New Clustering methods for interval data

Marie Chavent, Francisco de A.T. de Carvahlo, Yves Lechevallier, Rosanna Verde

New clustering methods for interval data

Marie Chavent, Francisco de A.T. de Carvalho, Yves Lechevallier and Rosanna Verde

MAB-Mathématiques Appliquées de Bordeaux, Université Bordeaux1, 351 cours de la libération, 33405 Talence cedex, France
CIn - Centro de Informática UFPE - Universidade Federal de Pernambuco Av. Prof. Luiz Freire s/n - Cidade Universitária - CEP 50740-540 Recife-PE Brasil
INRIA - Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau - Rocquencourt B.P. 105 - 78153 Le Chesnay Cedex, France
Dip. Strategie Aziendali e Metodologie Quantitative - SUN - Seconda Università di Napoli, Corso Gran Priorato di Malta, 81043 Capua, Italie

Summary

In this paper we propose two clustering methods for interval data based on the dynamic cluster algorithm. These methods use different homogeneity criteria as well as different kinds of cluster representations (prototypes). Some tools to interpret the final partitions are also introduced. An application of one of the methods concludes the paper.

Computational Statistics 21, 211-229. 2006.

Keywords: Dynamic clustering, interval data, distances, prototypes

1 Introduction

In this paper we propose two new methods suitable for clustering a special kind of data, called symbolic data, and characterized by multi-valued descriptors (Diday (1988), Bock & Diday (2000)). Such data are usually collected in a symbolic data table, where the individuals are represented in the rows and the multi-valued variables in the columns. The cells contain multi-values (intervals, multi-categories, distributions). In the present context of interval data analysis, each cell contains an interval of real values. In Table 1, we show an example of an interval data table, where the columns represent the monthly intervals of temperatures and the rows (objects) describe 60 Chinese meteorological stations.

Stations	January	February	...	November	December
AnQing	[1,8:7,1]	[2,1:7,2]	...	[7,8:17,9]	[4,3:11,8]
BaoDing	[-7,1:1,7]	[-5,3:4,8]	...	[0,8:14]	[-3,9:5,2]
BeiJing	[-7,2:2,1]	[-5,9:3,8]	...	[1,5:12,7]	[-4,4:4,7]
...	...	...	...	...	...
ZhiJiang	[2,7:8,4]	[2,7:8,7]	...	[8,2:20]	[5,1:13,3]

Table 1: Monthly averages of minimal and maximal daily temperatures observed on 60 Chinese meteorological stations in 1988

Symbolic Data Analysis has provided partitioning methods in which different types of symbolic data are considered. Diday & Brito (1989) used a transfer algorithm to partition a set of symbolic objects into clusters described by distribution vectors. Ralambondrainy (1995) extended the classical k-means clustering method in order to deal with data characterized by numerical and categorical variables. Verde et al. (2000) generalized the dynamic clustering algorithm to partition categorical multivalued symbolic data. In this approach a context-dependent proximity measure is used as an allocation function. Gordon (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. Bock (2001) proposed several clustering algorithms for symbolic data described by interval variables, based on a clustering criterion and thereby generalized similar approaches in classical data analysis. Recently, De Souza & De Carvalho (2004) have proposed partitioning clustering methods for interval data based on city-block distances.

The two new clustering approaches presented in this paper are based on the Dynamic Cluster Algorithm (DCA) introduced by Diday (1971), Diday & Simon (1976). Recall that DCA needs to define an allocation function (a dissimilarity measure) and a way to represent the classes (prototypes). The methods hereafter proposed are defined with different dissimilarity measures, criteria and prototypes.

In the first approach the prototypes are defined in the same representation space as the objects to be clustered. Hence, the prototypes are described by vectors of intervals. The dissimilarity function used to compare an object to the description of a prototype (more generally, to compare two vectors of intervals) is based on the Hausdorff distance (Chavent & Lechevallier (2002)). In the second approach, the prototypes and the objects are not in the same description space and the comparison function is not a dissimilarity but a matching function.

Some criteria are proposed to interpret the quality of the partitions achieved by the first or the second clustering method. An application of the first method is performed on real data: 60 meteorological Chinese stations described each month by an interval of temperatures (Long-Term Instrumental Climatic Data Base of the People's Republic of China http://dss.ucar.edu/datasets/ds578.5/data/).

2 Notations and definitions

An interval variable Y is a correspondence from a set E into the set of real values $\Re$ which has the following property on its graph: for all $s \in E$, the sub-set $[a, b] = Y(s)$ is a bounded interval of $\Re$.

Let $E = \{1, ..., s, ..., n\}$ be a set of n objects described by p interval variables $Y_1, \ldots, Y_j, \ldots, Y_p$. The interval data table is then a matrix $(x_s^j)_{n \times p}$ where the n rows describe the n objects to be clustered and the p columns correspond to the p interval variables. Each cell of the data table contains a bounded interval $x_s^j = [a_s^j, b_s^j]$ of $\Re$.

We will note :

- $x_s = (x_s^1, ..., x_s^p)$ the vector of intervals describing the object s
- $P = (C_1, ..., C_i, ..., C_k)$ a partition in k clusters of E
- $G_i = (g_i^1, \ldots, g_i^j, \ldots, g_i^p)$ a prototype of the cluster C_i
- Λ a representation space of the prototype G_i

3 The dynamic cluster algorithm

The aim of the dynamic cluster algorithm is to find a partition $P^* = (C_1, ..., C_k)$ of E in k non empty clusters and a vector $L^* = (G_1, \ldots, G_i, \ldots, G_k)$ of k prototypes so that both P^* and L^* optimize a criterion Δ:

$$\Delta(P^*, L^*) = Min\,\{\Delta(P, L) \;/\; P \in P_k, L \in \Lambda^k\} \tag{1}$$

with P_k the set of all the k-clusters partitions of E and Λ the representation space of the prototypes .

The criterion Δ measures the adequacy between a partition P and a vector L of k prototypes. This criterion is defined as the sum on the k clusters C_i and on all the objects $s \in C_i$ of dissimilarities $D(x_s, G_i)$:

$$\Delta(P, L) = \sum_{i=1}^{k} \sum_{s \in C_i} D(x_s, G_i) \tag{2}$$

This algorithm alternately performs a *representation* and an *allocation* step. In the particular case of classical (non interval) real data and of prototypes defined as the barycenters of the clusters, the dynamic cluster algorithm is equivalent to the *k-means* batch algorithm.

In order to introduce the next sections, we recall the general scheme of the DCA:

DYNAMIC CLUSTER ALGORITHM (DCA)

a) *Initialization*: Start from a random partition $P = (C_1, \ldots, C_i, \ldots, C_k)$ or, alternatively, from a vector $(G_1, \ldots, G_i, \ldots, G_k)$ of k prototypes randomly chosen among the elements of E. In such a case, an allocation step is achieved as follows:

- $C_i = \emptyset$ for $i = 1, \ldots, k$
- For $s = 1$ to n do:
  - Assign s to cluster C_l such that $l = argmin_{i=1,\ldots,k} D(x_s, G_i)$
  - $C_l = C_l \cup \{s\}$

b) *representation step*: for $i = 1$ to k, perform the prototype G_i which minimizes the criterion:

$$f_{C_i}(G) = \sum_{s \in C_i} D(x_s, G), \quad G \in \Lambda \tag{3}$$

c) *allocation step*

- $test \longleftarrow 0$
- for $s = 1$ to n do:
  - Find the cluster C_m to which s belongs

* Find the index l such that:

$$l = argmin_{i=1,\ldots,k} D(x_s, G_i)$$

* if $l \neq m$
  - $test \longleftarrow 1$
  - $C_l = C_l \cup \{s\}$ and $C_m = C_m - \{s\}$

d) if $test = 0$ then stop, otherwise go to b)

At each iteration of this algorithm, a new couple (P, L) is found and the decrease of the Δ criterion can be proved under the following conditions:

- uniqueness of the affectation cluster for each object $s \in E$
- uniqueness of the prototype G_i minimizing the criterion f_{C_i} given in (3) for all the clusters C_i of the partition P of E.

The uniqueness of a cluster to which an object s is allocated (in the case of equality of the distances) is easy to solve by assigning s to the cluster having the smallest index.

The existence and the uniqueness of the prototype G_i is however more difficult to prove because it depends on the comparison function D. In the next section we propose two different prototypes for a cluster of interval data, related to the choice of two different comparison functions D which are then parameters of two different clustering algorithms.

4 Two new clustering methods

The prototype G of a cluster C is defined according to the criterion $f_C(G)$ (defined in (3)) and optimized as an adequacy measure between the prototype and the cluster. Because this criterion is based on the function D, chosen to compare the prototype and an object to be clustered, two prototypes will be defined according to the choice of the two different comparison functions D.

4.1 The first method

The first dynamical clustering method, here proposed, compares two vectors of intervals x_1 and x_2 with a distance d_1 based on the Hausdorff distance. We do not use here the Hausdorff distance on a real $\Re^p$-set, as in Chavent (2004), but the sum of Hausdorff distances between intervals. First, we recall the definition of the Hausdorff distance in the case of two intervals and,

starting from this definition, we deduce the distance d_1 between two vectors of intervals used in this first approach. Then, we look for an explicit formula of the prototype G of the cluster C able to optimize the adequacy criterion f_C (Chavent & Lechevallier (2002)), based on d_1.

4.1.1 Definition of the Hausdorff-based distance

The Hausdorff distance (Nadler 1978), (Rote 1991) is often used in image processing (Huttenlocher et al. 1993). It is used to compare two sets of objects A and B. Such a distance d_H depends on the particular metric d (L_1 norm, L_2 norm, etc.) chosen to compare two objects u and v in A and B, respectively:

$$d_H(A,B) = \max(h(A,B), h(B,A)) \tag{4}$$

where

$$h(A,B) = \sup_{u \in A} \inf_{v \in B} d(u,v) \tag{5}$$

Let A and B be two intervals in $\Re$, $d(u,v)$ is simply $|u-v|$ and it is easy to show that the Hausdorff distance between two intervals $x_1^j = [a_1^j, b_1^j]$ and $x_2^j = [a_2^j, b_2^j]$ is:

$$d_H(x_1^j, x_2^j) = max(|a_1^j - a_2^j|, |b_1^j - b_2^j|) \tag{6}$$

Finally, the distance d_1 between two vectors of intervals x_1 and x_2 is the sum on the p variables of the Hausdorff distances between the intervals:

$$d_1(x_1, x_2) = \sum_{j=1}^{p} max(|a_1^j - a_2^j|, |b_1^j - b_2^j|) \tag{7}$$

For intervals reduced to single points, that is when $a^j = b^j$, d_1 is the L_1 distance in $\Re^p$.

4.1.2 The prototype

The prototype $G = (g^1, ..., g^p)$ of a cluster C is the vector of p intervals which minimizes the adequacy criterion:

$$f_C(G) = \sum_{s \in C} d_1(x_s, G) = \sum_{s \in C} \sum_{j=1}^{p} d_H(x_s^j, g^j) \tag{8}$$

Criterion (8) can also be written as:

$$f_C(G) = \sum_{j=1}^{p} \overbrace{\sum_{s \in C} d_H(x_s^j, g^j)}^{\tilde{f_C}(g^j)} \tag{9}$$

and the problem is now to find the interval $g^j = [\alpha^j, \beta^j]$ for $(j = 1, ..., p)$ which minimizes:

$$\tilde{f_C}(g^j) = \sum_{s \in C} d_H(x_s^j, g^j) = \sum_{s \in C} \max(|\alpha^j - a_s^j|, |\beta^j - b_s^j|) \tag{10}$$

We will see how to solve this minimization problem by transforming it into two well-known minimization problems. Let m_s^j be the midpoint of an interval $x_s^j = [a_s^j, b_s^j]$ and l_s^j be half of its length, i.e.:

$$m_s^j = \frac{a_s^j + b_s^j}{2} \text{ and } l_s^j = \frac{b_s^j - a_s^j}{2} \tag{11}$$

and let μ^j and λ^j be the midpoint and half-length of the interval $g^j = [\alpha^j, \beta^j]$, respectively. According to the following property defined for x and y in $\Re$:

$$\max(|x - y|, |x + y|) = |x| + |y| \tag{12}$$

the function (10) can be written as:

$$\begin{aligned} \tilde{f_C}(g^j) &= \sum_{s \in C} \max(|(\mu^j - \lambda^j) - (m_s^j - l_s^j)|, |(\mu^j + \lambda^j) - (m_s^j + l_s^j)|) \\ &= \sum_{s \in C} |\mu^j - m_s^j| + \sum_{s \in C} |\lambda^j - l_s^j| \end{aligned} \tag{13}$$

This yields two well-known minimization problems: find $\mu^j \in \Re$ and $\lambda^j \in \Re$ which minimizes, respectively:

$$\sum_{s \in C} |\mu^j - m_s^j| \text{ and } \sum_{s \in C} |\lambda^j - l_s^j| \tag{14}$$

The solutions $\hat{\mu^j}$ and $\hat{\lambda^j}$ are respectively the median of $\{m_s^j, s \in C\}$, which are the midpoints of the intervals $x_s^j = [a_s^j, b_s^j]$, $s \in C$, and the median of the set $\{l_s^j, s \in C\}$ of their half-lengths. Finally, the solution $\hat{g^j} = [\hat{\alpha^j}, \hat{\beta^j}]$ is the interval $[\hat{\mu^j} - \hat{\lambda^j}, \hat{\mu^j} + \hat{\lambda^j}]$ and the prototype of C is $G = (\hat{g}^1, ..., \hat{g}^p)$.

4.2 The second method

The second dynamic clustering method, proposed in this paper, compares two vectors of intervals by means of a dissimilarity d_2. This dissimilarity compares two couples $p_1^j = (S_1^j, q_1)$ and $p_2^j = (S_2^j, q_2)$, where q_1 and q_2 are weight functions associated, respectively, to suitable sets of intervals S_1^j and S_2^j.

These subsets are obtained by a pre-processing step descretizing the intervals x_1^j and x_2^j in "basic" or elementary intervals. Firstly, we describe how to get such subsets of intervals and the associating weight functions. Then, we introduce the "two components" dissimilarity measure d_{2c} to compare p_1^j and p_2^j and we define d_2 as the sum of those dissimilarities with respect to the p variables. Finally, we perform a prototype G.

4.2.1 Pre-processing step

The aim of the pre-processing step (De Carvalho (1995), De Carvalho et al. (1999), Chavent et al. (2003)) is to discretize x_s^j to obtain a subset of a set $\{I_1^j, \ldots, I_{H_j}^j\}$ of elementary intervals and the corresponding set of weights q_s^j.

A column j of the data table is a set $\{x_1^j, ..., x_s^j, ..., x_n^j\}$ of n intervals. Starting from this set of intervals, another set of H_j disjoint intervals $\{I_1^j, ..., I_h^j, ..., I_{H_j}^j\}$ is performed. The elementary intervals are obtained by sorting the set of lower and upper bounds of the n intervals $\{x_1^j, ..., x_s^j, ..., x_n^j\}$. The so called "elementary" intervals I_h^j have to verify the following properties:

i) $\bigcup_{h=1}^{H_j} I_h^j = \bigcup_{s=1}^{n} x_s^j$

ii) $I_h^j \cap I_{h'}^j = \emptyset$ if $h \neq h'$

iii) $\forall s \in E, \forall h \quad I_h^j \subseteq x_s^j$ or $I_h^j \cap x_s^j = \emptyset$

iv) $\forall s \in E, \exists S_s^j \subset \{I_1^j, ..., I_{H_j}^j\} : \bigcup_{I_h^j \in S_s^j} I_h^j = x_s^j$ and $\forall I_h^j \in S_s^j, I_h^j \subseteq x_s^j$

Then, $S_s^j = \{I_h^j : I_h^j \subseteq x_s^j\}$. The weight function q_s defined on the subset of elementary intervals S_s^j which discretizes the interval x_s^j is then defined as:

$$\begin{array}{lcl} q_s : S_s^j & \longrightarrow & [0,1] \\ I_h^j \in S_s^j & \longrightarrow & q_s(I_h^j) = \frac{|I_h^j|}{b_s^j - a_s^j} \end{array} \tag{15}$$

where $|I_h^j|$ is the length of the interval I_h^j. Notice that, $\forall I_h^j \in S_s^j$, $q_s(I_h^j) \geq 0$ and that $\sum_{I_h^j \in S_s^j} q_s(I_h^j) = 1$.

An example is given in Figure 1. Let us consider four intervals x_1^j, x_2^j, x_3^j and x_4^j (describing a set E of four objects on a variable Y_j). The set of elementary intervals $\{I_1^j, I_2^j, I_3^j, I_4^j, I_5^j\}$ is shown and, because the set of elementary intervals which discretize the interval x_1^j is $\{I_1^j, I_2^j\}$, their associated weights are $q_1(I_1^j) = \alpha$ and $q_1(I_2^j) = 1 - \alpha$.

Figure 1: Elementary intervals construction

4.2.2 Definition of the "two components" dissimilarity

As in the first method, where we compared two intervals x_1^j and x_2^j (for each variable j) by the Hausdorff distance, here we compare the couples $p_1^j = (x_1^j, q_1)$ and $p_2^j = (x_2^j, q_2)$ by a "two components" dissimilarity measure, noted as d_{2c}, and defined as:

$$d_{2c}(p_1^j, p_2^j) = d_{ci}(x_1^j, x_2^j) + d_{cd}(q_1, q_2) \tag{16}$$

where d_{ci} is a dissimilarity which measures the difference in "position" between two intervals, and d_{cd} is a dissimilarity between two weight functions q_1 and q_2 which express the difference in content between two intervals (De Carvalho & Souza (1998)).

The first "component" of the dissimilarity d_{2c} is the dissimilarity d_{ci} between two intervals $x_1^j = [a_1^j, b_1^j]$ and $x_2^j = [a_2^j, b_2^j]$ defined by:

$$d_{ci}(x_1^j, x_2^j) = \frac{|(\bar{x}_1^j \cap \bar{x}_2^j) \cap (x_1^j \oplus x_2^j)|}{|x_1^j \oplus x_2^j|} \tag{17}$$

where:

- $|.|$ the length of an interval
- $x_1^j \oplus x_2^j = [min(a_1^j, a_2^j), max(b_1^j, b_2^j)]$
- $\bar{x}_s^j =]-\infty, a_s^j[\cup]b_s^j, +\infty[$ the complementary set of x_s^j in $\Re$

Another formulation of this dissimilarity is:

$$d_{ci}(x_1^j, x_2^j) = \begin{cases} \frac{|min(b_1^j, b_2^j) - max(a_1^j, a_2^j)|}{max(b_1^j, b_2^j) - min(a_1^j, a_2^j)} & \text{if } x_1^j \cap x_2^j = \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Notice that, according to the d_{ci} component, there is difference in "position" between two intervals only when the intersection between them is empty.

The second "component" of d_{2c} is the dissimilarity d_{cd} between the two weight functions q_1 and q_2 defined, respectively, on subsets S_1^j and S_2^j of elementary intervals which discretizes x_1^j and x_2^j:

$$d_{cd}(q_1, q_2) = \frac{1}{2}\Big(\sum_{\{I_h^j : I_h^j \in S_1^j, I_h^j \notin S_2^j\}} q_1(I_h^j) + \sum_{\{I_h^j : I_h^j \in S_2^j, I_h^j \notin S_1^j\}} q_2(I_h^j)\Big) \tag{19}$$

Notice that $0 \leq d_{cd} \leq 1$, with $d_{cd} = 0$ if $x_1^j = x_2^j$ and $d_{cd} = 1$ if $x_1^j \cap x_2^j = \emptyset$.

Finally the dissimilarity d_2 between the two couples p_1 and p_2 is the sum of the dissimilarities d_{2c} computed with respect to the p variables Y_j's:

$$d_2(p_1, p_2) = \sum_{j=1}^{p} d_{2c}(p_1^j, p_2^j) = \sum_{j=1}^{p} (d_{ci}(x_1^j, x_2^j) + d_{cd}(q_1, q_2)) \tag{20}$$

4.2.3 The prototype

The prototype G of a cluster C is now a vector where the components are p couples (Γ^j, q). The components of the prototype are defined in the following way:

- Γ^j may be defined in two different ways (see Figure 2):
  (a) $\Gamma^j = [\min_{s \in C} a_s^j, \max_{s \in C} b_s^j]$ is an interval "generalizing" the intervals $x_s^j = [a_s^j, b_s^j]$, for $s \in C$;
  (b) $\Gamma^j = \{x_s^j : s \in C\}$.

  In both the cases the intervals can be considered discretized, in the pre-processing step, into elementary intervals, as above described.

- q is defined as $g = \begin{cases} \frac{1}{card(C)} \sum_{\{h,s : I_h \in S_s^j \text{ and } s \in C\}} q_s(I_h^j) \\ 0, \text{ otherwise} \end{cases}$

Because in the definition (b) of Γ^j the result is not an interval, the prototype is not, in this particular case, represented in the same description space as

Figure 2: The two different ways to define Γ

the objects of the cluster. The dissimilarity d_2 cannot be directly used to compare an object s and the prototype G. For this reason, we propose to replace in d_2 and, more precisely, in d_{2c} (in 16) the first component d_{ci} by:

$$d^*_{ci}(x^j_s, \Gamma^j) = \frac{|(\bar{x}^j_s \cap (\bigcap_{s' \in C} \bar{x}^j_{s'})) \cap (x^j_s \oplus (\bigcup_{s' \in C} x^j_{s'})|}{|x^j_s \oplus (\bigcup_{s' \in C} x^j_{s'}))|} \tag{21}$$

with $x^j_s \oplus (\bigcup_{s' \in C} x^j_{s'}) = [min(a^j_s, \min_{s' \in C} a^j_{s'}), max(b^j_s, \max_{s' \in C} b^j_{s'})]$

The comparison function D (used in the allocation step of the algorithm) between an object s and the prototype G is then defined by:

$$d^*_2(p_s, G) = \sum_{j=1}^{p} (d^*_{ci}(x^j_s, \Gamma^j) + d_{cd}(q_s, q)) \tag{22}$$

Remark: Because p_s and G are not in the same description space, d^*_2 is not a dissimilarity function but a "matching" function.

Moreover, the prototype G here defined doesn't minimize the adequacy function (as in the representation step of the dynamic cluster algorithm). For this reason this second clustering approach is not a proper dynamic cluster method and an exchange algorithm is performed.

4.2.4 The algorithm

The algorithm used in the second method is based on the exchange algorithm (Chavent et al. (2003)). At each iteration of this algorithm the n objects are considered one after the other and are assigned to the cluster such that the decrease of the Δ criterion is maximum. As in the first version of the well known k-means algorithm, the prototypes are updated at each moving of an object from one cluster to another:

EXCHANGE ALGORITHM:

a) *Initialization*: Start from a random partition $P = (C_1, \ldots, C_i, \ldots, C_k)$

c) *allocation step*

- $test \longleftarrow 0$
- for $s = 1$ to n do:
  - Find the cluster C_m to which s belongs
  - If $card(C_m) \neq 1$ for $l = 1, ..., k$ and $l \neq m$
    - perform the new prototypes G_m of $C_m - \{s\}$ and G_l of $C_l \cup \{s\}$
    - perform the criterion $\Delta_l = \sum_{i=1}^{k} \sum_{s' \in C_i} D(p_{s'}, G_i)$ where D can be d_2 or d_2^* according to the selected type of prototype
  - find the cluster C_{l^*} such that

$$l^* = arg \min_{l=1,\ldots,k} \Delta_l$$

  - if $l^* \neq m$ move s to C_{l^*}
    - $test \longleftarrow 1$
    - $C_{l^*} = C_{l^*} \cup \{s\}$ and $C_m = C_m - \{s\}$

b) if $test = 0$ then stop, otherwise go to b)

5 Interpretation

The aim of this section is to provide the user with various criteria to measure and interpret the quality of the partition or the quality of the clusters of the partition. These criteria are obtained by generalizing some criteria proposed in Celeux et al. (1989) for a partition P of n points x_s of $\Re^p$ weighted by p_s and performed by dynamical clustering. All these criteria are based on the decomposition of the total inertia into within-cluster and between-cluster inertia. In order to simplify our presentation we take $p_s = 1$ $(\forall s = 1, ..., n)$ and, in this particular case, the inertia is the total sum of squares (TSS) of the $\Re^p$ points around their mean. The decomposition of the total sum of squares (TSS) into the within-cluster (WSS) and between-cluster (BSS) sum of squares is:

$$\underbrace{\sum_{s=1}^{n} d^2(x_s, G)}_{TSS} = \underbrace{\sum_{i=1}^{k} \sum_{s \in C_i} d^2(x_s, G_i)}_{WSS} + \underbrace{\sum_{i=1}^{k} n_i d^2(G_i, G)}_{BSS} \tag{23}$$

where d is the squared Euclidean distance, G is the mean of the n points $x_s \in E$, G_i is the mean of the points $x_s \in C_i$ and $n_i = card(C_i)$.

A well known result is that the mean G of a cluster C is the point $g \in \Re^p$ which minimizes the following adequacy criterion:

$$f_C(g) = \sum_{s \in C} d^2(x_s, g)$$

In the first clustering method proposed in section 4.1 we generalized the idea of the mean G of a cluster C to the idea of a prototype G which minimizes the adequacy criterion:

$$f_C(g) = \sum_{s \in C} D(x_s, g) \tag{24}$$

where $D = d_1$ is based on the Hausdorff distance and the solution $G = (\hat{g}^1, ..., \hat{g}^p)$ is a vector of intervals (defined in section 4.1.2).

The TSS (or the total inertia) and the WSS (or the within-cluster inertia) defined in (23) can then be generalized by using a prototype G_i of a cluster C_i which optimizes the adequacy criterion (24) for a specific comparison function D. We then have :

- $WSS = \sum_{i=1}^{k} \sum_{s \in C_i} D(x_s, G_i) = \sum_{i=1}^{k} f_{C_i}(G_i)$ which is then equal to the criterion $\Delta(P, L)$ given in (1).
- $TSS = \sum_{s=1}^{n} D(x_s, G_E)$ which is the adequacy criterion $f_E(G_E)$ defined in (24) with G_E is the prototype of the whole set of n objects in E .

Of course, the equality (23) is not true after generalization. The gain of inertia obtained by replacing the propotype G of E by the k prototypes $(G_1, ..., G_k)$ of the partition P is no more the between-cluster inertia (or BSS). The gain of homogeneity obtained by replacing the n objects by the k prototypes is simply defined as the difference between $f_E(G_E)$ and $\Delta(P, L)$.

Finally the three following criteria will be used to interpret a partition and its clusters:

- $f_{C_i}(G_i)$ which is a measure of homogeneity of the cluster C_i;
- $\Delta(P, L)$ which is a measure of within-cluster homogeneity of the partition P;
- $f_E(G_E)$ which is a measure of total homogeneity of the set E.

Remark: Because in the second method the prototypes are not defined by the minimization of an adequacy criterion, the criteria given below can not be applied to interpret a partition obtained using this algorithm.

5.1 Interpretation of the partition

In this section we give a criterion to globally measure the quality of a k-clusters partition and two criteria to interpret this partition according to a variable Y_j.

The quality of a partition is measured by the gain of homogeneity obtained by replacing the n objects of E by the k prototypes of P and normalized by the total homogeneity $f_E(G_E)$:

$$Q(P) = 1 - \frac{\Delta(P, L)}{f_E(G_E)} \tag{25}$$

This criterion takes its values between 0 and 1. It is equal to 1 when all the clusters are reduced to single objects or to identical objects. It is equal to 0 for the one-cluster partition E. Because this criterion decreases with the number of clusters it can only be used to compare two partitions having the same number of clusters. Because a k-clusters partition is better than another partition in k clusters if the criterion $\Delta(P, L)$ is smaller, this partition will be better if $Q(P)$ is bigger. For classical quantitative data

$$Q(P) = \frac{BSS}{TSS} = 1 - \frac{WSS}{TSS}$$

is called the part of inertia of E explained by P. The criterion $Q(P)$ measures in the same way the part of the homogeneity of E explained by P.

Similarly, we define the quality of the partition for each variable Y_j as:

$$Q_j(P) = 1 - \frac{\sum_{i=1}^{k} \tilde{f}_{C_i}(\hat{g}_i^j)}{\tilde{f}_E(\hat{g}_E^j)} \tag{26}$$

which is the part of the homogeneity of the variable Y_j explained by P. This criterion measures the *power of discrimination* of the variable Y_j to the partition P. Moreover because the quality of P, $Q(P)$, is a weighted average of the values $Q_j(P)$:

$$Q(P) = \sum_{j=1}^{p} \frac{\tilde{f}_E(\hat{g}_E^j)}{f_E(G_E)} Q_j(P) \tag{27}$$

this criterion also measures the importance of the variable Y_j in the construction of the partition. Finally the criteria $Q_j(P)$ must be compared to $Q(P)$.

5.2 Interpretation of the clusters

The quality of a cluster C_i of E is defined by:

$$Q(C_i) = 1 - \frac{f_{C_i}(G_i)}{f_{C_i}(G_E)} \tag{28}$$

This criterion measures the gain of homogeneity of the cluster C_i obtained when replacing the prototype G_E by the prototype G_i in the calculation of the homogeneity.

The contribution of a cluster C_i to the within-cluster homogeneity of P is defined by:

$$K(C_i) = \frac{f_{C_i}(G_i)}{\Delta(P, L)} \tag{29}$$

The sum of the k contributions is obviously 1.

A final criterion that is useful to interpret a cluster according to a variable Y_j is:

$$Q_j(C_i) = 1 - \frac{\tilde{f}_{C_i}(\hat{g}_i^j)}{\tilde{f}_{C_i}(\hat{g}_E^j)} \tag{30}$$

This criterion measures the part of discrimination power of the variable Y_j taken into account by C_i. In other words this criterion helps the user to find the variables which characterize the cluster C_i. Because the quality of the cluster, $Q(C_i)$, is a weighted average of the values $Q_j(C_i)$:

$$Q(C_i) = \sum_{j=1}^{p} \frac{\tilde{f}_{C_i}(\hat{g}_E^j)}{f_{C_i}(G_E)} Q_j(C_i) \tag{31}$$

the values of the criterion $Q_j(C_i)$ have to be interpreted by comparison with the value $Q(C_i)$. In other words we will consider that a variable Y_j characterizes the cluster C_i if $Q_j(C_i) > Q(C_i)$.

6 Application to the 60 meteorological stations in China

The criteria proposed to interpret a partition have been applied to an interval data set extracted from the Long-Term Instrumental Climatic Database of

the People's Republic of China. This Database contains among other variables the temperatures observed in 60 meteorological stations in China. In order to compare the 60 meteorological stations according to their temperatures, a natural representation of each station is to describe each month by an interval of minimal and maximal temperatures (see Table 1). Here we worked with the temperatures of the year 1988 and we built an interval data table of 60 rows and 12 columns corresponding to the 60 stations and the 12 months of the year.

On this data set we applied the first clustering method based on the Hausdorff distance in order to interpret the results with the criteria proposed in the previous section.

The number of clusters of the partitions was fixed to 5. The first algorithm was repeated 50 times with different initializations and the best 5-clusters partitions (i.e. minimizing $\Delta(P, L)$) were retained. We computed the quality criterion $Q(P)$ defined in (25) for the best partition and we found that this partition explained 64,33% of the homogeneity.

In order to know a little bit more about the clusters of the best partition we performed their quality $Q(C_i)$ ($\times$ 100) and their contribution $K(C_i)$ ($\times$ 100) defined in (28) and (29) (see Table 2).

i	size	$Q(C_i)$	$K(C_i)$
1	10	78.57	13.82
2	13	66.05	25.26
3	17	47.68	27.99
4	13	7.68	18.82
5	7	79.11	14.11

Table 2: Quality ($\times$ 100) and contribution ($\times$ 100) of the clusters C_i of the best partition

Because the homogeneity of a cluster naturally increases with the number of objects, this information has to be taken into account in the interpretation of these criteria. For this reason we compare here the cluster C_2 with the cluster C_4 both of which have 13 objects. We first notice that in Table 2 $Q(C_2) = 66,05\%$ whereas $Q(C_4) = 7,68\%$. We have seen that the quality of a cluster ($\times$ 100) is the percentage of gain of homogeneity obtained by replacing the global prototype G_E by the cluster prototype G_i. This criterion also measures the adequacy between the two prototypes G_E and G_i. This means that here the cluster C_2 is more "atypical" than C_4 or, in other words, that the objects of C_4 are more similar to the "average" object G_E. Concerning the contributions we have $K(C_2) = 25,26\%$ and $Q(C_4) = 18,82\%$. This means that cluster C_4 is more homogeneous than C_1.

Finally, in order to interpret the partition or the clusters according to the variables, we computed, for the best partition and for each variable Y_j, the criterion $Q_j(P)$ defined in (26) and, for the 5 clusters of the best partition, the criterion $Q_j(C_i)$ defined in (30) (see Table 6).

Variable	Q_j	$Q_j(C_1)$	$Q_j(C_2)$	$Q_j(C_3)$	$Q_j(C_4)$	$Q_j(C_5)$
January	**69.50**	**82.85**	**70.17**	**56.13**	**11.28**	**79.39**
February	**66.18**	**77.62**	63.61	**48.43**	**8.45**	**82.82**
March	**64.52**	**78.69**	64.75	44.60	**14.03**	78.31
April	**64.36**	72.31	**71.39**	40.78	6.36	**84.75**
May	61.68	69.56	**67.23**	37.89	7.34	**79.97**
June	53.36	77.41	63.91	32.88	5.55	66.87
July	46.31	74.50	50.00	31.79	4.73	62.26
August	47.19	75.76	53.99	28.04	3.24	46.54
September	61.10	78.18	65.58	28.16	6.85	76.59
October	**70.41**	**82.98**	**75.09**	**49.59**	7.20	**83.37**
November	**70.63**	**79.55**	**73.60**	**61.02**	4.22	**84.67**
December	**71.33**	**82.01**	62.55	**68.39**	**11.23**	**81.86**
Threshold	*64.33*	*78.57*	*66.05*	*47.68*	*7.68*	*79.11*

Table 3: Quality ($\times$ 100) of each variable (month), to the best partition and its clusters C_i

Because a variable Y_j can be considered as discriminant for a partition P if $Q_j(P) > Q(P)$ and for the cluster C_i if $Q_j(C_i) > Q(C_i)$, we have indicated in bold in Table 6 the values of $Q_j(p)$ and $Q_j(C_i)$ which verify these conditions. Finally we see, in Table 6, that globally, the less discriminant variables are June, July and August i.e. the three summer months.

7 Conclusion

The two methods shown above can be considered as the most suitable generalization of the classical DCA to the analysis of data expressed as intervals of real values. In fact, the generalization of DCA to such a kind of data could be formalized in a different way. The proposed approaches synthesize the case of classes represented by an element belonging to the same space as the objects to be clustered, as well as the case of classes that synthesize the characteristics of the elements of a class.

As we have seen, the search for the partitions of objects described by interval variables can be achieved by an optimization process as general as for the DCA on classical data. Like for DCA in the classical context, the best fitting between the representation and the allocation function is formalized by the criterion, which is a guarantee of coherent results. Even if the proposed

approaches appear different a priori, the results are quite similar because the model of the prototype, chosen in each approach, characterizes the same notion of coherence of a class.

In conclusion, we can remark that the choice of the kind of prototype usually guides the choice of the clustering method.

Acknowledgments: The second author would like to thank CNPq and FACEPE (Brazilian Agencies) for their financial support.

References

Bock, H.-H. (2001), Clustering algorithms and kohonen maps for symbolic data, *in* 'ICNCB Proceedings', Osaka, pp. 203–215.

Bock, H.-H. & Diday, E., eds (2000), *Analysis of symbolic data. Exploratory methods for extracting statistical information from complex data*, Studies in classification, data analysis and knowledge organisation, Springer Verlag, Heidelberg.

Celeux, G., Diday, E., Govaert, G., Lechevallier, Y. & Ralambondrainy, H. (1989), *Classification automatique des donnes*, Dunod.

Chavent, M. (2004), An hausdorff distance between hyper-rectangles for clustering interval data, *in* D. Banks, L. House, F. McMorris, P. Arabie & W. Gaul, eds, 'Classification, Clustering, and Data Mining applications', Springer Verlag, pp. 333–339.

Chavent, M., De Carvalho, F. A. T., Lechevallier, Y. & Verde, R. (2003), 'Trois nouvelles méthodes de classification automatique de données symboliques de type intervalle', *Rev. Stat. Appliquées* **LI**(4), 5–29.

Chavent, M. & Lechevallier, Y. (2002), Dynamical clustering of interval data. optimization of an adequacy criterion based on hausdorff distance, *in* K. Jajuga, A. Sokolowski & H.-H. Bock, eds, 'Classification, Clustering, and Data Analysis', Springer Verlag, Berlin, pp. 53–60.

De Carvalho, F. A. T. (1995), 'Histograms in symbolic data analysis.', *Annals of Operations Research* **55**, 289–322.

De Carvalho, F. A. T. & Souza, R. M. C. (1998), Statistical proximity functions of boolean symbolic objects based on histograms, *in* A. Rizzi, M. Vichi & H.-H. Bock, eds, 'Advances in Data Science and Classification', Springer Verlag, pp. 391–396.

De Carvalho, F. A. T., Verde, R. & Lechevallier, Y. (1999), A dynamical clustering of symbolic objcts based on a context dependent proximity measure.,

in H. e. a. Barcelar, ed., 'Proceedings of the IX International Symposium on Applied Stochastic Models and Data analysis', Universidade de Lisboa, pp. 237–242.

De Souza, R. M. C. R. & De Carvalho, F. A. T. (2004), 'Clustering of interval data based on city-block distances.', *Pattern Recognition Letters* **25**(3), 353–365.

Diday, E. (1971), 'La méthode des nuées dynamiques', *Rev. Stat. Appliquées* **19**(2).

Diday, E. (1988), The symbolic approach in clustering and related methods of data analysis: The basic choices, *in* H.-H. Bock, ed., 'Classification and related methods of data anlysis', North Holland, Amsterdam, pp. 673–684.

Diday, E. & Brito, P. (1989), Symbolic cluster analysis, *in* O. Opitz & H.-H. Bock, eds, 'Conceptual and Numerical Analysis of Data', Springer Verlag, Berlin, pp. 45–84.

Diday, E. & Simon, J. C. (1976), Clustering analysis, *in* K. Fu, ed., 'Digital Pattern Classification', Springer Verlag, pp. 47–94.

Gordon, A. D. (2000), An iteractive relocation algorithm for classifying symbolic data., *in* W. e. a. Gaul, ed., 'Data Analysis: Scientific Modeling and Practical Application', Springer Verlag, Berlin, pp. 17–23.

Huttenlocher, D. P., Klanderman, G. A. & Rucklidge, W. J. (1993), 'Comparing images using the hausdorff distance', *IEE Transaction on Pattern Analysis and Machine Intelligence* **15**, 850–863.

Nadler, S. B. J. (1978), *Hyperspaces of sets*, Marcel Dekker, Inc., New York.

Ralambondrainy, H. (1995), 'A conceptual version of the k-means algorithm.', *Pattern Recognition Letters* **16**, 1147–1157.

Rote, G. (1991), 'Computing the minimum hausdorff distance between two point sets on a line under translation', *Information Processing Letters* **38**, 123–127.

Verde, R., De Carvalho, F. A. T. & Lechevallier, Y. (2000), A dynamical clustering algorithm for multi-nominal data, *in* H. A. L. K. et al., ed., 'Data Analysis, Classification and Related methods', Springer Verlag, pp. 387–394.