1, \ldots, L_K)$ by locally minimizing a criterion W that evaluates the fit between the clusters and their corresponding representatives.

Table 1
Mixed feature-type symbolic data table describing four cities.

City	X_1	X_2	X_3
1	[70,100]	$((D, C, S, N), (0.4, 0.3, 0.2, 0.1))$	$\{F, I\}$
2	[50,70]	$((D, C, S, N), (0.3, 0.3, 0.3, 0.1))$	$\{S, G\}$
3	[20,40]	$((D, C, S, N), (0.2, 0.2, 0.2, 0.4))$	$\{GB, G\}$
4	[60,100]	$((D, C, S, N), (0.1, 0.3, 0.4, 0.2))$	$\{B, GB\}$

Table 2
Mixed feature-type symbolic data table.

Object	X_1	$\cdots$	X_j	$\cdots$	X_p
1	$[a_1^j, b_1^j]$	$\cdots$	$\{t_{11}^j, \ldots, t_{1H_{1j}}^j\}$	$\cdots$	$(S^p(1), \mathbf{q}^p(1) = (q_{11}^p, \ldots, q_{1H_p}^p))$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
i	$[a_i^j, b_i^j]$	$\cdots$	$\{t_{i1}^j, \ldots, t_{iH_{ij}}^j\}$	$\cdots$	$(S^p(i), \mathbf{q}^p(i) = (q_{i1}^p, \ldots, q_{iH_p}^p))$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
n	$[a_n^j, b_n^j]$	$\cdots$	$\{t_{n1}^j, \ldots, t_{nH_{nj}}^j\}$	$\cdots$	$(S^p(n), \mathbf{q}^p(n) = (q_{n1}^p, \ldots, q_{nH_p}^p))$

This section presents two dynamic clustering algorithms based on suitable adaptive Euclidean distances in order to cluster mixed feature-type symbolic data: the first one for which the adaptive distance used to compare clusters and prototypes is unique and changes at each iteration and last one for which this adaptive distance is different from one cluster to another and changes at each iteration.

In order to be able to manage ordered and non-ordered mixed feature-type symbolic data, these methods assume a previous pre-processing step that aims to obtain a suitable homogenization of mixed feature-type symbolic data into histogram-valued symbolic data.

2.1. Data homogenization pre-processing step

The data homogenization is accomplished according to the type of symbolic variable: set-valued, ordered list-valued variables and interval-valued variable.

2.1.1. Set-valued and list-valued variables

If X_j is a set-valued variable, its transformation into a symbolic histogram-valued variable $\widetilde{X}_j$ is accomplished in the following way: $\widetilde{X}_j(i) = \tilde{x}_i^j = (A_j, \mathbf{q}^j(i))$, where $A_j = \{t_1^j, \ldots, t_{H_j}^j\}$ is the domain of variable X_j which is the support of the vector of weights $\mathbf{q}^j(i) = (q_1^j(i), \ldots, q_{H_j}^j(i))$. The weight $q_h^j(i)\,(h = 1, \ldots, H_j)$ of the category $t_h^j \in A_j$ is defined as (De Carvalho, 1995):

$$q_h^j(i) = \begin{cases} \frac{1}{c(x_i^j)}, & \text{if } t_h^j \in x_i^j, \\ 0, & \text{if } t_h^j \notin x_i^j, \end{cases} \tag{1}$$

where $c(A)$ is the cardinality of a finite set of categories A.

If X_j is an ordered list-valued variable, its transformation into a symbolic histogram-valued variable $\widetilde{X}_j$ is accomplished in the following way: $\widetilde{X}_j(i) = \tilde{x}_i^j = (A_j, \mathbf{Q}^j(i))$, where $A_j = [t_1^j, \ldots, t_{H_j}^j]$ is the domain of variable X_j which is the support of the vector of cumulative weights $\mathbf{Q}^j(i) = (Q_1^j(i), \ldots, Q_{H_j}^j(i))$. The cumulative weight $Q_h^j(i)\,(h = 1, \ldots, H_j)$ of the category t_h^j from the ordered list A_j is defined as (De Carvalho, 1995):

$$Q_h^j(i) = \sum_{r=1}^{h} q_r^j(i), \quad \text{where } q_r^j(i) = \begin{cases} \frac{1}{l(x_i^j)}, & \text{if category } t_r^j \text{ is in the sub-list } x_i^j, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $l(A)$ is the length of an ordered list of categories A.

It can be shown (De Carvalho, 1995) that $0 \leqslant q_h^j(i) \leqslant 1\,(h = 1, \ldots, H_j)$ and $\sum_{h=1}^{H_j} q_h^j(i) = 1$. Moreover, $q_1^j(i) = Q_1^j(i)$ and $q_h^j(i) = Q_h^j(i) - Q_{h-1}^j(i)\,(h = 2, \ldots, H_j)$.

2.1.2. Interval-valued variables

In this case, the interval-valued variable X_j is transformed into a symbolic histogram-valued variable $\widetilde{X}_j$ in the following way: $\widetilde{X}_j(i) = \tilde{x}_i^j = (\widetilde{A}_j, \mathbf{Q}^j(i))$, where $\widetilde{A}_j = \{I_1^j, \ldots, I_{H_j}^j\}$ is a list of elementary intervals which is the support of the vector of cumulative weights $\mathbf{Q}^j(i) = (Q_1^j(i), \ldots, Q_{H_j}^j(i))$. The cumulative weight $Q_h^j(i)\,(h = 1, \ldots, H_j)$ of the elementary interval I_h^j is defined as (De Carvalho, 1995):

$$Q_h^j(i) = \sum_{r=1}^{h} q_r^j(i), \quad \text{where } q_r^j(i) = \frac{l(I_r^j \cap x_i^j)}{l(x_i^j)}, \tag{3}$$

where $l(I)$ is the length of a closed interval I.

It can be shown (De Carvalho, 1995) that also in this case $0 \leqslant q_h^j(i) \leqslant 1\,(h = 1, \ldots, H_j)$ and $\sum_{h=1}^{H_j} q_h^j(i) = 1$. Moreover, again $q_1^j(i) = Q_1^j(i)$ and $q_h^j(i) = Q_h^j(i) - Q_{h-1}^j(i)\,(h = 2, \ldots, H_j)$.

The boundaries of these elementary intervals $I_h^j\,(h = 1, \ldots, H_j)$ are obtained from the ordered boundaries of the $n+1$ intervals $\{x_1^j, \ldots, x_n^j, [a, b]\}$ and the number of elementary intervals (bin size of the weight distribution) H_j is at most $2n$. They have the following properties (De Carvalho, 1995; De Carvalho et al., 1999; Chavent et al., 2003):

(1) $\bigcup_{h=1}^{H_j} I_h^j = [a, b]$.
(2) $I_h^j \bigcap I_{h'}^j = \emptyset$ if $h \neq h\prime$.
(3) $\forall h \exists i \in \Omega$ such that $I_h^j \bigcap x_i^j \neq \emptyset$.
(4) $\forall i \exists S_i^j \subset \{1, \ldots, H_j\} : \bigcup_{h \in S_i^j} I_h^j = x_i^j$.

Table 3 shows a generic symbolic data table at the end of the homogenization pre-processing step.

2.1.3. Example

In order to illustrate this data homogenization pre-processing step, we consider here a symbolic data table which describes four countries (items), each country being described by a symbolic interval-valued variable X_1 and a symbolic set-valued variable X_2. In Table 4, symbolic variable X_1 is the minimum and the maximum of the gross national product (in millions) whereas symbolic variable X_2 indicates the main industries from the set $A_2 =$ {A = agriculture, C = chemistry, Co = commerce, E = engineering, En = energy, I = informatic}.

Concerning the symbolic variable X_1, the set of elementary intervals $\widetilde{A}_1 = \{I_1^1, \ldots, I_{H_1}^1\}$ are obtained as follows: at first, we consider the set of values formed by every bound (lower and upper) of all the intervals associated to the items. Then, such set of bounds is sorted in a growing way. Therefore, the set of elementary intervals is $\widetilde{A}_1 = \{I_1^1, I_2^1, I_3^1, I_4^1, I_5^1, I_6^1, I_7^1\}$, where $I_1^1 = [10, 25[$, $I_2^1 = [25, 30[$, $I_3^1 = [30, 35[$, $I_4^1 = [35, 90[$, $I_5^1 = [90, 125[$, $I_6^1 = [125, 130[$ and $I_7^1 = [130, 140]$.

Concerning symbolic variable X_2, $A_2 =$ {A = agriculture, C = chemistry, Co = commerce, E = engineering, En = energy, I = informatic}.

Table 3
Histogram-valued symbolic data table.

Object	X_1	$\cdots$	X_p
1	$(A^1(1), \mathbf{q}^1(1) = (q_{11}^1, \ldots, q_{1H_1}^1))$	$\cdots$	$(A^p(1), \mathbf{q}^p(1) = (q_{11}^p, \ldots, q_{1H_p}^p))$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
i	$(A^1(i), \mathbf{q}^1(i) = (q_{i1}^1, \ldots, q_{iH_1}^1))$	$\cdots$	$(A^p(i), \mathbf{q}^p(i) = (q_{i1}^p, \ldots, q_{iH_p}^p))$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
n	$(A^1(n), \mathbf{q}^1(n) = (q_{n1}^1, \ldots, q_{nH_1}^1))$	$\cdots$	$(A^p(n), \mathbf{q}^p(n) = (q_{n1}^p, \ldots, q_{nH_p}^p))$

Table 4
Countries described by symbolic variables.

Country	X_1	X_2
1	[10,30]	{A, Co}
2	[25,35]	{C, Co, E}
3	[90,130]	{A, C, E}
4	[125,140]	{A, C, Co, E}

In this way, each item (country) $i\,(i=1,2,3,4)$ is represented as a vector of histogram-valued symbolic data $\tilde{\mathbf{x}}_i=(\tilde{x}_i^1,\tilde{x}_i^2)$, where $\tilde{x}_i^1=(\widetilde{A}_1,\mathbf{Q}^1(i))$ and $\tilde{x}_i^2=(A_2,\mathbf{q}^2(i))$, with $\mathbf{q}^2(i)$ and $\mathbf{Q}^1(i)$ being obtained, respectively, as described in paragraphs 2.1.1 and 2.1.2.

Finally, Table 5 shows the new histogram-valued symbolic data table obtained after the application of the data homogenization pre-processing step to the original mixed feature-type symbolic data table:

2.2. Clustering adequacy criterion based on adaptive distances

After the pre-processing step, each object $i(i=1,\ldots,n)$ is represented by a vector of histogram-valued symbolic data $\tilde{\mathbf{x}}_i=(\tilde{x}_i^1,\ldots,\tilde{x}_i^p),\tilde{x}_i^j=(D_j,\mathbf{u}^j(i))$, where D_j (the domain of variable $\widetilde{X}_j$) is a set of categories if $\widetilde{X}_j$ is a histogram-valued or set-valued variable, D_j is an ordered list of categories if $\widetilde{X}_j$ is a list-valued variable and D_j is a list of elementary intervals if $\widetilde{X}_j$ is an interval-valued variable. Moreover, $\mathbf{u}^j(i)=(u_1^j(i),\ldots,u_{H_j}^j(i))$ is a vector of weights if D_j is a set of categories and $\mathbf{u}^j(i)$ is a vector of cumulative weights if D_j is an ordered list of categories or a list of elementary intervals.

We assume here that the prototype of cluster $C_k(k=1,\ldots,K)$ is also represented as a vector of histogram-valued symbolic data $\mathbf{g}_k=(g_k^1,\ldots,g_k^p),g_k^j=(D_j,\mathbf{v}^j(k))\,(j=1,\ldots,p)$, where $\mathbf{v}^j(k)=(v_1^j(k),\ldots,v_{H_j}^j(k))$ is a vector of weights if D_j is a set of categories and $\mathbf{v}^j(k)$ is a vector of cumulative weights if D_j is a list of categories or a list of elementary intervals. Note that for each variable $j\,(j=1,\ldots,p)$ the support D_j is the same for all individuals and prototypes.

According to the general schema of the dynamic clustering algorithm (Diday and Simon, 1976), this clustering method looks for the partition $P^*=\{C_1^*,\ldots,C_K^*\}$ of Ω into K clusters, the corresponding K prototypes $\mathbf{G}^*=(\mathbf{g}_1^*,\ldots,\mathbf{g}_K^*)$ representing the clusters in P^* and K squared adaptive Euclidean distances parameterized by K vectors of weights $\mathbf{D}^*=(\boldsymbol{\lambda}_1^*,\ldots,\boldsymbol{\lambda}_K^*)$ such that

$$W(\mathbf{G}^*,\mathbf{D}^*,P^*)=min\{W(\mathbf{G},\mathbf{D},P):\mathbf{G}\in\mathbb{L}^K,\mathbf{D}\in\boldsymbol{\Lambda}^K,P\in\mathbb{P}_K\},\tag{4}$$

where

- $\mathbb{P}_K$ is the set of all the possible partitions of Ω in K classes such that $C_k\in\mathbf{P}(\Omega)$ (the set of subsets of Ω) and $P\in\mathbb{P}_K$;
- $\mathbb{L}$ is the representation space of prototypes such that $\mathbf{g}_k\in\mathbb{L}\,(k=1,\ldots,K)$ and $\mathbf{G}\in\mathbb{L}^K=\mathbb{L}\times\ldots\times\mathbb{L}$. In this paper, $\mathbb{L}=(D_1\times[0,1]^{H_1})\times\ldots\times(D_p\times[0,1]^{H_p})$;
- $\boldsymbol{\Lambda}$ is the space of vectors of weights that parameterize the adaptive Euclidean distances such that $\boldsymbol{\lambda}_k\in\boldsymbol{\Lambda}(k=1,\ldots,K)$ and $\mathbf{D}\in\boldsymbol{\Lambda}^K=\boldsymbol{\Lambda}\times\ldots\times\boldsymbol{\Lambda}$. In this paper, $\boldsymbol{\Lambda}=\mathbb{R}^p$.

The adequacy criterion $W(\mathbf{G},\mathbf{D},P)$ is defined as:

$$W(\mathbf{G},\mathbf{D},P)=\sum_{k=1}^{K}\sum_{i\in C_k}d(\tilde{\mathbf{x}}_i,\mathbf{g}_k|\boldsymbol{\lambda}_k)\tag{5}$$

Squared adaptive Euclidean distances are considered depending on whether they are parameterized by a unique and same weight vector or by a different weight vector for each cluster. They are:

(a) Single squared adaptive Euclidean distances parameterized by the weight vector $\boldsymbol{\lambda}_k=\boldsymbol{\lambda}\,(k=1,\ldots,K)$, where $\boldsymbol{\lambda}=(\lambda^1,\ldots,\lambda^p)$, which changes at each iteration but it is the same for all clusters:

$$\begin{aligned}d(\tilde{\mathbf{x}}_i,\mathbf{g}_k|\boldsymbol{\lambda})&=\sum_{j=1}^{p}\lambda^j\phi^2(\mathbf{u}^j(i),\mathbf{v}^j(k))\\&=\sum_{j=1}^{p}\lambda^j\sum_{h=1}^{H_j}(u_h^j(i)-v_h^j(k))^2\end{aligned}\tag{6}$$

(b) Cluster squared adaptive Euclidean distances parameterized by the weight vectors $\boldsymbol{\lambda}_k=(\lambda_k^1,\ldots,\lambda_k^p)\,(k=1,\ldots,K)$, which changes at each iteration but is different from one cluster to another.

$$\begin{aligned}d(\tilde{\mathbf{x}}_i,\widetilde{\mathbf{g}}_k|\boldsymbol{\lambda}_k)&=\sum_{j=1}^{p}\lambda_k^j\phi^2(\mathbf{u}^j(i),\mathbf{v}^j(k))\\&=\sum_{j=1}^{p}\lambda_k^j\sum_{h=1}^{H_j}(u_h^j(i)-v_h^j(k))^2\end{aligned}\tag{7}$$

Single adaptive quadratic distances are parameterized by a weight vector in which the weights are estimated globally for all clusters at once while cluster adaptive quadratic distances are parameterized by a weight vector in which the weights are estimated locally for each cluster.

From a initial solution $v_0=(\mathbf{G}^0,\mathbf{D}^0,P^0)$, the dynamic clustering algorithm alternates three steps until convergence when the criterion W reaches a stationary value representing a local minimum.

2.2.1. Step 1: definition of the best prototypes

In the first step, the partition of Ω in K clusters $P=\{C_1,\ldots,C_K\}$ and the corresponding K vectors of weights $\mathbf{D}=(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_K)$ are fixed.

Proposition 2.1. *Whichever the distance function (Eqs. 6 and 7), the vector of prototypes $G=(\mathbf{g}_1,\ldots,\mathbf{g}_K)$, where $\mathbf{g}_k=(g_k^1,\ldots,g_k^p)$ $(k=1,\ldots,K)$ with $g_k^j=(D_j,\mathbf{v}^j(k))\,(j=1,\ldots,p)$, which minimizes the clustering criterion W, is such that the components $v_h^j(k)\,(h=1,\ldots,H_j)$ of the weight vector $\mathbf{v}^j(k)=(v_1^j(k),\ldots,v_{H_j}^j(k))$ are calculated according to:*

$$v_h^j(k)=\frac{1}{n_k}\sum_{i\in C_k}u_h^j(i)\tag{8}$$

where n_k is the cardinality of the class C_k.

Proof 1. As the criterion W is additive, the optimization problem becomes to find for $k=1,\ldots,K,j=1,\ldots,p$ and $h=1,\ldots,H_j$, the weight $v_h^j(k)$ minimizing $\sum_{i\in C_k}(u_h^j(i)-v_h^j(k))^2$. One can easily see that,

$$v_h^j(k)=\frac{1}{n_k}\sum_{i\in C_k}u_h^j(i).\qquad\square$$

2.2.2. Step 2: definition of the best distances

In the second step, the partition of Ω into K clusters $P=\{C_1,\ldots,C_K\}$ and the corresponding vector of prototypes $\mathbf{G}=(\mathbf{g}_1,\ldots,\mathbf{g}_K)$ are fixed.

Table 5
Countries described by two histogram-valued symbolic variables.

Country	$\widetilde{X}_1$	$\widetilde{X}_2$
1	$(A_1,\mathbf{Q}^1(1)=(0.75,1,1,1,1,1,1))$	$(A_2,\mathbf{q}^2(1)=(0.5,0,0.5,0,0,0))$
2	$(A_1,\mathbf{Q}^1(2)=(0,0.5,0.5,1,1,1,1))$	$(A_2,\mathbf{q}^2(2)=(0,0.33,0.33,0.33,0,0))$
3	$(A_1,\mathbf{Q}^1(3)=(0,0,0,0,0.88,1,1))$	$(A_2,\mathbf{q}^2(3)=(0.33,0.33,0,0.33,0,0))$
4	$(A_1,\mathbf{Q}^1(4)=(0,0,0,0,0,0.33,1))$	$(A_2,\mathbf{q}^2(4)=(0.25,0.25,0.25,0.25,0,0))$

Proposition 2.2. *The K vectors of weights* $\mathbf{D} = (\lambda_1, \ldots, \lambda_K)$*, which minimize the clustering criterion W are calculated according to the adaptive distance function used:*

(a) if the distance function is given by Eq. (6), the vectors of weights $\lambda_k = \boldsymbol{\lambda}(k = 1, \ldots, K)$*, where* $\boldsymbol{\lambda} = (\lambda^1, \ldots, \lambda^p)$*, which minimize the clustering criterion W under* $\lambda^j > 0$ *and* $\prod_{j=1}^{p} \lambda^j = \eta$*, where* $\eta \in \mathbb{R}$ *is a constant, have its weights* λ^j *calculated according to:*

$$\lambda^j = \frac{\left\{\eta \prod_{l=1}^{p} \left(\sum_{k=1}^{K} \left[\sum_{i \in C_k} \left(\sum_{h=1}^{H_l} (u_h^l(i) - v_h^l(k))^2\right)\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K} \left[\sum_{i \in C_k} \left(\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2\right)\right]}. \quad (9)$$

(b) if the distance function is given by Eq. (7), the vectors of weights $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^p)$ $(k = 1, \ldots, K)$*, which minimize the clustering criterion W under* $\lambda_k^j > 0$ *and* $\prod_{j=1}^{p} \lambda_k^j = \chi$*, where* $\chi \in \mathbb{R}$ *is a constant, have its weights* λ_k^j *calculated according to:*

$$\lambda_k^j = \frac{\left\{\chi \prod_{l=1}^{p} \left(\sum_{i \in C_k} \left(\sum_{h=1}^{H_j} (u_h^l(i) - v_h^l(k))^2\right)\right)\right\}^{\frac{1}{p}}}{\sum_{i \in C_k} \left(\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2\right)}. \quad (10)$$

Proof 2. If the adaptive distance in Eq. (6) is considered, as the partition of Ω into K clusters and the prototypes $\mathbf{g}_k$ $(k = 1, \ldots, K)$ are fixed, one can rewrite the criterion W as

$$W(\lambda^1, \ldots, \lambda^p) = \sum_{j=1}^{p} \lambda^j W^j \text{ in which } W^j = \sum_{k=1}^{K} \sum_{i \in C_k} \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2\right]$$

Let $g(\lambda^1, \ldots, \lambda^p) = \lambda^1 \times \ldots \times \lambda^p - \eta$. We want to determine the extremes of $W(\lambda^1, \ldots, \lambda^p)$ with the restriction $g(\lambda^1, \ldots, \lambda^p) = 0$. From the Lagrange multiplier method, and after some algebra, it follows that (for $j = 1, \ldots, p$)

$$\lambda^j = \frac{\left\{\eta \prod_{l=1}^{p} W^l\right\}^{\frac{1}{p}}}{W^j} = \frac{\left\{\eta \prod_{l=1}^{p} \left(\sum_{k=1}^{K} \left[\sum_{i \in C_k} \left(\sum_{h=1}^{H_l} (u_h^l(i) - v_h^l(k))^2\right)\right]\right)\right\}^{\frac{1}{p}}}{\sum_{k=1}^{K} \left[\sum_{i \in C_k} \left(\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2\right)\right]}.$$

As $W\left(\eta^{\frac{1}{p}}, \ldots, \eta^{\frac{1}{p}}\right) = \eta^{\frac{1}{p}} \sum_{j=1}^{p} W^j = \eta^{\frac{1}{p}}(W^1 + \cdots + W^p)$ and as it is well known that the arithmetic mean is greater than the geometric mean i.e., $\frac{1}{p}(W^1 + \cdots + W^p) > \{W^1, \ldots, W^p\}^{1/p}$ (the equality holds only if $W^1 = \cdots = W^p$), we conclude that this extreme is a minimum value.

If the adaptive distance in Eq. (7) is considered, the proof can be obtained in a similar way as presented above.

2.2.3. Step 3: definition of the best partition

In this step, the vector of prototypes $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_K)$ and the corresponding K vectors of weights $\mathbf{D} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K)$ are fixed. □

Proposition 2.3. *The partition* $P = \{C_1, \ldots, C_K\}$*, which minimize the criterion W, has clusters* $C_k (k = 1, \ldots, K)$ *updated according to the following allocation rule:*

$$C_k = \{i \in \Omega : d(\tilde{\mathbf{x}}_i, \mathbf{g}_k|\boldsymbol{\lambda}_k) < d(\tilde{\mathbf{x}}_i, \mathbf{g}_m|\boldsymbol{\lambda}_m) \text{ and when } d_k(\tilde{\mathbf{x}}_i, \mathbf{g}_k|\boldsymbol{\lambda}_k) = d_k(\tilde{\mathbf{x}}_i, \mathbf{g}_m|\boldsymbol{\lambda}_m) \text{ then } i \in C_k \text{ if } k < m \quad \forall m \neq k (m = 1, \ldots, K)\} \quad (11)$$

Proof 3. The proof of Proposition 2.3 is straightforward. □

2.2.4. Properties of the algorithm

According to Diday and Simon (1976), 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 $u_t = W(v_t) = W(\mathbf{G}^t, \mathbf{D}^t, P^t)$, $t = 0, 1, \ldots$.

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 W achieves a stationary value.

Proposition 2.4. *The series* $u_t = W(v_t)$ *decreases at each iteration and converges.*

Proof 4. Following Diday and Simon (1976), first we will show that the inequalities $(I), (II)$ and (III)

$$\underbrace{W(\mathbf{G}^t, \mathbf{D}^t, P^t)}_{u_t} \overset{(I)}{\geqslant} W(\mathbf{G}^{t+1}, \mathbf{D}^t, P^t) \overset{(II)}{\geqslant} W(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^t) \overset{(III)}{\geqslant} \underbrace{W(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^{t+1})}_{u_{t+1}}$$

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

The inequality (I) holds because

$$W(\mathbf{G}^t, \mathbf{D}^t, P^t) = \sum_{k=1}^{K} \sum_{i \in C_k^{(t)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t)}|\boldsymbol{\lambda}_k^{(t)}),$$

$$W(\mathbf{G}^{t+1}, \mathbf{D}^t, P^t) = \sum_{k=1}^{K} \sum_{i \in C_k^{(t)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t+1)}|\boldsymbol{\lambda}_k^{(t)}),$$

and according to Proposition (2.1)),

$$\mathbf{g}_k^{(t+1)} = \underbrace{argmin}_{\mathbf{g} \in \mathbb{L}} \sum_{i \in C_k^{(t)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}|\boldsymbol{\lambda}_k^{(t)}) \quad (k = 1, \ldots, K).$$

Moreover, inequality (II) also holds because

$$W(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^t) = \sum_{k=1}^{K} \sum_{i \in C_k^{(t)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t+1)}|\boldsymbol{\lambda}_k^{(t+1)})$$

and according Proposition (2.2),

$$\boldsymbol{\lambda}_k^{(t+1)} = \underbrace{argmin}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \sum_{i \in C_k^{(t)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t+1)}|\boldsymbol{\lambda}) \quad (k = 1, \ldots, K).$$

The inequality (III) also holds because

$$W(\mathbf{G}^{t+1}, \mathbf{D}^{t+1}, P^{t+1}) = \sum_{k=1}^{K} \sum_{i \in C_k^{(t+1)}} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t+1)}|\boldsymbol{\lambda}_k^{(t+1)})$$

and according Proposition (2.3),

$$C_k^{(t+1)} = \underbrace{argmin}_{C \in \mathbf{P}(\Omega)} \sum_{i \in C} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k^{(t+1)}|\boldsymbol{\lambda}_k^{(t+1)}) \quad (k = 1, \ldots, K).$$

Finally, because the series u_t decreases and it is bounded $(W(v_t) \geqslant 0)$, it converges. □

Proposition 2.5. *The series* $v_t = (\mathbf{G}^t, \mathbf{D}^t, P^t)$ *converges.* □

Proof 5. Assume that the stationarity of the series u_t is achieved in the iteration $t = T$. Then, we have that $u_T = u_{T+1}$ and then $W(v_T) = W(v_{T+1})$.

From $W(v_T) = W(v_{T+1})$, we have $W(\mathbf{G}^T, \mathbf{D}^T, P^T) = W(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^{T+1})$ and this equality, according to Proposition 2.4, can be rewritten as equalities $(I), (II)$ and (III):

$$W(\mathbf{G}^T, \mathbf{D}^T, P^T) \overset{I}{=} W(\mathbf{G}^{T+1}, \mathbf{D}^T, P^T) \overset{II}{=} W(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^T) \overset{III}{=} W(\mathbf{G}^{T+1}, \mathbf{D}^{T+1}, P^{T+1})$$

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

third equality (*III*), we have that $P^T = P^{T+1}$ because P is unique minimizing W when the vector of prototypes $\mathbf{G}^{T+1}$ and the vector vectors of weights $\mathbf{D}^T$ are fixed.

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

Concerning the time complexity of the algorithm, it can be analyzed considering the complexity of each single step. The pre-processing step computes modal descriptions assuming that each individual is described by interval-valued variables (worst case). Initially, a list of elementary intervals for each variable is obtained once and stored and it costs $O(nlogn)$ per variable. The vectors of cumulative weights for individuals are calculated leading to a cost $O(n^2p)$. Thus, the overall time complexity of this step is $O(n^2p)$. The initialization step is computed once and it costs $O(Kn^2p)$. The clustering algorithm performs interactively three steps until the convergence and each one also costs $O(Kn^2p)$. Then, the complexity of the algorithm is $O(Kn^2p)$. □

2.2.5. Algorithm schema

The dynamic clustering algorithms for mixed feature-type symbolic data has the following steps:

1. **Pre-processing step: data homogenization**
   for all objects $i = 1, \ldots, n$ and for all variables $j = 1, \ldots, p$
   compute $\tilde{x}_i^j = (D_j, \mathbf{u}^j(i))$ using:
   Eq. (1) for the set-valued variables,
   Eq. (2) for ordered list-valued variables,
   Eq. (3) for the interval-valued variables.
2. **Initialization step:**
   Randomly choose a partition $P^{(0)} = (C_1^{(0)} \ldots, C_K^{(0)})$ of Ω or randomly choose K distinct objects $\mathbf{g}_1^{(0)}, \ldots, \mathbf{g}_K^{(0)}$ belonging to Ω and assign each objects i to the closest prototype $\mathbf{g}_{k*}^{(0)}$, where $k* = \arg\min_{k=1,\ldots,K} \{\sum_{j=1}^{p} \sum_{h=1}^{H_j} ((u_h^j(i))^{(0)} - (v_h^j(k))^{(0)})^2\}$.
3. $t = 1$.
4. **Step 1: definition of the best prototypes:**
   The partition $P^{(t)}$ and the vector of vectors of weights $\mathbf{D}^{(t)}$ are fixed.For $k = 1, \ldots, K$, compute the vector of histogram-valued symbolic data $\mathbf{g}_k^{(t)} = ((g_k^1)^{(t)}, \ldots, (g_k^p)^{(t)}), (g_k^j)^{(t)} = (D_j, (\mathbf{v}^j(k))^{(t)})$ $(j = 1, \ldots, p)$, where $(\mathbf{v}^j(k))^{(t)} = ((v_1^j(k))^{(t)}, \ldots, (v_{H_j}^j(k))^{(t)})$ and $(v_h^j(k))^{(t)} (h = 1, \ldots, H_j)$ is given by Eq. (8).
5. **Step 2: definition of the best vector of weights:**
   The partition $P^{(t)}$ and the vector of prototypes $\mathbf{G}^{(t)}$ are fixed.-For the method based on single squared adaptive Euclidean distances, compute the components of $\boldsymbol{\lambda}_k^{(t)} = \boldsymbol{\lambda}^{(t)} = ((\lambda^1)^{(t)}, \ldots, (\lambda^p)^{(t)})$ according to Eq. (9). For the method based on squared adaptive Euclidean distances for each cluster, compute the components of $\boldsymbol{\lambda}_k^{(t)} = ((\lambda_k^1)^{(t)}, \ldots, (\lambda_k^p)^{(t)})$ according to Eq. (10).
6. **Step 3: definition of the best partition:**
   The vector of prototypes $\mathbf{G}^{(t)}$ and the vector of vectors of weights $\mathbf{D}^{(t)}$ are fixed.
   $test \leftarrow 0$
   for $i = 1, \ldots, n$ do
   find the cluster $C_m^{(t)}$ to which i belongs
   find the index k such that: $k* = \arg\min_{1 \leqslant l \leqslant K} d(\tilde{\mathbf{x}}_i, \mathbf{g}_l^{(t)} | \boldsymbol{\lambda}_k^{(t)})$
   if $k \neq m$
   $test \leftarrow 1$
   $C_k^{(t)} \leftarrow C_k^{(t)} \cup \{i\}$ and $C_m^{(t)} \leftarrow C_m^{(t)} \setminus \{i\}$
   $t = t + 1$
7. **Stopping criterion**
   If $test = 0$ then STOP, else go to 3 (Step 1).

3. Cluster interpretation

Partition and cluster interpretation tools allows the user to evaluate the overall data heterogeneity, the intra-cluster and the between-cluster heterogeneity, the contribution of each variable to the cluster formation, etc. For usual quantitative data partitioned by the standard dynamic clustering algorithm, Celeux et al. (1989) have introduced a family of indices for cluster and partition interpretation that are based on dispersion measures. Later, De Carvalho et al. (2006b) adapted these indices for the case of interval-valued data partitioned by a dynamic clustering algorithm based on a (non-adaptive) squared Euclidean distance.

Here, we adapt these indices for the interpretation of the partition and the corresponding clusters of histogram-valued data obtained after the pre-processing step and partitioned by the dynamic clustering algorithms that have the adequacy criterion based on the adaptive distances presented in Section 2.2.

Let P be a partition of Ω into K clusters $C_1, \ldots, C_K$ of cardinality n_k which was obtained from one of the adaptive dynamic clustering algorithms presented in Section 2. Let $\mathbf{g}_k = (g_k^1, \ldots, g_k^p), g_k^j = (D_j, \mathbf{v}^j(k))\,(j = 1, \ldots, p)$, be the symbolic description of the representative of cluster C_k, where $\mathbf{v}^j(k) = (v_1^j(k), \ldots, v_{H_j}^j(k))$ is a vector of weights if D_j is a set of categories and $\mathbf{v}^j(k)$ is a vector of cumulative weights if D_j is an ordered list of categories or a list of elementary intervals with $v_h^j(k)\,(h = 1, \ldots, H_j)$ calculated according to Eq. 8.

Moreover, the vector of histograms $\mathbf{g} = (g^1, \ldots, g^p), g^j = (D_j, \mathbf{v}^j)$ $(j = 1, \ldots, p)$, where $\mathbf{v}^j = (v_1^j, \ldots, v_{H_j}^j)$ is a vector of weights if D_j is a set of categories and $\mathbf{v}^j$ is a vector of cumulative weights if D_j is an ordered list of categories or a list of elementary intervals, is the overall representative of the objects belonging to Ω.

3.1. Measures based on the sum of squares

In the following, three sums of squares (SSQ) are defined for this partition: the overall SSQ, the SSQ within clusters and SSQ between clusters. These measures are the basis for defining the interpretation tools in Section 3.2.

3.1.1. Overall sum of squares

The overall heterogeneity of all n objects belonging to Ω is measured by the overall sum of squares according to the distance function used:

$$T = \sum_{i=1}^{n} d(\tilde{\mathbf{x}}_i, \mathbf{g} | \boldsymbol{\lambda}_k) = \sum_{k=1}^{K} \sum_{i \in C_k} d(\tilde{\mathbf{x}}_i, \mathbf{g} | \boldsymbol{\lambda}_k). \tag{12}$$

Proposition 3.1. *Whichever the distance functions (Eqs. 6 and 7), the overall prototype $\mathbf{g} = (g^1, \ldots, g^p), g^j = (D_j, \mathbf{v}^j)\,(j = 1, \ldots, p)$, that minimizes the overall dispersion T has the components $v_h^j\,(h = 1, \ldots, H_j)$ of the weight vector $\mathbf{v}^j = (v_1^j, \ldots, v_{H_j}^j)$ calculated according to:*

$$v_h^j = \frac{1}{n} \sum_{i=1}^{n} u_h^j(i). \tag{13}$$

Proof 6. The proof is similar to that presented in Proposition 2.1.

The overall SSQ T decomposes as:

(a) $T = \sum_{k=1}^{K} T_k$ with $T_k = \sum_{i \in C_k} d(\tilde{\mathbf{x}}_i, \mathbf{g} | \boldsymbol{\lambda}_k)$ (whichever the distance functions (6) and (7));

(b) $T = \sum_{j=1}^{p} T_j$ with $T_j = \sum_{k=1}^{K} \sum_{i \in C_k} \lambda^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j)^2\right]$ (distance given by Eq. (6) and $T_j = \sum_{k=1}^{K} \sum_{i \in C_k} \lambda_k^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j)^2\right]$ (distance given by Eq. (7));

(c) $T = \sum_{k=1}^{K} \left(\sum_{j=1}^{p} T_{kj}\right)$ with $T_{kj} = \sum_{i \in C_k} \lambda^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j)^2\right]$ (distance given by Eq. (6) and $T_{kj} = \sum_{i \in C_k} \lambda_k^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j)^2\right]$ (distance given by Eq. (7)). □

3.1.2. Within-class sum of squares: W

Similarly, we may consider the heterogeneity within the clusters C_k and measure it by the within-cluster SSQ

$$W = \sum_{k=1}^{K} \sum_{i \in C_k} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k | \boldsymbol{\lambda}_k). \tag{14}$$

The overall within-cluster W decompose as

(a) $W = \sum_{k=1}^{K} W_k$ with $W_k = \sum_{i \in C_k} d(\tilde{\mathbf{x}}_i, \mathbf{g}_k | \boldsymbol{\lambda}_k)$ (whichever the distance functions (6) and (7));
(b) $W = \sum_{j=1}^{p} W_j$ with $W_j = \sum_{k=1}^{K} \sum_{i \in C_k} \lambda^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2 \right]$ (distance given by Eq. (6)) and $W_j = \sum_{k=1}^{K} \sum_{i \in C_k} \lambda_k^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2 \right]$ (distance given by Eq. (7));
(c) $W = \sum_{k=1}^{K} \left(\sum_{j=1}^{p} W_{kj} \right)$ with $W_{kj} = \sum_{i \in C_k} \lambda^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2 \right]$ (distance given by Eq. (6)) and $W_{kj} = \sum_{i \in C_k} \lambda_k^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2 \right]$ distance given by Eq. (7)).

3.1.3. Between-cluster sum of squares: B

The between-cluster SSQ is given by

$$B = \sum_{k=1}^{K} n_k d(\mathbf{g}_k, \mathbf{g} | \boldsymbol{\lambda}_k). \tag{15}$$

It measures the dispersion of the cluster representatives, i.e., the distinctness of all clusters. B is decomposed as:

(a) $B = \sum_{k=1}^{K} B_k$ with $B_k = n_k d(\mathbf{g}_k, \mathbf{g} | \boldsymbol{\lambda}_k)$ (whichever the distance functions (6) and (7));
(b) $B = \sum_{j=1}^{p} B_j$ with $B_j = \sum_{k=1}^{K} n_k \lambda^j \left[\sum_{h=1}^{H_j} (v_h^j(k) - v_h^j)^2 \right]$ (distance given by Eq. (6)) and $B_j = \sum_{k=1}^{K} n_k \lambda_k^j \left[\sum_{h=1}^{H_j} (v_h^j(k) - v_h^j)^2 \right]$ (distance given by Eq. (7));
(c) $B = \sum_{j=1}^{p} B_{kj}$ with $B_{kj} = n_k \lambda^j \left[\sum_{h=1}^{H_j} (v_h^j(k) - v_h^j)^2 \right]$ (distance given by Eq. (6)) and $B_{kj} = n_k \lambda_k^j \left[\sum_{h=1}^{H_j} (v_h^j(k) - v_h^j)^2 \right]$ (distance given by Eq. (7)).

Proposition 3.2. *Whichever the distance functions (Eqs. 6 and 7), the following relations hold:*

$$\begin{aligned} &T = W + B \\ &T_k = W_k + B_k (k = 1, \ldots, K) \\ &T_j = W_j + B_j (j = 1, \ldots, p) \\ &T_{kj} = W_{kj} + B_{kj} (k = 1, \ldots, K; j = 1, \ldots, p) \end{aligned} \tag{16}$$

Proof 7. Here, we will show that $T = W + B$ if the distance in Eq. (6) is used. The other expressions can be easily obtained in a similar way.

Let,

$$\begin{aligned} (u_h^j(i) - v_h^j)^2 &= [(u_h^j(i) - v_h^j(k)) + (v_h^j(k) - v_h^j)]^2 \\ &= [(u_h^j(i) - v_h^j(k))^2 + (v_h^j(k) - v_h^j)^2 + 2(u_h^j(i) - v_h^j(k)) \\ &\quad \times (v_h^j(k) - v_h^j)]. \end{aligned}$$

Then, from $T = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \lambda^j \left[\sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2 \right]$, we have

$$\begin{aligned} T = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \lambda^j \sum_{h=1}^{H_j} [&(u_h^j(i) - v_h^j(k))^2 + (v_h^j(k) - v_h^j)^2 \\ &+ 2(u_h^j(i) - v_h^j(k))(v_h^j(k) - v_h^j)] \end{aligned}$$

So,

$$T = W + B + 2 \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \lambda^j \sum_{h=1}^{H_j} [(u_h^j(i) - v_h^j(k))(v_h^j(k) - v_h^j)]$$

We have that,

$$((u_h^j(i) - v_h^j(k))(v_h^j(k) - v_h^j) = v_h^j(k)(u_h^j(i) - v_h^j(k)) - v_h^j(u_h^j(i) - v_h^j(k)).$$

Then,

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \lambda^j \sum_{h=1}^{H_j} [(u_h^j(i) - v_h^j(k))(v_h^j(k) - v_h^j)] \\ &= \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \lambda^j \sum_{h=1}^{H_j} [v_h^j(k)(u_h^j(i) - v_h^j(k)) - v_h^j(u_h^j(i) - v_h^j(k))] \\ &= \sum_{j=1}^{p} \sum_{k=1}^{K} \sum_{h=1}^{H_j} \left\{ [v_h^j(k) \left[\sum_{i \in C_k} (u_h^j(i) - v_h^j(k)) \right] - v_h^j \left[\sum_{i \in C_k} (u_h^j(i) - v_h^j(k)) \right] \right\} \end{aligned}$$

As $\left[\sum_{i \in C_k} (u_h^j(i) - v_h^j(k)) \right] = 0$, it follows that $\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \sum_{h=1}^{H_j} [(u_h^j(i) - v_h^j(k))(v_h^j(k) - v_h^j)] = 0$ and then $T = W + B$. □

3.2. Interpretation indices

Celeux et al. (1989) introduced a family of indices based on the notion of the sum of squares for interpreting the quality of the partition and the quality of the clusters of the partition of usual quantitative data obtained by the standard dynamic clustering algorithm. De Carvalho et al. (2006b) adapted these indices to the case of the partitioning clustering method for symbolic interval data based on a suitable non-adaptive squared Euclidean distance. Here, we adapt these indices also for the case of a partition of histogram-valued symbolic data given by the dynamic clustering algorithms based on the adaptive distances presented in Section 2. All these indices range from 0 to 1.

3.2.1. Partition interpretation indices

Interpreting the overall quality of a partition after having applied a clustering algorithm to the data is an important problem in clustering analysis.

3.2.1.1. Overall heterogeneity index. The proportion of the overall SSQ explained by the partition P is defined as

$$R = \frac{B}{T} = \frac{B}{B + W} = 1 - \frac{W}{T} \tag{17}$$

Minimizing the clustering criterion W is equivalent to maximizing R. A greater value of R leads to more homogeneous clusters and a better representation of the elements of a cluster C_k by its prototype $\mathbf{g}_k (k = 1, \ldots, K)$.

3.2.1.2. Overall heterogeneity indices with respect to single variables. The proportion of the overall dispersion without clustering (T_j) concerning the jth variable that corresponds to the dispersion of the partition after clustering concerning the jth variable (B_j), each cluster represented by its prototype, is defined as

$$COR(j) = \frac{B_j}{T_j} = \frac{B_j}{B_j + W_j}. \tag{18}$$

By comparing the value of $COR(j)$ with the value of the general index R, which measures the average discriminant power of all variables, may be evaluated if the discriminant power of the jth variable is above or below the average.

The relative contribution of the jth variable to the between-cluster sum of squares B is given by

$$CTR(j) = \frac{B_j}{B}. \quad (19)$$

Note that $\sum_{j=1}^{p} CTR(j) = 1$. A high value of $CTR(j)$ indicates that the jth variable provides an important contribution to the separation of the prototypes of the clusters. An interesting case arises when $COR(j)$ has a low value and $CTR(j)$ is large: this means that the jth variable has a low discriminant power, although it makes an important contribution to the sum of squares (Celeux et al., 1989).

3.2.2. Cluster interpretation indices

Another important problem in clustering analysis is evaluating the homogeneity and eccentricity of the individual clusters of a partition after having applied a clustering algorithm to the data.

3.2.2.1. Cluster heterogeneity indices. The proportion of the overall sum of squares in cluster C_k is given by

$$T(k) = \frac{T_k}{T}. \quad (20)$$

The contribution of a cluster C_k to the between-cluster sum of squares is measured by the ratio

$$B(k) = \frac{B_k}{B}. \quad (21)$$

A high value of $B(k)$ indicates that cluster C_k is quite distant from the global center in comparison to the totality of all clusters.

The contribution of cluster C_k to the within-cluster sum of squares is given by

$$W(k) = \frac{W_k}{W}. \quad (22)$$

A relatively large value of $W(k)$ indicates that cluster C_k is relatively heterogeneous in comparison with the other clusters.

Note that $\sum_{k=1}^{K} T(k) = \sum_{k=1}^{K} B(k) = \sum_{k=1}^{K} W(k) = 1$.

3.2.2.2. Cluster heterogeneity indices with respect to single variables. The proportion of the discriminant power of the jth variable with respect to cluster C_k is given by

$$COR(j,k) = \frac{B_{kj}}{T_j}. \quad (23)$$

Note that $\sum_{k=1}^{K} COR(j,k) = COR(j)$. A high value of $COR(j,k)$ shows that the jth variable has a relatively homogeneous behaviour within the cluster C_k.

The contribution of the jth variable to the heterogeneity in cluster C_k is given by

$$CTR(j,k) = \frac{B_{kj}}{B_k}. \quad (24)$$

Finally, we may consider the relative contribution of the jth variable and of the cluster C_k to the between-cluster sum of squares given by

$$CE(j,k) = \frac{B_{kj}}{B}. \quad (25)$$

If $CE(j,k)$ is close to 1, the jth variable has a large contribution to the eccentricity of the cluster C_k.

4. Experimental evaluation

To show the usefulness of the adaptive methods presented in this paper, two synthetic interval data sets with class overlapping and classes of different shapes and sizes have been drawn. For these data sets, the evaluation was performed in the framework of a Monte Carlo experience with 100 replications for each interval data set. Concerning real symbolic data, an application with two data sets is also considered.

Our aim is to achieve an evaluation of the dynamic clustering algorithm for mixed feature-type symbolic data considering different adaptive squared Euclidean distances between vectors of histogram-valued data: single adaptive squared Euclidean distances and adaptive squared Euclidean distance for each class. As in Diday and Govaert, 1977, here the vectors of weights of the single and cluster squared parameterized Euclidean distances are computed under the restrictions $\prod_{j=1}^{p} \lambda^j = \eta = 1$ and $\prod_{j=1}^{p} \lambda_k^j = \chi = 1$, respectively.

The dynamic clustering algorithm for mixed feature-type symbolic data considering a non-adaptive squared Euclidean distance between vectors of histogram-valued data is also evaluated in order to compare this distance with the adaptive distances introduced in this paper. The non-adaptive squared Euclidean distance between vectors of non-cumulative and cumulative weights is given by

$$d(\tilde{\mathbf{x}}_i, \mathbf{g}_k) = \sum_{j=1}^{p} \phi^2(\mathbf{u}^j(i), \mathbf{v}^j(k)) = \sum_{j=1}^{p} \sum_{h=1}^{H_j} (u_h^j(i) - v_h^j(k))^2. \quad (26)$$

Moreover, the dynamic clustering method for interval data based on L_2 distance presented in De Carvalho et al. (2006b) is also considered in order to provide comparison results with the adaptive methods introduced in this paper.

To measure the quality of the results furnished by the dynamic clustering algorithm considering different distances, the overall error rate of classification (OERC) and an external validity index are used. The OERC is obtained by computing the confusion matrix. The idea of external validity is simply to compare the a priori partition with the partition obtained from the clustering algorithm. In this paper, we use the corrected Rand (CR) index defined in Hubert and Arabie, 1985 for comparing two partitions, the definition of which is as follows. Let $U = \{u_1, \ldots, u_i, \ldots, u_R\}$ and $V = \{v_1, \ldots, v_j, \ldots, v_C\}$ be two partitions of the same data set having respectively R and C clusters. The corrected Rand index is:

$$CR = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2} - \binom{n}{2}^{-1}\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{j}}{2}}{\frac{1}{2}\left[\sum_{i=1}^{R}\binom{n_{i.}}{2} + \sum_{j=1}^{C}\binom{n_{j}}{2}\right] - \binom{n}{2}^{-1}\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{j}}{2}} \quad (27)$$

where $\binom{n}{2} = \frac{n(n-1)}{2}$ and n_{ij} represents the number of objects that are in clusters u_i and v_i; $n_{i.}$ indicates the number of objects in cluster u_i; $n_{.j}$ indicates the number of objects in cluster v_j; and n is the total number of objects in the data set. CR takes its values from the interval $[-1,1]$, where the value 1 indicates perfect agreement between partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance.

4.1. Synthetic interval-valued data sets

First, we simulated two classical quantitative data sets in $\Re^2$. Each synthetic data set was created having classes with different sizes and shapes. The synthetic data sets have 300 points each, divided into three classes of unequal sizes: one class of size 150, one class with 50 and one with 100. Each class in these data were drawn according to a bi-variate normal with dependent components. The mean vector and the covariance matrix of the bi-variate normal distributions are noted:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

In order to build interval-valued data sets from these quantitative data sets, each point (x_1, x_2) of these data sets is considered as the 'seed' of a rectangle. Each rectangle is therefore a vector of two intervals defined by:

$$(v1 = [x_1 - \gamma_1/2, x_1 + \gamma_1/2], v2 = [x_2 - \gamma_2/2, x_2 + \gamma_2/2]).$$

The parameters γ_1 and γ_2 are the width and the height of the rectangle. They are drawn randomly within a given range of values. For example, the width and the height of all the rectangles can be drawn randomly within the interval [1,10] (see Figs. 1 and 2).

Data Configuration 1 (Fig. 1) showing class covariance matrices almost identical were drawn according to the following parameters:

(a) Class 1: $\mu_1 = 28$, $\mu_2 = 23$, $\sigma_1^2 = 100$, $\sigma_2^2 = 16$ and $\rho_{12} = 0.8$;
(b) Class 2: $\mu_1 = 50$, $\mu_2 = 15$, $\sigma_1^2 = 100$, $\sigma_2^2 = 16$ and $\rho_{12} = 0.6$;
(c) Class 3: $\mu_1 = 57$, $\mu_2 = 48$, $\sigma_1^2 = 100$, $\sigma_2^2 = 16$ and $\rho_{12} = 0.9$;

Data Configuration 2 (Fig. 2) showing different class covariance matrices were drawn according to the following parameters:

(a) Class 1: $\mu_1 = 28$, $\mu_2 = 23$, $\sigma_1^2 = 144$, $\sigma_2^2 = 16$ and $\rho_{12} = 0.8$;
(b) Class 2: $\mu_1 = 50$, $\mu_2 = 15$, $\sigma_1^2 = 49$, $\sigma_2^2 = 81$ and $\rho_{12} = 0.6$;
(c) Class 3: $\mu_1 = 60$, $\mu_2 = 65$, $\sigma_1^2 = 16$, $\sigma_2^2 = 144$ and $\rho_{12} = 0.9$;

4.1.1. The Monte Carlo experiment results and comparison

In the framework of a Monte Carlo experiment, 100 replications of the previous process were carried out for parameters γ_1 and γ_2 drawn randomly 100 times from each of the following intervals: [1,10], [1,20], [1,30] and [1,40]. This process has also been repeated for interval data sets 1 and 2. The clustering algorithm have been performed on these data sets. The 3-cluster partition obtained with these clustering methods were compared with the 3-class partition known a priori based on the corrected Rand index CR

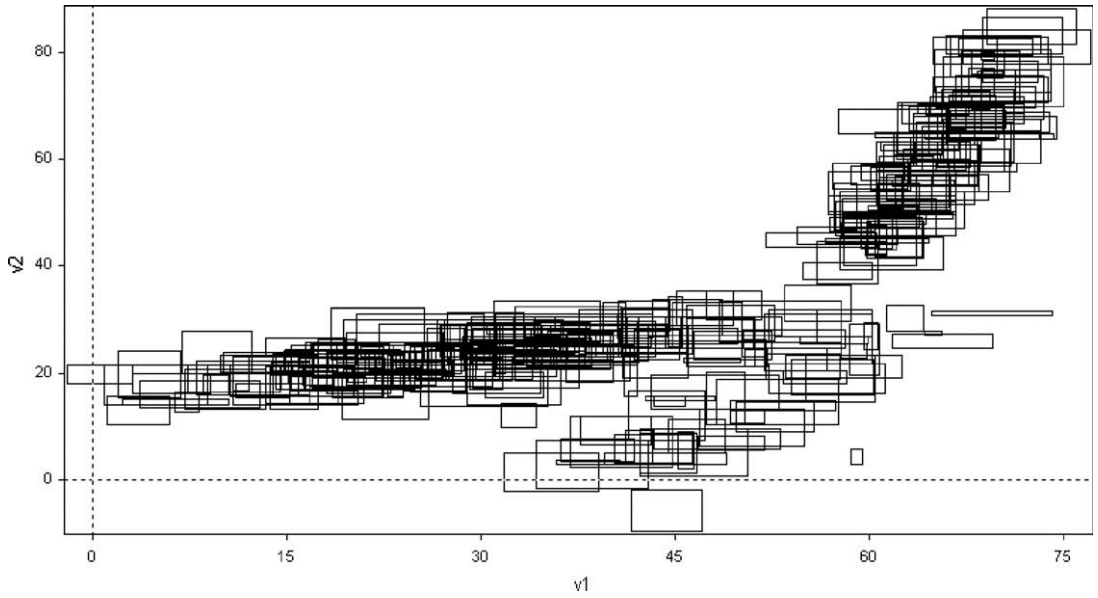

Fig. 1. Symbolic interval data set 1.

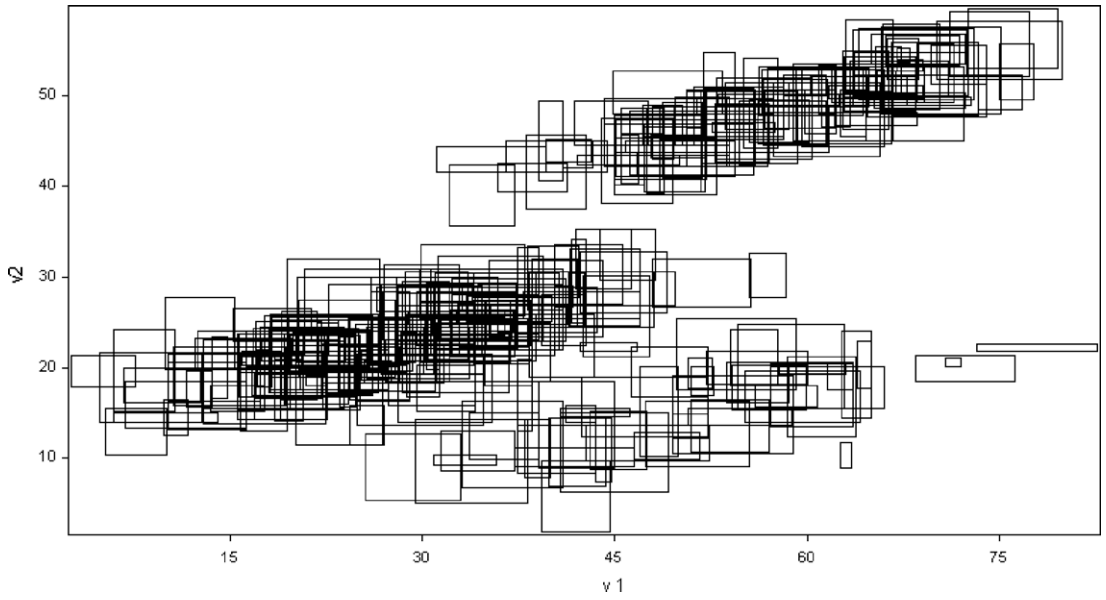

Fig. 2. Symbolic interval data set 2.

given in Eq. (27) as well as on the overall error rate of classification OERC. For each 100 replications, the average corrected Rand index CR and the average overall error rate of classification OERC are calculated.

Table 6 gives the values of the average and standard-deviation (in parenthesis) of the CR index obtained with these dynamic clustering algorithms and for interval data sets 1 and 2.

Table 7 gives the values of the average and standard-deviation (in parenthesis) of the OERC obtained with these dynamic clustering algorithms and for interval data sets 1 and 2.

The results in Table 6 for CR index and in Table 7 for OERC index show that the method based on single adaptive distances performs better than all the other methods in data configuration 1 (class covariances matrices of original quantitative data almost identical) whereas the method based on class adaptive distances is the best option in data configuration 2 (different class covariances matrices of original quantitative data). The L_2 method is the worse option for both configurations.

Table 9
CR index and OERC: comparison of the methods for the horse mixed feature-type symbolic data.

Accuracy measure	Single method	Class method	Non-adaptive method
CR index	0.138	0.329	0.138
OERC	0.417	0.333	0.417

Table 6
CR index: comparison of the clustering methods.

γ_i	Data set 1				Data set 2			
	Single method	Class method	Non-adaptive method	L_2 method	Single method	Class method	Non-adaptive method	L_2 method
[1,10]	0.645 (0.0102)	0.642 (0.0056)	0.833 (0.0079)	0.712 (0.0728)	0.576 (0.0036)	0.594 (0.0033)	0.604 (0.0066)	0634 (0.0500)
[1,20]	0.790 (0.0175)	0.767 (0.0108)	0.780 (0.0114)	0.717 (0.0735)	0.633 (0.0033)	0.690 (0.0058)	0.659 (0.0056)	0.632 (0.0520)
[1,30]	0.907 (0.0130)	0.819 (0.0096)	0.754 (0.0049)	0.717 (0.0656)	0.699 (0.0041)	0.749 (0.0041)	0.693 (0.0034)	0.628 (0.0469)
[1,40]	0.898 (0.0123)	0.819 (0.0086)	0.753 (0.0041)	0.718 (0.0678)	0.695 (0.0023)	0.765 (0.0035)	0.712 (0.0026)	0.631 (0.0490)

Table 7
OERC: comparison of the clustering methods.

γ_i	Data set 1				Data set 2			
	Single method	Class method	Non-adaptive method	L_2 method	Single method	Class method	Non-adaptive method	L_2 method
[1,10]	0.152 (0.0625)	0.144 (0.0469)	0.056 (0.0423)	0.119 (0.0380)	0.203 (0.0295)	0.190 (0.0485)	0.188 (0.0492)	0.162 (0.0283)
[1,20]	0.085 (0.0661)	0.086 (0.0402)	0.088 (0.0529)	0.118 (0.0390)	0.163 (0.0392)	0.135 (0.0435)	0.153 (0.0419)	0.164 (0.0309)
[1,30]	0.038 (0.0519)	0.057 (0.0460)	0.100 (0.0362)	0.117 (0.0341)	0.136 (0.0335)	0.098 (0.0299)	0.133 (0.0317)	0.166 (0.0275)
[1,40]	0.037 (0.0451)	0.060 (0.0382)	0.100 (0315)	0.116 (0.0343)	0.123 (0.0269)	0.094 (0.0247)	0.122 (0.0264)	0.164 (0.0289)

Table 8
Horse mixed feature-type data set described by seven interval-valued and three histogram-valued symbolic variables.

Symbolic variables	Horse/Label			
	ES/R	MA/R	···	AM/P
Country (support)	(1,2,3,4,5,6,7,8, 9,10,11,12,13,14,15)	(1,2,3,4,5,6,7,8, 9,10,11,12,13,14,15)	··· ···	(1,2,3,4,5,6,7,8, 9,10,11,12,13,14,15)
(weights)	(0.33,0.5,0,0,0.17,0,0,0, 0,0,0,0,0,0,0)	(0,0.33,0,0,0.67,0,0,0, 0,0,0,0,0,0,0)	··· ···	(0,0,0,0,0,0,0,0, 1,0,0,0,0,0,0)
Robe (support)	(1,2,3,4,5, 6,7,8,9,10)	(1,2,3,4,5, 6,7,8,9,10)	··· ···	(1,2,3,4,5, 6,7,8,9,10)
(weights)	(0.33,0.5,0.0,0.0,0.17, 0.0,0.0,0.0,0.0,0.0)	(0,0.33,0,0.33,0, 0,0.34,0,0,0)	··· ···	(0,0,1,0,0, 0,0,0,0,0)
Ability (support)	(1,2,3,4,5, 6,7,8,9)	(1,2,3,4,5, 6,7,8,9)	··· ···	(1,2,3,4,5, 6,7,8,9)
(weights)	(0.33,0,0,0.17,0.5, 0,0,0,0)	(0,0.33,0,0,0.67, 0,0,0,0)	··· ···	(0,0,0,0,0, 0,1,0,0)
Size (min)	[145,155]	[130,155]	···	[120,120]
Size (max)	[158,175]	[150,167]	···	[147,147]
Weight (min)	[410,460]	[390,430]	···	[170,170]
Weight	[550,630]	[570,580]	···	[290,290]
Mares	[150,480]	[0,200]	···	[230,230]
Stallions	[40,130]	[0,50]	···	[60,60]
Birth	[60,180]	[0,70]	···	[80,80]

4.2. Application with two real symbolic data sets

We apply the adaptive methods on two real symbolic data sets: horse and city temperature. In addition, we highlight the results of interpretation indices to the city temperature data set.

4.2.1. Horse mixed feature-type symbolic data set

The horse data set (available at http://www.ceremade.dauphine.fr/~touati) is a mixed feature-type symbolic data set that has 12 horses. Each horse is described by 7 interval-valued variables (Height at the withers (min), Height at the withers (max), Weight (min), Weight (max), Mares, Stallions, Birth) and 3 histogram-valued symbolic variables (Country, Robe and Aptitude). A categorical variable defines four a priori classes (Racehorse (R), Leisure Horse (L), Poney (P) and Draft horse (D)).

The a priori classification for this data set is as follows:

Racehorse (R): ES/R, MA/R, EN/R, AM/R
Leisure Horse (L): EN/L, AM/L, ES/L
Poney (P): EN/P, ES/P, AM/P
Draft horse (D): ES/D, EN/D

Table 8 shows part of this data set regarding three horses ES/R, MA/R and AM/P.

The clustering algorithm have been performed on this data set. The 4-cluster partition obtained with these clustering methods were compared with the 4-class partition known a priori based on the corrected Rand index CR as well as on the overall error rate of classification OERC.

Table 9 presents a comparison, based on CR index and OERC, among the dynamic clustering methods with non-adaptive and adaptive distances applied on the horse mixed feature-type symbolic data. The values in this table show that the class adaptive method is the best options for this horse data set.

Table 10 presents the clusters provided by these methods.

In order to compare the L_2 method for interval-valued data with the non-adaptive and adaptive methods presented in this paper, the modal variables of the horse data set were removed and an evaluation were carried out. Table 11 presents the comparison among the clustering methods. The values in this table show that the L_2 method for interval data outperforms the remain methods in terms of CR index and OERC.

Table 12 presents the clusters provided by these methods.

4.2.2. City temperature interval-valued data set

This interval-valued data set (Guru et al., 2004) concerns 37 cities, each city is described by 12 interval-valued variables which are minimum and the maximum temperatures of 12 months in degree centigrade.

Table 13 shows this data set.

A a priori classification given by a panel of human observers is as follows:

Class 1: Bahraim, Bombay, Cairo, Calcutta, Colombo, Dubal, Hong Kong, Kula Lampur, Madras, Manila, Mexico, Nairobi, New Delhi, Sydney and Singapore.
Class 2: Amsterdam, Athens, Copenhagen, Frankfurt, Geneva, Lisbon, London, Madrid, Moscow, Munich, New York, Paris, Rome, San Francisco, Seoul, Stockholm, Tokyo, Toronto, Vienna and Zurich.
Class 3: Mauritius.
Class 4: Tehran.

The clustering algorithm also have been performed on this data set. The 4-cluster partition obtained with these clustering methods were compared with the 4-class partition known a priori based on the corrected Rand index CR as well as on the overall error rate of classification OERC.

The CR index was 0,543 for all adaptive and non-adaptive clustering methods as well as for the L_2 method. They provide the same clusters and Table 14 shows these clusters. The error rate of classification for this data set was 0.297 for all adaptive and non-adaptive clustering methods and also for the L_2 based clustering method.

4.2.3. Partition and cluster interpretation: city temperature interval-valued data set

In order to show the usefulness of the partition and cluster interpretation indices introduced in Section 3, we consider the results obtained with the application of the dynamic clustering

Table 10
Clustering results for the horse mixed feature-type data set.

Method	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Non-adaptive	EN/L AM/R AM/L	MA/R EN/P ES/L	ES/R ES/D EN/R EN/D	ES/P AM/P
Single	EN/L AM/R AM/L	MA/R EN/P ES/L	ES/R ES/D EN/R EN/D	ES/P AM/P
Class	MA/R ES/L	EN/L AM/R AM/L	ES/R ES/D EN/R EN/D	EN/P ES/P AM/P

Table 11
CR index and OERC: comparison of the methods for the horse mixed feature-type symbolic data.

Accuracy	Single method	Class method	Non-adaptive method	L_2 method
CR index	0.209	0.138	0.209	0.366
OERC	0.417	0.417	0.417	0.333

Table 12
Clustering results for horse data set.

Method	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Non-adaptive	ES/P AM/P	ES/R ES/D EN/R EN/D	EN/P ES/L	MA/R EN/L AM/R AM/L
Single	ES/P AM/P	EN/P AM/L	MA/R EN/L AM/R ES/L	ES/R ES/D EN/R EN/D
Class	ES/R ES/P AM/P	MA/R EN/L AM/R ES/L	ES/D EN/R EN/D	EN/P AM/L
L_2	ES/R ES/D EN/R EN/D	ES/L	EN/P ES/P AM/P	MA/R EN/L AM/R AM/L

Table 13
Minimum and maximum temperatures of cities in degree centigrade (Guru et al., 2004).

Cities	January	February	March	April	May	June	July	August	September	October	November	December
Amsterdam	[−4,4]	[−5,3]	[2,12]	[5,15]	[7,17]	[10,20]	[10,20]	[12,23]	[0,20]	[5,15]	[1,10]	[−1,4]
Athens	[6,12]	[6,12]	[8,16]	[11,19]	[16,25]	[19,29]	[22,32]	[22,32]	[19,28]	[16,23]	[11,18]	[8,14]
Bahrain	13,19]	[14,19]	[17,23]	[21,27]	[25,32]	[28,34]	[29,36]	[30,36]	[28,34]	[24,31]	[20,26]	[15,21]
Bombay	[19,28]	[19,28]	[22,30]	[24,32]	[27,33]	[26,32]	[25,30]	[25,30]	[24,30]	[24,32]	[23,32]	[20,30]
Cairo	[8,20]	[9,22]	[11,25]	[14,29]	[17,33]	[20,35]	[22,36]	[22,35]	[20,33]	[18,31]	[14,26]	[10,20]
Calcutta	[13,27]	[16,29]	[21,34]	[24,36]	[26,36]	[26,33]	[26,32]	[26,32]	[26,32]	[24,32]	[18,29]	[13,26]
Colombo	[22,30]	[22,30]	[23,31]	[24,31]	[25,31]	[25,30]	[25,29]	[25,29]	[25,30]	[24,29]	[23,29]	[22,30]
Copenhagen	[−2,2]	[−3,2]	[−1,5]	[3,10]	[8,16]	[11,20]	[14,22]	[14,21]	[11,18]	[7,12]	[3,7]	[1,4]
Dubai	[13,23]	[14,24]	[17,28]	[19,31]	[22,34]	[25,36]	[28,39]	[28,39]	[25,37]	[21,34]	[17,30]	[14,26]
Frankfurt	[−10,9]	[−8,10]	[−4,17]	[0,24]	[3,27]	[7,30]	[8,32]	[8,31]	[5,27]	[0,22]	[−3,14]	[−8,10]
Geneva	[−3,5]	[−6,6]	[3,9]	[7,13]	[10,17]	[15,17]	[16,24]	[16,23]	[11,19]	[6,13]	[3,8]	[−2,6]
Hong Kong	[13,17]	[12,16]	[15,19]	[19,23]	[22,27]	[25,29]	[25,30]	[25,30]	[25,29]	[22,27]	[18,23]	[14,19]
KulaLumpur	[22,31]	[23,32]	[23,33]	[23,33]	[23,32]	[23,32]	[23,31]	[23,32]	[23,32]	[23,31]	[23,31]	[23,31]
Lisbon	[8,13]	[8,14]	[9,16]	[11,18]	[13,21]	[16,24]	[17,26]	[18,27]	[17,24]	[14,21]	[11,17]	[8,14]
London	[2,6]	[2,7]	[3,10]	[5,13]	[8,17]	[11,20]	[13,22]	[13,21]	[11,19]	[8,14]	[5,10]	[3,7]
Madras	[20,30]	[20,31]	[22,33]	[26,35]	[28,39]	[27,38]	[26,36]	[26,35]	[25,34]	[24,32]	[22,30]	[21,29]
Madrid	[1,9]	[1,12]	[3,16]	[6,19]	[9,24]	[13,29]	[16,34]	[16,33]	[13,28]	[8,20]	[4,14]	[1,9]
Manila	[21,27]	[22,27]	[24,29]	[24,31]	[25,31]	[25,31]	[23,29]	[24,28]	[25,28]	[24,29]	[22,28]	[22,27]
Mauritius	[22,28]	[22,29]	[22,29]	[21,28]	[19,25]	[18,24]	[17,23]	[17,23]	[17,24]	[18,25]	[19,27]	[21,28]
Mexico City	[6,22]	[15,23]	[17,25]	[18,27]	[18,27]	[18,27]	[18,27]	[18,26]	[18,26]	[16,25]	[14,25]	[8,23]
Moscow	[−13,−6]	[−12,−5]	[−8,0]	[0,8]	[7,18]	[11,23]	[13,24]	[11,22]	[6,16]	[1,8]	[−5,0]	[−11,5]
Munich	[−6,1]	[−5,3]	[−2,9]	[3,14]	[7,18]	[10,21]	[12,23]	[11,23]	[8,20]	[4,13]	[0,7]	[−4,2]
Nairobi	[12,25]	[13,26]	[14,25]	[14,24]	[13,22]	[12,21]	[11,21]	[11,21]	[11,24]	[13,24]	[13,23]	[13,23]
New Delhi	[6,21]	[10,24]	[14,29]	[20,36]	[26,40]	[28,39]	[27,35]	[26,34]	[24,34]	[18,34]	[11,28]	[7,23]
New York	[−2,4]	[−3,4]	[1,9]	[6,15]	[12,22]	[17,27]	[21,29]	[20,28]	[16,24]	[11,19]	[5,12]	[−2,6]
Paris	[1,7]	[1,7]	[2,12]	[5,16]	[8,19]	[12,22]	[14,24]	[13,24]	[11,21]	[7,16]	[4,10]	[1,6]
Rome	[4,11]	[5,13]	[7,16]	[10,19]	[13,23]	[17,28]	[20,31]	[20,31]	[17,27]	[13,21]	[9,16]	[5,12]
San Francisco	[6,13]	[6,14]	[7,17]	[8,18]	[10,19]	[11,21]	[12,22]	[12,22]	[12,23]	[11,22]	[8,18]	[6,14]
Seoul	[0,7]	[1,6]	[1,8]	[6,16]	[12,22]	[16,25]	[18,31]	[16,30]	[9,28]	[3,24]	[7,19]	[1,8]
Singapore	[23,30]	[23,30]	[24,31]	[24,31]	[24,30]	[25,30]	[25,30]	[25,30]	[24,30]	[24,30]	[24,30]	[23,30]
Stockholm	[−9,−5]	[−9,−6]	[−4,2]	[1,8]	[6,15]	[11,19]	[14,22]	[13,20]	[9,15]	[5,9]	[1,4]	[−2,2]
Sydney	[20,30]	[20,30]	[18,26]	[16,23]	[12,20]	[5,17]	[8,16]	[9,17]	[11,20]	[13,22]	[16,26]	[20,30]
Tehran	[0,5]	[5,8]	[10,15]	[15,18]	[20,25]	[28,30]	[36,38]	[38,40]	[28,30]	[18,20]	[9,12]	[−5,0]
Tokyo	[0,9]	[0,10]	[3,13]	[9,18]	[14,23]	[18,25]	[22,29]	[23,31]	[20,27]	[13,21]	[8,16]	[2,12]
Toronto	[−8,−1]	[−8,−1]	[−4,4]	[−2,11]	[−8,18]	[13,24]	[16,27]	[16,26]	[12,22]	[6,14]	[−1,17]	[−5,1]
Vienna	[−2,1]	[−1,3]	[1,8]	[5,14]	[10,19]	[13,22]	[15,24]	[14,23]	[11,19]	[7,13]	[2,7]	[1,3]
Zurich	[−11,9]	[−8,15]	[−7,18]	[−1,21]	[2,27]	[6,30]	[10,31]	[8,25]	[5,23]	[3,22]	[0,19]	[−11,8]

Table 14
Clustering results for the temperature data set.

Partition	Cities
Cluster 1	Bahrain, Cairo, Colombo, HongKong Madras, NewDelhi, Bombay, Calcutta, Dubai KulaLumpur, Manila, Singapore
Cluster 2	Athens, Madrid, Rome, Seoul, Tokyo, Lisbon, New York, San Francisco, Tehran
Cluster 3	Amsterdam, Frankfurt, London, Munich, Stockholm, Vienna, Copenhagen, Geneva Moscow, Paris, Toronto, Zurich
Cluster 4	Mauritius, Nairobi, Mexico City, Sydney

algorithm based on adaptive distances for each cluster on the city temperature interval-valued data set. Note that the results described below are related to histogram-valued variables (the original interval-valued variables were transformed into histogram-valued variables through a suitable homogenization pre-processing step).

4.2.3.1. Partition interpretation. The proportion of the overall SSQ explained by the 4-cluster partition was $R = 0.796$ (see Eq. 17).

Table 16
Cluster heterogeneity indices (%) (see Eqs. (20)–(22)).

Clusters	Cardinal	$T(k)$	$B(k)$	$W(k)$
1	9	42.9	45.9	30.9
2	12	13.6	9.2	30.3
3	4	34.5	36.0	28.3
4	12	9.0	8.7	10.5

Comparing the values of *COR* with the values of *R* (see Table 15) for the 4-cluster partition obtained using the adaptive clustering method for each class, we may conclude that the discriminant power of the variables 4 (april), 5 (may), 9 (september), 10 (october) and 11 (november) is above the average, whereas all other variables have a discriminant power below the average. Moreover, variable 10 (october) provides an important contribution to the separation of the prototypes of the clusters ($CTR = 19\%$).

4.2.3.2. Cluster's interpretation. Table 16 shows the cluster heterogeneity indices. From this Table, we can see that cluster 3 has the closest mean vector to the global center ($B(4) = 8.7$) and also it is the most homogeneous of the four clusters ($W(4) = 10.5$).

Table 15
Overall heterogeneity index concerning the variables (%) (see Eqs. 18 and 19).

Variables	1	2	3	4	5	6	7	8	9	10	11	12
COR	67.8	68.1	76.6	81.3	81.0	79.6	71.1	76.8	84.2	89.9	81.0	77.8
CTR	0.05	0.05	0.07	0.09	0.09	0.08	0.05	0.07	0.11	0.19	0.09	0.06

Table 17
Cluster heterogeneity indices concerning single variables (%) (see Eqs. (22)–(24)).

Variable	Cluster 1			Cluster 2			Cluster 3			Cluster 4		
	COR	*CTR*	*CE*	*COR*	*CTR*	*CE*	*COR*	*CTR*	*CE*	*COR*	*CTR*	*CE*
1	23.3	3.3	1.5	10.7	7.6	0.7	23.9	4.4	1.6	9.9	7.5	4.5
2	25.0	3.6	1.7	14.1	10.1	0.9	9.2	1.7	0.6	19.8	15.2	4.5
3	25.6	5.1	2.3	12.4	12.2	1.1	20.7	5.2	1.9	17.9	18.9	7.0
4	31.3	7.8	3.6	17.6	21.6	2.0	25.1	7.9	2.9	7.2	9.5	9.3
5	37.8	9.2	4.2	37.0	8.4	0.8	35.0	10.9	3.9	1.2	1.6	9.1
6	37.0	8.4	3.9	1.4	1.6	0.1	39.7	11.5	4.1	1.4	1.7	8.3
7	28.2	4.5	2.1	8.7	0.7	0.0	37.9	7.8	2.8	4.1	3.5	5.2
8	22.9	4.6	2.1	0.8	0.8	0.0	49.0	12.5	4.5	4.1	4.4	7.1
9	44.1	13.0	6.0	1.0	1.5	0.1	37.4	14.0	5.1	1.8	2.8	11.4
10	61.9	28.6	13.1	18.7	13.8	1.3	18.5	10.9	3.9	3.4	8.3	19.0
11	30.9	7.6	3.5	23.5	17.1	1.6	22.3	6.9	2.5	13.5	17.5	9.1
12	26.2	4.3	2.0	5.6	4.6	0.4	29.5	6.3	2.2	1.04	9.1	5.4

Table 17 shows the cluster heterogeneity indices concerning single variables.

From Table 17, we can see that variables 3 (March), 4 (April), 9 (September) and 10 (October) play the most important role in heterogeneity of the clusters 4, 2, 3 and 1, respectively ($CTR(3,4) = 18.9\%, CTR(4,2) = 21.6\%, CTR(9,3) = 14.0\%$ and $CTR(10,1) = 28.6\%$). Moreover, variables 2 (February), 5 (May), 8 (August) and 10 (October) had the most homogeneous behaviour in, respectively, clusters 4, 2, 3 and 1 ($COR(2,4) = 18.9\%, COR(5,2) = 37.0\%, COR(8,3) = 49.0\%$ and $COR(10,1) = 61.9\%$). Finally, variables 4 (April), 9 (September) and 10 (October) had the highest contribution to the eccentricity of, respectively, clusters 2, 3 and 1 and 4 ($CE(4,2) = 2.0\%$, $CE(9,3) = 2.0\%$, $CE(10,1) = 13.1\%$ and $CE(10,4) = 19.0\%$).

5. Concluding remarks

Two unsupervised pattern recognitions methods for mixed feature-type symbolic data based on dynamic clustering methodology with adaptive distances are presented. The adaptive clustering dynamic algorithm locally optimizes an adequacy criterion that measures the fitting between the clusters and their representatives (prototypes) based on distances that change at each iteration. The first method uses a single adaptive squared Euclidean distance that changes at each algorithm's iteration and the second one uses an adaptive squared Euclidean distance for each class that is different from one cluster to another and moreover changes at each iteration. The advantage of these adaptive distances is that the clustering algorithm is able to recognize clusters of different shapes and sizes. To be able to manage ordered and non-ordered mixed feature-type symbolic data, the methods assume a previous pre-processing step to obtain a suitable homogenization of mixed feature-type symbolic data into cumulative and non-cumulative histogram-valued symbolic data.

The adaptive dynamic clustering algorithm starts from an initial partition and alternates three steps until convergence when the adequacy criterion reaches a stationary value representing a local minimum. In the two first steps, the algorithm gives the solution for the best prototype of each cluster as well as the solution for the best single adaptive distance or for the best adaptive distance for each cluster. In last step, the algorithm gives the solution for the best partition. The convergence as well as the time complexity of this algorithm was addressed.

An experimental evaluation in order to show the usefulness of the proposed methods which use, respectively, single and class adaptive squared Euclidean distances for clustering mixed feature-type symbolic data was carried out. These adaptive distances were compared with a non-adaptive squared Euclidean distance for histogram-valued symbolic data. In addition, the methods were compared with a L_2 dynamic clustering algorithm for interval-valued symbolic data introduced by De Carvalho et al. (2006b). The accuracy of the results furnished by these clustering methods was assessed by the corrected Rand index and the overall error rate of classification regarding synthetic interval-valued data sets with different degree of class overlapping and clusters of different shapes and sizes in the framework of a Monte Carlo experience. An application with two real (interval-valued and mixed feature-type) symbolic data sets was also considered in this evaluation.

Synthetic interval data sets were constructed from configurations of quantitative data sets following bi-variate normal distributions with dependent components. The experimental evaluation for these data showed clearly the superiority, in terms of the quality of clusters as well as prediction accuracy, the partitioning clustering method based on the single adaptive squared Euclidean distance in a configuration of interval-valued data with a priori classes dispersion almost identical and the superiority of the partitioning clustering method based on the adaptive squared Euclidean distance for each class in a configuration of interval-valued data with unequal a priori classes dispersion. Concerning the application with the temperature interval-valued data, the methods had the same performance in terms of the quality of clusters and prediction accuracy. For horse mixed feature-type symbolic data set, the adaptive method for each class was the best option.

Acknowledgements

The authors thank the anonymous referees for their helpful suggestions and comments. We are also grateful to the Brazilian agencies CNPq, CAPES and FACEPE for their financial support.

References

Billard, L., Diday, E., 2007. Symbolic Data Analysis: Conceptual Statistics and Data Mining. Wiley.
Bock, H.-H., 2003. Clustering algorithms and kohonen maps for symbolic data. J. Jpn. Soc. Comp. Statist. 15, 217–229.
Bock, H.H., Diday, E., 2000. Analysis of symbolic data. Exploratory Methods for Extracting Statistical Information from Complex Data. Springer, Berlin, Heidelberg.
Celeux, G., Diday, E., Govaert, G., Lechevallier, Y., Ralambondrainy, H., 1989. Classification Automatique des Données. Bordas, Paris.
Chavent, M., 2000. Criterion-based divisive clustering for symbolic data. In: Bock, H.H., Diday, E. (Eds.), Analysis of symbolic data, Exploratory Methods for Extracting Statistical Information from Complex Data. Springer, Berlin, Heidelberg, pp. 299–311.
Chavent, M., Lechevallier, Y., 2002. Dynamical clustering algorithm of interval data: Optimization of an adequacy criterion based on Hausdorff distance. In: Sokolowsky, , Bock, H.H., Jaguja, K.A. (Eds.), Classification, Clustering and Data Analysis (IFCS2002). Springer, Berlin, pp. 53–59.
Chavent, M., De Carvalho, F.A.T., Lechevallier, Y., Verde, R., 2003. Trois nouvelles mthodes de classification automatique de donnes symboliques de type intervalle. Revue de Statistique Applique LI 4, 5–29.
De Carvalho, F.A.T., 1995. Histograms in symbolic data analysis. Annals of Operations Research 55, 229–322.

De Carvalho, F.A.T., Lechevallier, Y., 2009. Partitional clustering algorithms for symbolic interval data based on single adaptive distances. Pattern Recognition 42, 1223–1236.
De Carvalho, F.A.T., Verde, R., Lechevallier, Y., 1999. A dynamical clustering of symbolic objcts based on a context dependent proximity measure. In: Proceedings of the IX International Symposium on Applied Stochastic Models and Data analysis. Lisboa, Universidade de Lisboa, pp. 237–242.
De Carvalho, F.A.T., Souza, R.M.C.R., Chavent, M., Lechevallier, Y., 2006a. Adaptive Hausdorff distances and dynamic clustering of symbolic data. Pattern Recognition Lett. 27 (3), 167–179.
De Carvalho, F.A.T., Brito, P., Bock, H.H., 2006b. Dynamic clustering for interval data based on L2 distance. Computat. Statist. (2), 231–25.
Diday, E., 1971. La méthode des Nuées dynamiques em Revue de Statistique Appliquée 19 (2), 19–34.
Diday, E., Brito, M.P., 1989. Symbolic cluster analysis. In: Opitz, O. (Ed.), Conceptual and Numerical Analysis of Data. Springer-Verlag, Heidelberg, pp. 45–84.
Diday, E., Govaert, G., 1977. Classification automatique avec distances adaptatives. R.A.I.R.O. Informatique Computer Science 11 (4), 329–349.
Diday, E., Noirhome-Fraiture, M., 2008. Symbolic Data Analysis and the SODAS Software. Wiley.
Diday, E., Simon, J.C., 1976. Clustering analysis. In: Fu, K.S. (Ed.), Digital Pattern Clasification. Springer, Berlin, pp. 47–94.
Gordon, A.D., 1999. Classification. Chapman and Hall/CRC, Boca Raton, Florida.
Gordon, A.D., 2000. An Iteractive relocation algorithm for classifying symbolic data. In: Data Analysis: Scientific Modeling and Practical Application. Springer-Verlag, Berlin, pp. 17–23.
Gowda, K.C., Diday, E., 1991. Symbolic clustering using a new dissimilarity measure. Pattern Recognition 24 (6), 567–578.
Gowda, K.C., Ravi, T.V., 1995a. Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. Pattern Recognition 28 (8), 1277–1282.
Gowda, K.C., Ravi, T.V., 1995b. Agglomerative clustering of symbolic objects using the concepts of both similarity and dissimilarity. Pattern Recognition Lett. 16, 647–652.
Gowda, K.C., Ravi, T.V., 1999. Clustering of symbolic objects using gravitational approach. IEEE Trans. Systems Man Cybernet. 29 (6), 888–894.
Guru, D.S., Kiranagi, B.B., 2005. Multivalued type dissimilarity measure and concept of mutual dissimilarity value for clustering symbolic patterns. Pattern Recognition 38, 151–256.
Guru, D.S., Kiranagi, B.B., Nagabhushan, P., 2004. Multivalued type dissimilarity measure and concept of mutual dissimilarity value for clustering symbolic patterns. Pattern Recognition 38, 1203–1213.
Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2, 193–218.
Ichino, M., Yaguchi, H., 1994. Generalized Minkowski metrics for mixed feature type data analysis. IEEE Trans. Systems Man Cybernet. 24 (4), 698–708.
Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Computing Surveys 31 (3), 264–323.
Ralambondrainy, H., 1995. A onceptual version of the K-means algorithm. Pattern Recognition Lett. 16, 1147–1157.
Souza, R.M.C.R., De Carvalho, F.A.T., 2004. Clustering of interval data based on city-block distances. Pattern Recognition Lett. 25 (3), 353–365.
Verde, R., De Carvalho, F.A.T., Lechevallier, Y., 2001. A dynamical clustering algorithm for symbolic data. In: Tutorial on Symbolic Data Analisys, GfKl Conference, Munich.