REVUE DE STATISTIQUE APPLIQUÉE

M. CHAVENT
F. DE A. T. DE CARVALHO
Y. LECHEVALLIER
R. VERDE

Trois nouvelles méthodes de classification automatique de données symboliques de type intervalle

Revue de statistique appliquée, tome 51, n° 4 (2003), p. 5-29
<http://www.numdam.org/item?id=RSA_2003__51_4_5_0>

Rev. Statistique Appliquée, 2003, **LI** (4), 5-29

TROIS NOUVELLES MÉTHODES DE CLASSIFICATION AUTOMATIQUE DE DONNÉES SYMBOLIQUES DE TYPE INTERVALLE

M. CHAVENT[1], F. DE A. T. CARVALHO[2], Y. LECHEVALLIER[3], R. VERDE[4]

[1] *MAB-Mathématiques Appliquées de Bordeaux (UMR 5466), Université Bordeaux1, 351 cours de la libération, 33405 Talence cedex. chavent@math.u-bordeaux.fr*

[2] *CIn - Centro de Informática UFPE - Universidade Federal de Pernambuco, Av. Prof. Luiz Freire s/n - Cidade Universitária - CEP 50740-540 Recife-PE Brasil. fatc@cin.ufpe.br*

[3] *INRIA - Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau - Rocquencourt B.P. 105 - 78153 Le Chesnay Cedex, France yves.lechevallier@inria.fr*

[4] *Dip. Strategie Aziendali e Metodologie Quantitative - SUN - Seconda Università di Napoli, Piazza Umberto I, 81043 Capua, Italie. rosanna.verde@unina2.it*

RÉSUMÉ

Trois méthodes de classification automatique basées sur l'algorithme des Nuées Dynamiques sont présentées et comparées. Elles génèrent une partition d'un ensemble de données de type intervalle et utilisent des prototypes, des distances et des critères d'homogénéité différents. Une évaluation de ces trois approches sera faite à partir de deux jeux de données.

Mots-clés : *Partitionnement, Objets Symboliques, Variables Intervalles, Distances entre Intervalles.*

ABSTRACT

In this paper we present three partitioning methods based on a dynamical algorithm (like Nuées Dynamiques). These algorithms perform a partition of a set of interval data using different dissimilaries criterions as well as different kind of clusters representation (prototypes). The three approaches, here proposed, are compared on the basis of two data sets..

Keywords : *Partitioning, Symbolic Objects, Interval variables, Dissimilarities.*

1. Introduction

Dans ce papier nous proposons trois approches classificatoires d'un tableau de données symboliques [DID 88], [BOC 00] où les valeurs sont des intervalles. Dans le tableau 1 chaque colonne est une variable de type intervalle et chaque ligne est une description sous forme d'intervalles des relevés de température d'une station météorologique.

TABLEAU 1
Moyennes mensuelles des températures minimales et maximales journalières relevées dans les 60 stations météorologiques chinoises

Stations	températures mensuelles $[min : max]$ année 1988							
	Janvier	Février	Mars	Avril	...	Octobre	Novembre	Décembre
AnQing	[1,8:7,1]	[2,1:7,2]	[5,2:11,2]	[13,5:21,9]	...	[15,5:22,3]	[7,8:17,9]	[4,3:11,8]
BaoDing	[-7,1:1,7]	[-5,3:4,8]	[0,2:10,8]	[9,1:22,2]	...	[9,5:20,5]	[0,8:14]	[-3,9:5,2]
BeiJing	[-7,2:2,1]	[-5,9:3,8]	[-0,6:9,6]	[8,7:21,1	...	[8,7:20,4]	[1,5:12,7]	[-4,4:4,7]
BoKeTu	[-23,4:-15,5]	[-24:-14]	[-17,6:-4,7]	[-4,5:9,5]	...	[-4:8,9]	[-13,5:-4,2]	[-21,1:-13,1]
ChangChun	[-16,9:-6,7]	[-17,6:-6,8]	[-9,3:2,7]	[1,1:12,7]	...	[2,6:14,2]	[-7,9:2,2]	[-15,9:-7,2]
ChangSha	[2,7:7,4]	[3,1:7,7]	[6,5:12,6]	[12,9:22,9]	...	[15,3:22,8]	[7,6:19,6]	[4,1:13,3]
...	...	...	...	...	...	...	...	...
ZhiJiang	[2,7:8,4]	[2,7:8,7]	[6,2:13]	[12,1:23,5]	...	[14,4:22,7]	[8,2:20]	[5,1:13,3]

Nos trois approches utilisent un algorithme de type Nuées Dynamiques ([DID 71], [DIS 76], [DID 80] et [CEL 89]) avec des distances et des *prototypes* (noyaux, centroides) différents, le prototype d'une classe étant une modélisation de cette classe.

Dans la première approche, les prototypes sont des éléments de l'espace de représentation des objets à classer, c'est-à-dire un vecteur dont les coordonnées sont des intervalles. La distance entre un noyau et un individu ou plus généralement entre deux vecteurs d'intervalles est basée sur la *distance de Hausdorff* [CHA 97].

Dans la seconde et la troisième approche, les prototypes ne sont plus des vecteurs d'intervalles mais des vecteurs de distributions calculés à partir de systèmes de pondérations associés aux intervalles. Dans ce cas on parlera de *prototypes généralisés*. Dans la seconde approche, la distance utilisée est une distance classique entre distribution de type distance L_2. Dans la troisième approche, le prototype généralisé et les individus ne sont pas représentables dans le même espace de description. La mesure de comparaison utilisée n'est donc pas une dissimilarité mais une fonction de comparaison («matching»). Cette fonction est constituée de deux composantes car elle intègre non seulement l'écart entre deux intervalles mais également le système de pondération de ces deux intervalles.

Une comparaison numérique entre les deux approches a été réalisée sur deux jeux de données : les températures mensuelles de 60 stations chinoises (*Long-Term Instrumental Climatic Data Base of the People's Republic of China* http : //dss.ucar.edu/datasets/ds578.5/data/) et les «formes d'ondes» [BRE 84].

2. Notations et définitions

Une *variable intervalle* Y est une correspondance [AUB94] de l'ensemble E des objets dans $\mathfrak{R}$ qui vérifie la propriété suivante sur son graphe : pour tout individu $s \in E$ le sous-ensemble $[a, b] = Y(s)$ est un intervalle fermé de $\mathfrak{R}$. On notera $\mathfrak{J}$ l'ensemble des intervalles fermés de $\mathfrak{R}$.

Soit E un ensemble d'individus décrits par p variables intervalle $Y_1, ..., Y_j, ..., Y_p$. Chaque individu peut être modélisé par un objet symbolique [BOC 00] dont la description est le vecteur $\mathbf{x}_s$ d'intervalles de $\mathfrak{J}^p$ qui est l'*espace* de représentation des individus de E.

Finalement, notre tableau de données $(\mathbf{x}_s^j)_{n \times p}$ est constitué de n lignes représentant les n individus à classer et de p colonnes représentant les p variables, chaque case de ce tableau contenant un intervalle $\mathbf{x}_s^j = \left[a_s^j, b_s^j\right]$ fermé de $\mathfrak{R}$.

Un *prototype* G_i, associé à chacune des classes C_i, est un élément de l'*espace de représentation* Λ *des classes* qui peut être l'espace de représentation de E.

Une mesure de proximité D est une fonction positive ou nulle définie sur chaque couple d'éléments de $\mathfrak{J}^p$ (espace de représentation de E) et de Λ dont la valeur est d'autant plus petite que ces deux éléments sont «proches».

3. Schéma de l'algorithme de classification

Le schéma de l'algorithme de partitionnement est de type Nuées Dynamiques ([DID 71], [CEL 89]). Cet algorithme recherche une *partition* P^* de E en k classes non vides et un vecteur $L*$ de k prototypes $(G_1, ..., G_i, ..., G_k)$ qui représente, au mieux par rapport à un critère Δ, les k classes $(C_1, ..., C_i, ..., C_k)$ de la partition P^* :

$$\Delta(P^*, L^*) = \text{Min}\{\Delta(P, L) \,|P \in P_k, L \in \Lambda^k\}$$

avec P_k *l'ensemble des partitions* de E en k classes non vides et Λ *l'espace de représentation des prototypes.*

Ce critère Δ exprime l'adéquation entre la partition P et le vecteur L des k prototypes. Il est défini comme la somme sur toutes les classes C_i et sur tous les objets s de C_i des mesures de proximités $D(\mathbf{x}_s, G_i)$ entre chaque vecteur d'intervalles $\mathbf{x}_s$ et le prototype G_i de la classe C_i :

$$\Delta(P, L) = \sum_{i=1}^{k} \sum_{s \in C_i} D(\mathbf{x}_s, G_i) \quad C_i \in P, G_i \in \Lambda$$

L'algorithme procède alternativement par une étape de *représentation* suivie d'une étape d'*allocation*. Dans le cas où le prototype d'une classe est son centre de gravité on retrouve l'algorithme des *k-means* ou des centres mobiles.

Schéma de l'algorithme :

a) *initialisation* : On peut partir d'une partition $P = (C_1, ..., C_i, ..., C_k)$ choisie au hasard ou bien d'un ensemble de k prototypes $(G_1, ..., G_i, ..., G_k)$ tirés au hasard. Dans ce cas une étape d'affectation est réalisée de la manière suivante :

$C_i \leftarrow \varnothing$ pour $i = 1, .., k$

Pour $s = 1$ *à* n faire :

rechercher la classe C_l d'affectation de s, avec $l = \arg \min_{i=l,...,k} D(\mathbf{x}_s, G_i)$

$C_l \leftarrow C_l \cup \{s\}$

b) *étape de représentation* :

Pour $i = 1$ à k, on recherche le prototype G_i de L minimisant le critère $\sum_{s \in C_i} D(\mathbf{x}_s, G_i)$.

c) *étape d'allocation*

$test \leftarrow 0$

Pour $s = 1$ à n faire

m est la classe d'affectation du vecteur d'intervalles $\mathbf{x}_s$

rechercher sa nouvelle classe C_l d'affectation, avec $l = \arg \min_{i=1,...,k} D(\mathbf{x}_s, G_i)$

si $l \neq m$

$test \leftarrow 1$

$C_l \leftarrow C_l \cup \{s\}$ et $C_m \leftarrow C_m - \{s\}$

d) si $test = 0$ alors *stop*, autrement *aller* en *b)*

Comme le critère Δ est additif en fonction des k classes et des n objets de E, la recherche de la classe d'affection l de l'objet s dépend uniquement de la fonction de comparaison D entre le vecteur d'intervalles $\mathbf{x}_s$ et la description de prototype de cette classe C_l. Ainsi la décroissance du critère Δ est obtenue sous les conditions suivantes :

- unicité du choix de la classe d'affectation pour chaque individu de E;
- unicité du prototype G minimisant le critère $\sum_{s \in C} D(\mathbf{x}_s, G)$ pour toute classe C de E.

L'unicité de la classe d'affectation se résout facilement en prenant, en cas d'égalité des distances, la classe de plus petit indice.

Par contre l'existence et l'unicité du prototype est plus difficile à obtenir car elle est liée à la fonction de comparaison D. Cependant si l'espace Λ de représentation

des classes est identique à l'espace $\mathfrak{I}^p$ de représentation des objets de E alors cette résolution dépend uniquement de la distance entre vecteurs d'intervalles.

4. Définition des prototypes

La définition du meilleur prototype d'une classe est liée au choix de la fonction de comparaison D. Aussi un prototype de type différent sera associé à chacune des trois fonctions de comparaison.

• Dans la première méthode la fonction de comparaison D est une distance d_1 entre deux vecteurs d'intervalles $\mathbf{x}_1$ et $\mathbf{x}_2$ qui est basée sur la distance de Hausdorff :

$$d_1(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{p} d_H(\mathbf{x}_1^j, \mathbf{x}_2^j)$$

où d_H est la distance de Hausdorff entre les deux intervalles $\mathbf{x}_1^j = \left[a_1^j, b_1^j\right]$ et $\mathbf{x}_2^j = \left[a_2^j, b_2^j\right]$.

Le prototype G lié à cette fonction de comparaison sera décrit également par un vecteur d'intervalles.

• Dans la seconde méthode, on associe à chaque intervalle $\mathbf{x}_s^j = \left[a_s^j, b_s^j\right]$ du tableau de données une distribution discrète, notée $q(\mathbf{x}_s^j)$, toutes ces distributions ayant le même ensemble de définition. Ainsi, à chaque vecteur d'intervalles $\mathbf{x}_s$, on associe un vecteur $\mathbf{q}_s$ où les composantes $\mathbf{q}_s^j$ sont les vecteurs de poids des distributions $q(\mathbf{x}_s^j)$ associées aux intervalles $\mathbf{x}_s^j$. La fonction de comparaison D est donc une distance d_2 entre deux vecteurs $\mathbf{q}_1$ et $\mathbf{q}_2$, basée sur la distance L_2 de Minkowski :

$$d_2(\mathbf{q}_1, \mathbf{q}_2) = \sum_{j=1}^{p} d_M(\mathbf{q}_1^j, \mathbf{q}_2^j)$$

Le fait d'associer à chaque intervalle une distribution discrète revient à réaliser un codage du tableau de données par discrétisation des variables intervalles. Nous proposons de déterminer pour chaque variable un ensemble d'intervalles élémentaires disjoints calculés à partir des bornes supérieures et inférieures des intervalles observés sur cette variable. Après, pour chaque intervalle on associe une distribution dont les valeurs sont proportionnelles aux longueurs des intervalles élémentaires contenus dans cet intervalle.

Par exemple (figure 1), les objets s_1, s_2, s_3 et s_4 d'un ensemble E sont représentés par 4 intervalles. A partir de ces intervalles on construit la base $\{I_1, I_2, I_3, I_3, I_4, I_5\}$ qui est composée par 5 intervalles élémentaires. Le support de l'intervalle s_1 est l'ensemble $\{I_1, I_2\}$ avec une distribution $\{(I_1, \alpha), (I_2, 1-\alpha)\}$.

Ainsi, le prototype G associé à cette fonction de comparaison ne sera plus un vecteur d'intervalles mais un vecteur dont les composantes sont des distributions.

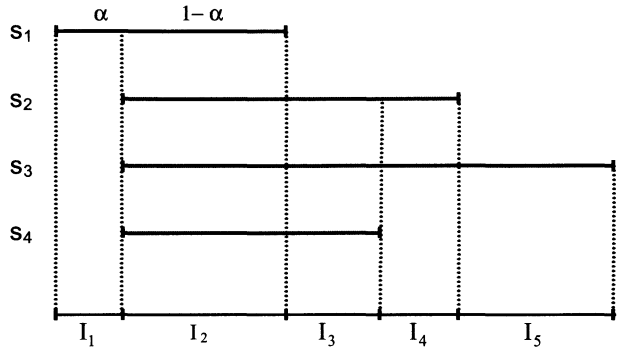

FIGURE 1
Exemple d'une base d'intervalles élémentaires

• Dans la troisième méthode, on décrit la relation entre la variable j et l'objet s, non pas par un intervalle $\mathbf{x}_s^j$ comme dans la première approche ou bien par le vecteur de poids $\mathbf{q}_s^j$ comme dans la seconde, mais par la paire : $\mathbf{p}_s^j = (\mathbf{x}_s^j, \mathbf{q}_s^j)$. La fonction de comparaison d_3 entre deux vecteurs $\mathbf{p}_1$ et $\mathbf{p}_2$ est la suivante :

$$d_3(\mathbf{p}_1, \mathbf{p}_2) = \sum_{j=1}^{p} d_{2c}(\mathbf{p}_1^j, \mathbf{p}_2^j) = \sum_{j=1}^{p} (d_{ic}(\mathbf{x}_1^j, \mathbf{x}_2^j) + d_{dc}(\mathbf{q}_1^j, \mathbf{q}_2^j))$$

où la dissimilarité d_3 est constituée de deux composantes : la dissimilarité d_{ic} qui est *indépendante du contexte* et la dissimilarité d_{dc} qui est *dépendante du contexte*. On dit qu'une dissimilarité est dépendante du contexte si le calcul de cette dissimilarité pour un couple d'éléments dépend des autres éléments de cet ensemble.

4.1. La distance de Hausdorff

4.1.1. Définition

La *distance de Hausdorff* [AUB94] d_H entre deux ensembles A_1 et A_2 fermés non vide de $\Re$ est définie par :

$$d_H(A_1, A_2) = \max \left\{ \sup_{x \in A_1} \inf_{y \in A_2} d(x, y), \sup_{y \in A_2} \inf_{x \in A_1} d(x, y) \right\}$$

où d est la distance euclidienne.

On déduit facilement de cette définition que la distance de Hausdorff entre deux intervalles $\mathbf{x}_1^j = \left[a_1^j, b_1^j\right]$ et $\mathbf{x}_2^j = \left[a_2^j, b_2^j\right]$ est égale à :

$$d_H(\mathbf{x}_1^j, \mathbf{x}_2^j) = \max(\left|a_1^j - a_2^j\right|, \left|b_1^j - b_2^j\right|)$$

Finalement, la comparaison entre deux vecteurs d'intervalles $\mathbf{x}_1$ et $\mathbf{x}_2$ est réalisée en effectuant une somme des distances de Hausdorff sur toutes les variables. D'où :

$$d_1(x_1, x_2) = \sum_{j=1}^{p} d_H(\mathbf{x}_1^j, \mathbf{x}_2^j) = \sum_{j=1}^{p} \max(\left|a_1^j - a_2^j\right|, \left|b_1^j - b_2^j\right|)$$

Lorsque tous les intervalles sont réduits à des points c'est-à-dire $\mathbf{x}_1$ et $\mathbf{x}_2 \in \Re^p$, cette distance correspond à la distance L_1.

4.1.2. Les prototypes

On cherche pour une classe C d'éléments de E un vecteur d'intervalles $G \in \Im^p$ qui minimise :

$$f(G) = \sum_{s \in C} d_1(\mathbf{x}_s, G) = \sum_{s \in C} \sum_{j=1}^{p} d_H(\mathbf{x}_s^j, G^j)$$

Ce critère f étant additif alors il suffit d'estimer pour chaque variable j, l'intervalle $G^j = [\alpha^j, \beta^j]$, prototype de la classe C, qui minimise :

$$\tilde{f}(G^j) = \sum_{s \in C} d_H(\mathbf{x}_s^j, G^j) = \sum_{s \in C} \max(\left|\alpha^j - a_s^j\right|, \left|\beta^j - b_s^j\right|)$$

Pour résoudre ce problème d'optimisation on pose :

$m_s^j = \dfrac{a_s^j + b_s^j}{2}$ le milieu de l'intervalle $[a_s^j, b_s^j]$

$l_s^j = \dfrac{b_s^j - a_s^j}{2}$ la moitié de sa longueur.

Et on pose μ^j et λ^j respectivement le milieu et la demi-longueur de l'intervalle $G^j = [\alpha^j, \beta^j]$.

En utilisant la propriété suivante définie pour x et y dans R :

$$\max(|x - y|, |x + y|) = |x| + |y|$$

la fonction $\tilde{f}$ à minimiser s'écrit :

$$\tilde{f}(G^j) = \sum_{s \in C} \max(\left|(\mu^j - \lambda^j) - (m_s^j - l_s^j)\right|, \left|(\mu^j + \lambda^j) - (m_s^j + l_s^j)\right|$$

soit

$$\tilde{f}(G^j) = \sum_{s \in C} \left|\mu^j - m_s^j\right| + \sum_{s \in C} \left|\lambda^j - l_s^j\right|$$

On retrouve donc deux problèmes d'optimisation bien connus : rechercher les valeurs $\hat{\mu}^j$ et $\hat{\lambda}^j$; $\hat{\mu}^j$ étant une solution de $\min_{\mu \in R} \sum_{s \in C} |\mu - m_s^j|$ et $\hat{\lambda}^j$ étant une solution de $\min_{\lambda \in R} \sum_{s \in C} |\lambda - l_s^j|$.

Les solutions $\hat{\mu}^j$ et $\hat{\lambda}^j$ sont donc respectivement les médianes :
de l'ensemble $\{m_s^j, s \in C\}$ des milieux des intervalles , $\left[a_s^j, b_s^j\right], s \in C$
de l'ensemble $\{l_s^j, s \in C\}$ de leurs demi-longueurs.
Donc la solution $\hat{G}^j = \left[\hat{\alpha}^j, \hat{\beta}^j\right]$ est l'intervalle $\left[\hat{\mu}^j - \hat{\lambda}^j, \hat{\mu}^j + \hat{\lambda}^j\right]$.

4.2. La distance L_2 de Minkowski

4.2.1. Définition d'une distribution discrète sur un intervalle x_s^j

On considère $\left\{x_1^j, \ldots, x_s^j, \ldots, x_n^j\right\}$ un ensemble de n intervalles de $\mathfrak{J}$. A partir de cet ensemble d'intervalles on construit un ensemble $I^j = \left\{I_1^j, \ldots, I_h^j, \ldots, I_{H_j}^j\right\}$ de H_j intervalles disjoints, dits *élémentaires*, vérifiant les propriétés suivantes :

i) $\bigcup_{h=1}^{H_j} I_h^j = \bigcup_{s=1}^{n} \mathbf{x}_s^j$;
ii) $I_h^j \cap I_{h'}^j = \varnothing$ si $h \neq h'$;
iii) $\forall h \quad \exists s \in E$ tel que $I_h^j \cap \mathbf{x}_s^j \neq \varnothing$;
iv) $\forall s \in E \quad \exists S_s^j \subset \{1, \ldots, H_j\}$ tel que $\bigcup_{h \in S_s^j} I_h^j = x_s^j$

Ils constituent une *base* de l'ensemble des intervalles $\left\{\mathbf{x}_1^j, \ldots, \mathbf{x}_s^j, \ldots, \mathbf{x}_n^j\right\}$ car chaque intervalle $\mathbf{x}_s^j$ est l'union d'un ensemble d'intervalles élémentaires disjoints. En pratique les bornes des intervalles élémentaires I_h^j de l'ensemble I^j sont construites à partir des bornes ordonnées des n intervalles $\left\{\mathbf{x}_1^j, \ldots, \mathbf{x}_s^j, \ldots, \mathbf{x}_n^j\right\}$. A chaque intervalle $\mathbf{x}_s^j$ on associe un vecteur $\mathbf{q}_s^j = (q_{s,1}^j, \ldots, q_{s,h}^j, \ldots, q_{s,H_j}^j)$ des poids défini par :

$$q_{s,h}^j = \begin{cases} \dfrac{\left|I_h^j\right|}{b_s^j - a_s^j} & \text{si } I_h^j \subseteq \mathbf{x}_s^j \\ 0 & \text{sinon} \end{cases}$$

ou $\left|I_h^j\right|$ est la longueur de l'intervalle I_h^j. Une *distribution modale* $q(\mathbf{x}_s^j)$ sur l'intervalle $\mathbf{x}_s^j$ est une distribution sur l'ensemble I^j des intervalles élémentaires inclus dans $\mathbf{x}_s^j$, définie par :

$$q(x_s^j) = \left\{(I_h^j, q_{s,h}^j) | h \in S_s^j\right\}, \text{ avec : } \sum_h q_{s,h}^j = 1$$

où l'ensemble $S_s^j = \left\{h | q_{s,h}^j > 0\right\}$ est appelé *support* de $q(x_s^j)$ et vérifie la propriété iv) de la base d'intervalles élémentaires.

4.2.2. Définition

La distance L_2 de Minkowski entre deux vecteurs de poids $\mathbf{q}_1$ et $\mathbf{q}_2$ associés à deux distributions définies sur l'ensemble I_j des intervalles élémentaires, est :

$$d_M(\mathbf{q}_1, \mathbf{q}_2) = \sum_{h=1}^{H_j} (q_{1,h}^j - q_{2,h}^j)^2$$

Finalement, la comparaison entre deux vecteurs $\mathbf{q}_1^j$ et $\mathbf{q}_2^j$ est réalisée en effectuant une somme des distances L_2 de Minkowski sur l'ensemble des variables :

$$d_2(\mathbf{q}_1^j, \mathbf{q}_2^j) = \sum_{j=1}^{p} d_M(\mathbf{q}_1^j, \mathbf{q}_2^j) = \sum_{j=1}^{p} \sum_{h=1}^{H_j} (q_{1,h}^j - q_{2,h}^j)^2$$

4.2.3. Les prototypes

Pour une classe C d'éléments de E on cherche, non plus un vecteur d'intervalles comme pour les prototypes définis avec la distance d_1 basée sur la distance de Hausdorff, mais un vecteur $G = (\mathbf{g}^1, \ldots, \mathbf{g}^j, \ldots, \mathbf{g}^p)$ dont chaque composante $\mathbf{g}^j$ est le vecteur des poids d'une distribution définie sur l'ensemble I^j des intervalles élémentaires sur la variable j. On recherche donc G qui minimise :

$$f(G) = \sum_{s \in C} d_2(\mathbf{q}_s, G) = \sum_{s \in C} \sum_{j=1}^{p} d_M(\mathbf{q}_s^j, \mathbf{g}^j)$$

où les composantes $\mathbf{q}_s^j$ du vecteur $\mathbf{q}_s$, sont les vecteurs des poids des distributions calculées sur les intervalles élémentaires associés aux vecteurs $\mathbf{x}_s^j$ (*cf.* 4.2.1.).

Comme le critère f est additif, il suffit de définir, sur chaque variable j, le vecteur de poids $\mathbf{g}^j = (g_1^j, \ldots, g_h^j, \ldots, g_{H_j}^j)$ qui minimise :

$$\tilde{f}(\mathbf{g}^j) = \sum_{s \in C} d_M(\mathbf{q}_s^j, \mathbf{g}^j) = \sum_{s \in C} \sum_{h=1}^{H_j} (q_{s,h}^j - g_h^j)^2$$

sous la contrainte :

$$\sum_{h=1}^{H_j} g_h^j = 1$$

Ce critère $\tilde{f}$ étant également additif, la solution sans la contrainte est évidente et on obtient :

$$\hat{g}_h^j = \frac{1}{|C|} \sum_{s \in C} q_{s,h}^j$$

et comme cette solution $\hat{\mathbf{g}}^j = (\hat{g}_1^j, \ldots, \hat{g}_h^j, \ldots, \hat{g}_{H_j}^j)$ vérifie la contrainte alors elle est la solution pour $\tilde{f}(\mathbf{g}^j)$.

Remarque. – Au lieu de prendre la distance L_2 pour réaliser les comparaisons entre les distributions discrètes nous pouvons choisir la distance du χ^2 entre ces distributions. Dans ce cas la distance d_2 est une somme de p distances du χ^2 qui correspond à la distance L_2 sur les vecteurs de poids $\dfrac{q_{s,h}^j}{\sqrt{\sum_{s \in E} q_{s,h}^j}}$ qui sont une pondération des distributions et le prototype G de chaque classe C est défini par $g_h^j = \dfrac{\sum_{s \in C} q_{s,h}^j}{|C| \sqrt{\sum_{s \in E} q_{s,h}^j}}$.

4.3. Fonction de comparaison ayant «deux composantes»

4.3.1. Définition

Dans cette approche, on utilise la fonction d_{2c} à deux composantes pour comparer deux paires $\mathbf{p}_1^j = (\mathbf{x}_1^j, \mathbf{q}_1^j)$ et $\mathbf{p}_2^j = (\mathbf{x}_2^j, \mathbf{q}_2^j)$ constituées d'un intervalle et d'un vecteur de poids :

$$d_{2c}(p_1^j, p_2^j) = d_{ic}(\mathbf{x}_1^j, \mathbf{x}_2^j) + d_{dc}(\mathbf{q}_1^j, \mathbf{q}_2^j)$$

où d_{ic} et d_{dc} sont deux dissimilarités l'une associée aux deux intervalles, l'autre associée aux deux vecteurs de pondération [DCS 98].

La dissimilarité d_{ic}, dite *indépendante du contexte*, entre deux vecteurs d'intervalles $\mathbf{x}_1^j = \left[a_1^j, b_1^j\right]$ et $\mathbf{x}_2^j = \left[a_2^j, b_2^j\right]$ est définie par :

$$d_{ic}(\mathbf{x}_1^j, \mathbf{x}_2^j) = \frac{\left|\overline{\mathbf{x}}_1^j \cap \overline{\mathbf{x}}_2^j \cap (\mathbf{x}_1^j \oplus \mathbf{x}_2^j)\right|}{\left|(\mathbf{x}_1^j \oplus \mathbf{x}_2^j)\right|}$$

avec $\mathbf{x}_1^j \oplus \mathbf{x}_2^j = [\min(a_1^j, a_2^j), \max(b_1^j, b_2^j)]$ et $\overline{\mathbf{x}}_1^j = \left]-\infty, a_1^j\right[\cup \left]b_1^j, +\infty\right[$ qui est le *complémentaire ensembliste* de l'intervalle $\mathbf{x}_1^j$ dans $\Re$. Dans ce cas, d_{ic} s'écrit :

$$d_{ic}(\mathbf{x}_1^j, \mathbf{x}_2^j) = \begin{cases} \dfrac{\left|\min(b_1^j, b_2^j) - \max(a_1^j, a_2^j)\right|}{\left|\max(b_1^j, b_2^j) - \min(a_1^j, a_2^j)\right|} & \text{si } \mathbf{x}_1^j \cap \mathbf{x}_2^j = \varnothing \\ 0 & \text{si } \mathbf{x}_1^j \cap \mathbf{x}_2^j \neq \varnothing \end{cases}$$

La dissimilarité d_{dc}, dite *dépendante du contexte*, entre les deux vecteurs de poids $\mathbf{q}_1^j$ et $\mathbf{q}_2^j$ est définie par :

$$d_{dc}(\mathbf{q}_1^j, \mathbf{q}_2^j) = \frac{1}{2}\left[\sum_{h \in S_1^j,\ h \notin S_2^j} q_{1,h}^j + \sum_{h \in S_2^j,\ h \notin S_1^j} q_{2,h}^j\right]$$

où S_1^j et S_2^j sont les deux supports définis par : $S_s^j = \left\{h | q_{s,h}^j > 0\right\}$ pour $s = 1, 2$.

Finalement, la comparaison entre deux vecteurs $\mathbf{p}_1$ et $\mathbf{p}_2$ dont les p composantes sont respectivement les paires $\mathbf{p}_1^j = (\mathbf{x}_1^j, \mathbf{q}_1^j)$ et $\mathbf{p}_2^j = (\mathbf{x}_2^j, \mathbf{q}_2^j)$ pour $j = 1 \ldots p$, est réalisée en effectuant une somme des fonctions d_{2c} à deux composantes sur toutes les variables :

$$d_3(p_1, p_2) = \sum_{j=1}^{p} d_{2c}(\mathbf{p}_1^j, \mathbf{p}_2^j) = \sum_{j=1}^{p} (d_{ic}(\mathbf{x}_1^j, \mathbf{x}_2^j) + d_{dc}(\mathbf{q}_1^j, \mathbf{q}_2^j))$$

4.3.2. Les prototypes

Dans cette troisième approche, le prototype G est un vecteur où chaque composante G^j est un couple (Γ^j, g^j) avec

– g^j est le vecteur de poids, égal à la moyenne des vecteurs de poids $\mathbf{q}_s^j$ des distributions associées aux vecteurs $\mathbf{x}_s$, $s \in C$. On retrouve le vecteur $\mathbf{g}^j$ des prototypes associés à la distance L_2 de Minkowsky soit $g_h^j = \frac{1}{|C|} \sum_{s \in C} q_{s,h}^j$ pour $h = 1, \ldots, H_j$.

– Γ^j est définie de deux manières différentes :

• (H1) Γ^j est l'intervalle défini comme la généralisation minimale de tous les intervalles $\mathbf{x}_s^j = [a_s^j, b_s^j]$ appartenant à la classe C. Dans ce cas Γ^j est égal à $[\min_{s \in C}(a_s^j), \max_{s \in C}(b_s^j)]$.

• (H2) Γ^j est le support de la distribution g^j associée à la classe C.

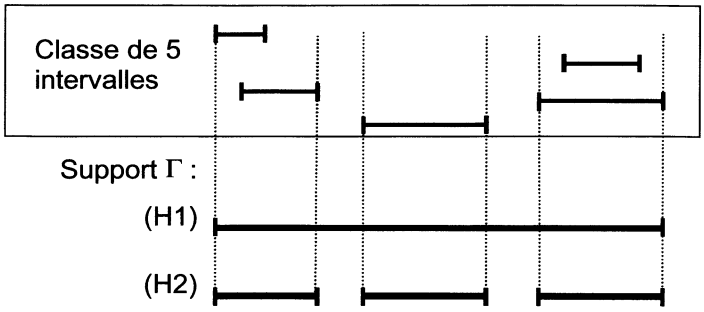

FIGURE 2
Exemple de supports pour une classe de 5 intervalles

PROPRIÉTÉ.
$\mathbf{g}^j$ est aussi le vecteur de poids qui vérifie la relation suivante :

$$2\sum_{s\in C} \mathrm{d}_{dc}(\mathbf{q}_s^j, \mathbf{g}^j) = \frac{1}{|C|}\sum_{s\in C}\sum_{s'\in C} \mathrm{d}_{dc}(\mathbf{q}_s^j, \mathbf{q}_{s'}^j)$$

Cette propriété se démontre en remarquant que :

$$\sum_{s\in C}\sum_{s'\in C} \mathrm{d}_{dc}(\mathbf{q}_s^j, \mathbf{q}_{s'}^j) = \sum_{s\in C}\sum_{s'\in C}\sum_{h\notin S_s^j,\ h\in S_{s'}^j} q_{s',h}^j$$

et, comme le support associé à $\mathbf{g}^j$ contient tous les support des objets s appartenant à la classe C alors on obtient :

$$\begin{aligned}\sum_{s\in C} \mathrm{d}_{dc}(\mathbf{q}_s^j, \mathbf{g}^j) &= \frac{1}{2}\sum_{s\in C}\sum_{h\notin S_s^j} g_h^j = \frac{1}{2|C|}\sum_{s\in C}\sum_{h\notin S_s^j}\sum_{s'\in C} q_{s',h}^j \\ &= \frac{1}{2|C|}\sum_{s\in C}\sum_{s'\in C}\sum_{h\notin S_s^j,\ h\in S_{s'}^j} q_{s',h}^j\end{aligned}$$

Cette propriété permet, de la manière qu'avec les deux précédentes distances, d'interpréter le vecteur de poids $\mathbf{g}^j$ comme le point «central» de la classe C.

Remarque. – Sachant que Γ^j n'est pas obligatoirement un intervalle on utilise entre un intervalle et Γ^j la distance indépendante du contexte suivante :

$$\mathrm{d}_{ic}(x_1^j, \Gamma^j) = \frac{\left|\overline{x}_1^j \cap (\bigcap_{s\in C} \overline{x}_s^j) \cap (x_1^j \oplus \bigcup_{s\in C} x_s^j)\right|}{\left|(x_1^j \oplus \bigcup_{s\in C} x_s^j)\right|}$$

avec : $x_1^j \oplus \bigcup_{s\in C} x_s^j = [\min(a_1^j, \min_{s\in C} a_s^j), \max(b_1^j, \max_{s\in C} b_s^j)]$ et $\bar{x}_1^j$ le *complémentaire ensembliste* de l'intervalle $\mathbf{x}_1^j$ dans $\mathfrak{R}$.

4.3.3. L'algorithme d'échange

Les prototypes ne sont pas définis, comme dans les deux approches précédentes, par l'optimisation du critère $f(C) = \sum_{s\in C} \mathrm{d}_3(p_s, G)$ mesurant l'adéquation entre le prototype G et l'ensemble des vecteurs $\mathbf{p}_s$ décrivant les objets de la classe C.

Dans ce cas nous proposons un ***algorithme d'échange*** qui réalise le changement de classe d'un objet s si les nouveaux prototypes, définis sur les deux classes modifiées par cet échange, font décroître le critère. Cet algorithme suit le schéma suivant :

a) *initialisation* : une partition $P = \{C_1, \ldots, C_i, \ldots, C_k\}$ est choisie au hasard

b) $test \leftarrow 0$

Pour chaque objet s de E l'*étape d'échange* (b.1 et b.2) suivante est réalisée :

b.1) C_m est la classe d'affectation du vecteur d'intervalles $\mathbf{x}_s$. Alors pour chaque C_i de la partition P on réalise un échange de s entre les deux classes C_m et C_i. A l'échange de s entre la classe C_m et la classe C_i on obtient :

- une nouvelle partition $P^{(i)} = (C_1^{(i)}, \ldots, C_l^{(i)}, \ldots, C_k^{(i)})$ avec

$C_i^{(i)} = C_i \cup \{s\}, C_m^{(i)} = C_m - \{s\}$ et $C_l^{(i)} = C_l$ pour $l \neq i, m$

- un nouveau vecteur $L^{(i)}$ des k prototypes des classes de $P^{(i)}$;

- le critère $\Delta(P^{(i)}, L^{(i)}) = \sum_{\ell=1}^{k} \sum_{u\in C_\ell^{(i)}} d_3(p_u, G_\ell^{(i)})$.

b.2) On recherche la nouvelle classe C_l de l'objet s telle que

$l = \arg \min_{i=1,\ldots,k} \Delta(P^{(i)}, L^{(i)})$.

Si $l \neq m$ alors on change l'objet s de classe, C_l est maintenant sa nouvelle classe d'affectation. D'où $test \leftarrow 1$ et $C_l \leftarrow C_l \cup \{s\}$ et $C_m \leftarrow C_m - \{s\}$.

c) si $test = 0$ alors *stop*, autrement *aller en b)*.

5. Applications

Une comparaison de trois méthodes proposées dans cet article a été effectuée à partir de deux applications.

La première application concerne un tableau de « moyennes » mensuelles de températures journalières observées dans 60 stations météorologiques chinoises (http://dss.ucar.edu/datasets/ds578.5/data). Une représentation naturelle de la température « mensuelle » d'une station est l'intervalle constitué par la moyenne des minima journaliers et la moyenne des maxima journaliers observés dans cette station durant

ce mois. En utilisant les mesures de l'année 1988 nous avons constitué un ensemble de 60 exemples décrits par 12 variables de type intervalle associées aux 12 mois de l'année 1988.

La deuxième application est issue du problème de reconnaissance des formes décrit dans le livre de Breiman *et al.* [BRE 84]. Chaque réalisation de l'échantillon est issue d'un modèle de génération utilisant deux formes parmi les trois formes h_1, h_2 et h_3 obtenues par $h_1(i) = \max\{6 - |i-7|, 0\}$, $h_2(i) = h_1(i+4)$, $h_3(i) = h_1(i+8)$ et un bruit gaussien.

Notre échantillon est constitué de 15000 réalisations des 50 sous-groupes des 3 classes *a priori* du problème. Chaque sous-groupe $C_{i,k} (i = 1, \ldots, 50, k = 1, 2, 3)$ comprend 100 exemples générés par le modèle suivant :
$x_m^j = u_m^i h_a(j) + (1 - u_m^i) h_b(j) + \varepsilon_m^j, j = 1, \ldots, 21, m = 1, \ldots, 100$ avec :

- ε_m^j est une réalisation de la variable aléatoire normale ε^j de moyenne nulle et de variance unitaire;
- u_m^i est une réalisation de la loi uniforme u^i dans l'intervalle $[(i-1)/50, i/50]$;
- a et b sont des paramètres dépendant de l'index k de la classe *a priori* (pour $k = 1$ alors $a = 1, b = 2$; $k = 2$ alors $a = 2$, $b = 3$; $k = 3$ alors $a = 1, b = 3$).

Le tableau de données intervalles, issu de notre échantillon, comprend 150 lignes qui sont les descriptions sous la forme d'intervalles des 50 sous-groupes des 3 classes *a priori*. Les vecteurs lignes de ce tableau sont constitués de 21 intervalles $\left[a_{i,k}^j, b_{i,k}^j\right]$, où $a_{i,k}^j = \min_{m \in C_{i,k}} x_m^j$ et $a_{i,k}^j = \max_{m \in C_{i,k}} x_m^j$.

Pour un nombre de classes fixé et pour chaque méthode proposée, 50 initialisations différentes sont réalisées et la meilleure solution est retenue. Ceci est effectué pour un nombre de classe compris entre 2 et 7. On note AL1 la méthode basée sur la distance de Hausdorff, AL2 celle basée sur la distance de Minkowski et AL3 celle basée sur la distance à deux composantes et utilisant l'hypothèse H2 dans la détermination du prototype.

Nous proposons de comparer ces trois méthodes à partir des tableaux suivants :

- Le premier tableau contient le taux de confusion entre les partitions obtenues par les méthodes AL1 et AL3, pour un nombre de classes fixé. Afin d'alléger la présentation des résultats nous avons choisi de comparer uniquement les méthodes AL1 et AL3 qui sont les plus différentes en termes de prototype et de distance. Le taux de confusion est calculé à partir du tableau de contingence entre les deux partitions. Si les partitions obtenues sont assez semblables alors il est facile de réaliser la correspondance entre les classes des deux partitions. Par ce fait le taux de confusion correspond au pourcentage des individus qui n'appartiennent pas aux classes mises en correspondances dans les deux partitions.

- Le second tableau contient un indicateur de qualité de la partition P_k qui est égal à $(\Delta(P_E, L_E) - \Delta(P_k, L_k))/\Delta(P_E, L_E)$ où P_E est la partition grossière et L_E le prototype associé à E ;

• Les derniers tableaux contiennent des statistiques sur la dispersion des solutions obtenues en fonction des différentes réitérations des algorithmes pour la première et la dernière méthodes.

5.1. Classification des stations météorologiques en Chine

Les résultats des trois méthodes ont été comparés, en utilisant le tableau des données des températures mensuelles de l'année 1988 de 60 stations chinoises, à partir des indicateurs proposés dans le paragraphe précédent.

Dans le tableau 2 on observe que les deux partitions les plus proches sont les partitions en 3 et 6 classes.

TABLEAU 2
Taux de confusion entre les partitions obtenues par les méthodes AL1 et AL3

Nombre de classes	AL1-AL3
2	8.33 %
3	1.67 %
4	13.33 %
5	21.67 %
6	5.00 %
7	15.00 %

Dans le tableau 3 le gain de qualité le plus important est obtenu en passant de 2 à 3 classes pour les trois méthodes étudiées.

TABLEAU 3
Indice de qualité des meilleures partitions obtenues par les 3 méthodes

Nombre de classes	AL1 Distance de Hausdorff	AL2 Distance L_2	AL3 Distance à deux composantes
2	39.29	23.48	28.45
3	50.40	39.32	43.45
4	57.45	49.50	52.12
5	64.33	57.29	56.13
6	67.85	61.24	62.75
7	70.82	65.33	64.92

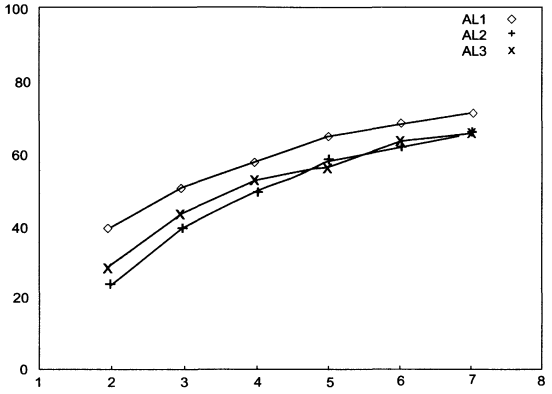

FIGURE 3
Représentation graphique des valeurs de l'indice de qualité des partitions

Le tableau 4 donne pour les partitions en 2 à 7 classes obtenues avec la méthode AL1, les statistiques élémentaires (min, max...) des 50 valeurs du critère Δ correspondant aux 50 initialisations différentes de l'algorithme. La figure 4 est une représentation graphique de ces résultats où l'axe horizontal représente le nombre de classes et l'axe vertical les valeurs du critère. La courbe représente l'évolution des meilleures solutions du critère. Le losange indique la moyenne des valeurs du critère sur les 50 réinitialisations de l'algorithme. La boite est centrée sur la valeur moyenne et a une longueur égale à deux écart-types. Elle permet donc de visualiser la variabilité des solutions. Dans chaque boite un trait relie la courbe (la meilleure solution) à la solution la plus éloignée de la solution optimale.

Cette figure permet de souligner, pour la méthode AL1, la faible variabilité des solutions du problème à deux classes. Pour les partitions ayant plus de deux classes la variabilité est plus importante mais elle touche essentiellement les valeurs maximales et elle ne s'accroît pas en fonction du nombre de classes.

TABLEAU 4
Statistiques sur les valeurs du critère pour l'algorithme AL1

	Valeur du critère (50 essais)			
Nombre de classes	Min	Max	Moyenne	Ecart-type
2	2677.70	2677.70	2677.70	0.00
3	2187.60	2524.75	2250.31	80.41
4	1876.80	2310.30	1965.30	128.89
5	1573.40	2057.20	1713.92	114.16
6	1417.80	1807.65	1565.00	124.06
7	1287.10	1659.05	1435.46	93.55

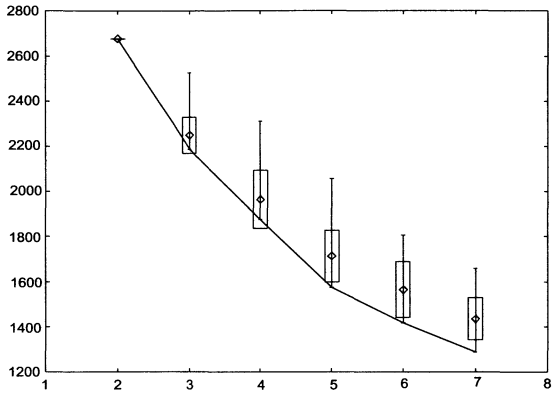

FIGURE 4
Représentation graphique des statistiques sur les valeurs du critère pour l'algorithme AL1

La figure 5 montre de la même manière, l'évolution et la variabilité des solutions de l'algorithme AL3 en fonction du nombre de classes. On note que cette variabilité est plus faible que pour l'algorithme AL1 (figure 4) et qu'elle est très faible quand le nombre de classe est égal à 2, 5 et 7. On en conclut que les solutions obtenues avec la méthode AL3 sont moins sensibles au choix de la partition initiale que celles obtenues avec AL1. Cependant si le nombre de classes est égal à 4 ou 6 alors l'algorithme AL3 donne quelques solutions très éloignées de la meilleure solution ce qui justifie des réinitialisations plus nombreuses dans AL3 que dans AL1.

TABLEAU 5
Statistiques sur les valeurs du critère pour l'algorithme AL3

	Valeur du critère (50 essais)			
Nombre de classes	Min	Max	Moyenne	Ecart-type
2	6.250	6.365	6.289	0.093
3	4.955	5.374	5.095	0.341
4	4.195	5.343	4.419	1.022
5	3.845	3.965	3.876	0.103
6	3.264	4.106	3.425	0.760
7	3.074	3.400	3.158	0.280

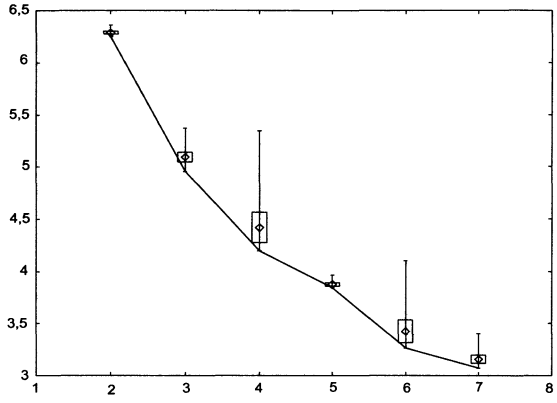

FIGURE 5
Représentation graphique des statistiques sur les valeurs du critère pour l'algorithme AL3

Le tableau 6a donne les prototypes associés aux cinq classes de la partition obtenue avec la méthode AL3, pour la variable «juillet». Ces prototypes sont des distributions sur les 9 intervalles élémentaires définis par l'algorithme. Ainsi pour la classe 1, l'intervalle élémentaire [23.60 : 25.79] a pour pondération 0.076.

Le tableau 6b donne les prototypes associés aux cinq classes de la partition obtenue avec AL1. Ces prototypes sont des intervalles comme par exemple l'intervalle de température [22.3 : 29.9] associé à la classe 1.

TABLEAU 6a
Prototype de la partition en 5 classes obtenues avec AL3, pour la variable «juillet»

Prototype méthode AL3	Classes				
	1	2	3	4	5
Intervalles élémentaires	Distributions sur les intervalles élémentaires				
[10,80 :17,19]	0.002	**0.099**	**0.115**	0.000	0.000
[17.20 :21.39]	0.087	0.288	0.373	0.000	0.030
[21.40 :23.69]	0.077	0.175	0.285	**0.000**	0.176
[23.60 :25.79]	**0.076**	0.170	0.194	0.044	**0.277**
[25.80 :27.29]	0.316	**0.109**	0.071	0.151	0.188
[27.30 :29.69]	0.231	0.124	**0.012**	0.295	0.223
[29.70 :32.59]	0.294	0.028	0.000	**0.357**	**0.099**
[32.60 :34.79]	**0.147**	0.007	0.000	0.122	0.010
[34.80 :36.90]	0.029	0.000	0.000	0.028	0.000

TABLEAU 6b
Prototype de la partition en 5 classes obtenues avec AL1 pour la variable «juillet»

Prototype	Classes				
méthode AL1	1	2	3	4	5
Intervalles	[22.32 :29.85]	[16.52 :28.88]	[16.77 :25.94]	[25.58 :34.02]	[24.09 :32.46]

Les partitions étant très semblables, il a été possible d'effectuer une correspondance entre les classes des deux partitions et de représenter sur un même graphique (Figure 6) les prototypes obtenus avec AL3 (des distributions) et les prototypes obtenus avec AL1 (des intervalles).

On observe sur cette figure que les classes 2 et 3 contiennent les stations météorologiques localisées dans des régions plutôt froides en juillet et les classes 4 et 5 contiennent les stations situées dans des régions plutôt chaudes.

On note également sur ce graphique que les résultats des deux algorithmes concordent puisque les prototypes de la méthode AL1 recouvrent en partie les intervalles élémentaires de poids élevé dans les prototypes de l'algorithme AL3 ou encore qu'ils ne recouvrent aucun intervalle élémentaire de poids faible.

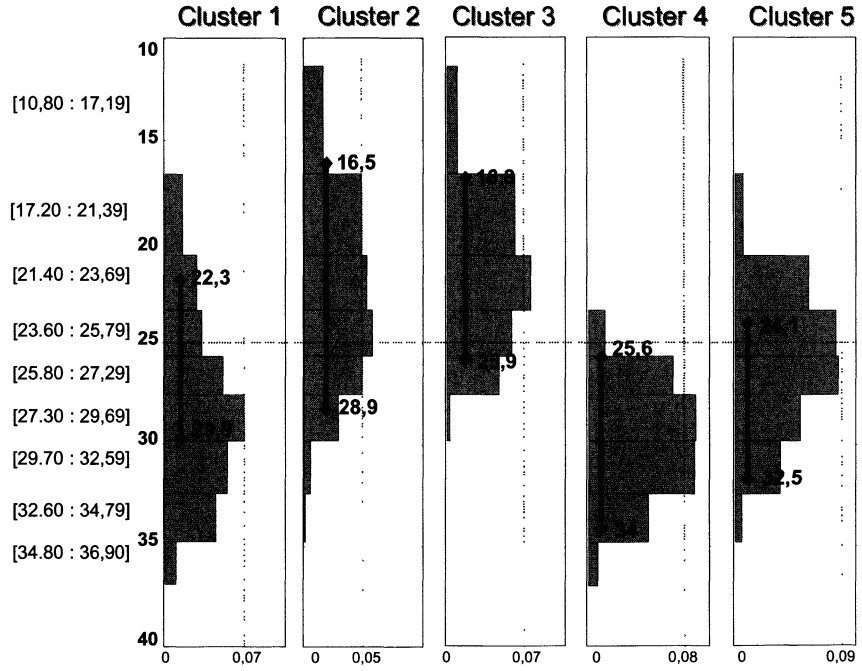

FIGURE 6
Les prototypes entre les méthodes AL1 et AL3 pour la variable «juillet»

En conclusion, pour chaque variable (ici le mois de juillet), le prototype obtenu par l'algorithme AL1 est un intervalle de température qui résume au mieux les intervalles de température des stations appartenant à chacune des classes. Par contre l'algorithme AL3 représente chaque classe par une distribution sur un découpage du domaine de variation des températures en Juillet.

5.2. Classification des 150 modèles élémentaires

Dans cette deuxième application, réalisée à partir d'un ensemble de courbes artificielles décrites dans le livre de Breiman *et al.* [BRE 84], les résultats des méthodes AL1, AL2 et AL3 ont été comparés avec les même outils que l'application précédente.

TABLEAU 7
Pourcentage de confusion entre les partitions obtenues par les méthodes AL1 et AL3

Nombre de classes	AL1-AL3
2	0.00 %
3	0.00 %
4	2.67 %
5	4.67 %
6	10.00 %
7	10.00 %

On note dans le tableau 7 que les partitions en 2 et 3 classes obtenues par AL1 et AL3 sont identiques et que celles en 4 et 5 classes sont très semblables.

TABLEAU 8
Indice de qualité des partitions obtenues par les 3 méthodes

Nombre de classes	AL1 Distance de Hausdorff	AL2 Distance L_2	AL3 Distance à deux composantes
2	29.20	25.68	32.30
3	40.71	36.46	46.05
4	45.22	41.10	50.96
5	50.04	44.80	56.41
6	52.92	47.63	59.36
7	55.22	49.08	61.83

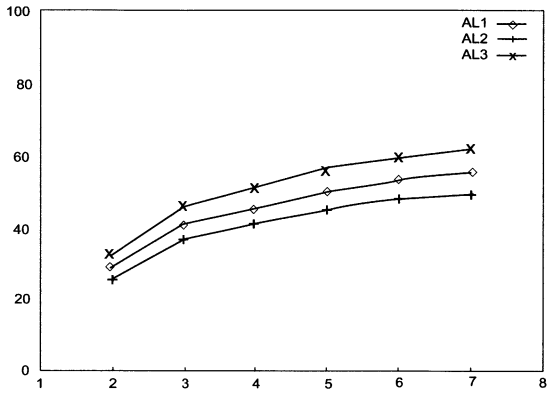

FIGURE 7
Représentation graphique des valeurs de l'indice de qualité des partitions

On observe sur la figure 7 que les courbes de qualité des partitions obtenues avec les trois méthodes évoluent de la même manière en fonction du nombre de classes. Ceci confirme les valeurs du taux d'erreur contenues dans le tableau 7.

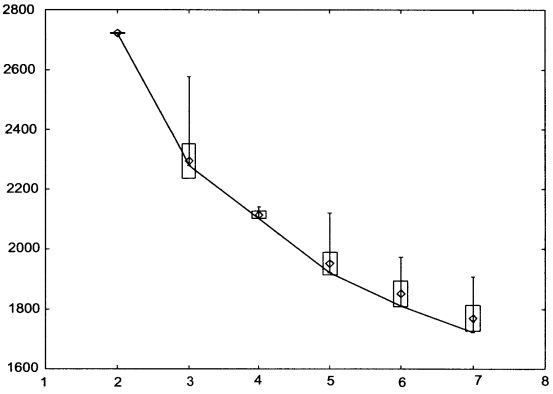

FIGURE 8
Représentation graphique des statistiques sur les valeurs du critère pour l'algorithme AL1

La figure 8 montre que la solution obtenue par AL1 est indépendante de l'initialisation pour un nombre de classes égal à 2 ou 4. Par contre pour un nombre de classes fixé à 3 on retrouve quelques solutions ayant une valeur de l'indice de qualité très éloignée de la valeur optimale.

Avec l'algorithme AL3 (figure 9) la variabilité des 50 solutions est très faible pour un nombre de classes fixé à 3. On retrouve ici le nombre de formes à partir desquelles a été généré le tableau de données.

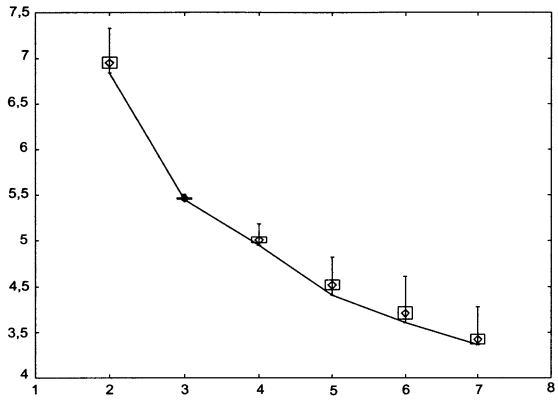

FIGURE 9
Représentation graphique des statistiques sur les valeurs du critère pour l'algorithme AL3

Les prototypes, déterminés par AL1 et AL3, de la classe 1 (taux de confusion nul) sont représentés dans la figure 10. Les bornes supérieures et inférieures des intervalles correspondant aux prototypes obtenus avec AL1 sont reliés par des courbes et les distributions correspondant aux prototypes obtenus avec AL3 sont représentés par des barres ayant des graduations plus ou moins foncées en fonction des poids associés à chaque intervalle élémentaire.

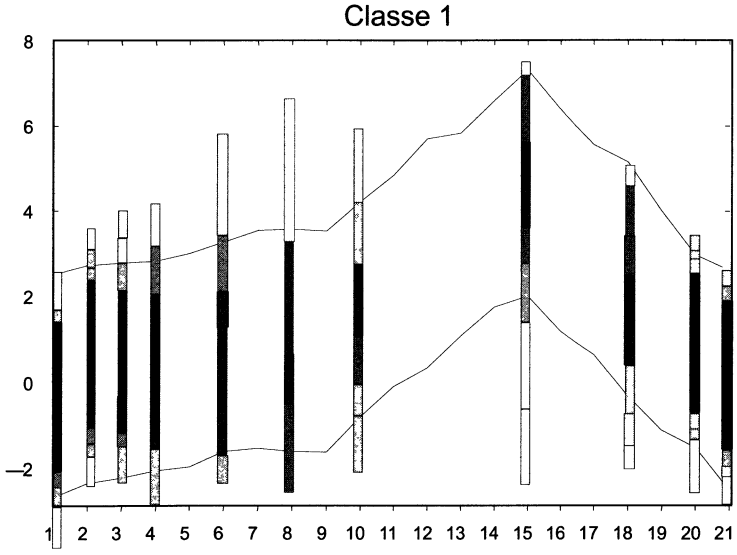

FIGURE 10
Visualisation des prototypes de la classe 1

On note là encore la concordance entre les résultats des méthodes AL1 et AL3.

6. Conclusion

Dans le cadre classique, il existe une grande diversité de méthodes de classification de type nuées dynamiques, en fonction de la distance choisie, cette distance dépendant elle même souvent du type des données (quantitatives, qualitatives...). On retrouve dans cet article, à travers les trois méthodes de classification proposées cette diversité pour des données non plus classique mais décrites par des intervalles.

On a pu remarquer que pour les deux applications auxquelles ont été appliquées les trois méthodes AL1, AL2 et AL3, les résultats changeaient peu en fonction de l'algorithme choisi. Ainsi, le choix d'une méthode de classification est un problème difficile pour l'utilisateur. Par contre la détermination du modèle de prototype est plus naturelle car l'utilisateur a toujours une idée *a priori* sur la représentation des classes (intervalles ou distributions). De ce fait le choix du type de prototype pourrait conditionner le choix par l'utilisateur d'une des trois méthodes de classification proposées ici.

Bibliographie

[AUB94] AUBIN J.-P. (1994), *Initiation à l'analyse appliquée*, Masson.

[BRE 84] BREIMAN L., FRIEDMAN J., OLSHEN R., STONE C. (1984), *Classification and regression trees*, Chapman Hall.

[BOC 00] 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.

[CEL 89] CELEUX G., DIDAY E., GOVAERT, G. LECHEVALLIER Y., RALAMBONDRAINY H. (1989), *Classification Automatique des Données.* Bordas, Paris.

[CHA 97] CHAVENT M. (1997), *Analyse des Données Symboliques. Une méthode divisive de classification.* Thèse de l'Université de PARIS-IX *Dauphine.*

[DCA 94] DE CARVALHO F.A.T. (1994), Proximity coefficients between Boolean symbolic objects, in *New Approaches* in *Classification and Data Analysis*, Diday *et al.* (Eds.), Springer Verlag, Heidelberg, 387-394.

[DCA 98] DE CARVALHO F.A.T. (1998), Extension based proximities between Boolean symbolic objects, in *Data Science, Classification and Related Methods*, Hayashi, C. *et al.* (eds.), Springer-Verlag, Tokyo, 370-378.

[DCS 98] DE CARVALHO F.A.T., SOUZA R. M. C. (1998), Statistical proximity functions of Boolean symbolic objects based on histograms. In : Rizzi, A., Vichi, M., Bock, H.-H. (Eds.) : *Advances in Data Science and Classification*, Springer-Verlag, Heidelberg, 391-396

[DCA 00] DE CARVALHO F.A.T., ANSELMO C.A.F., SOUZA, R.M.C.R. (2000), Symbolic approach to classify large data sets, in : *Data Analysis, Classification, and Related Methods*, Kiers, H.A.L. *et al.* (Eds.), Springer, 375-380.

[DVL 99] DE CARVALHO F.A.T, VERDE, R. et LECHEVALLIER Y. (1999), A dynamical clustering of symbolic objects based on a context dependent proximity measure. In : Bacelar-Nicolau, H., Nicolau, F.C. and Janssen, J. (Eds.) : *Proc. IX International Symposium - ASMDA'99*. LEAD, Univ. de Lisboa, 237–242.

[DID 71] DIDAY E. (1971), Le méthode des Nuées dynamiques, in *Revue de Statistique Appliquée*, 19, 2, 19-34.

[DID 88] DIDAY E. (1988), The symbolic approach in clustering and related methods of data analysis : The basic choice. In Proc. IFCS-97, Bock, H.-H. (Eds), Springer-Verlag, Heidelberg, 673-684.

[DID 98] DIDAY E. (1998), Symbolic Data Analysis : a Mathematical Framework and Tool for Data Mining, in *New Andvances in Data Science and Classification*, Rizzi, A. *et al.* (eds.), Springer -Verlag, Heidelberg, 409-416.

[DIS 76] DIDAY E. AND SIMON J. C. (1976), Clustering Analysis. In : Fu, K. S. (Eds.) : *Digital Pattern Recognition*. Springer-Verlag, Heidelberg, 47-94.

[DID 80] DIDAY E., GOVAERT G., LECHEVALLIER Y. et SIDI J. (1980), Clustering in pattern recognition, NATO Advanced study Institute on Digital Image Processing and Analysis, Bonas. Available at INRIA-Rocquencourt.

[ICY 94] ICHINO, M., YAGUCHI H. (1994), Generalized Minkowsky Metrics for Mixed Feature Type Data Analysis. *IEEE Transactions System, Man and Cybernetics* 24, 698-708.

[IYD 96] ICHINO M., YAGUCHI H., DIDAY E. (1996), A fuzzy symbolic pattern classifier, in : *Ordinal and Symbolic Data Analysis*, Diday, E. *et al.* (Eds.), Springer, 92-102.

[LEC 97] LECHEVALLIER Y. (1997), Classification non supervisée, in *Statistique et méthodes neuronales*, Thiria, Lechevallier *et al.* (Eds.), Dunod, Chap. 10, 171-189.

[LER 79] LEREDDE H. (1979), La méthode des pôles d'attraction - La méthode des pôles d'agrégation. Thèse de Diplôme de docteur de 3e cycle. Université Paris VI, 106-116.

[MIC 80] MICHALSKI R. S. (1980), Knowledge acquisition through conceptual clustering : A theoretical framework and an algorithm for partitioning data into conjunctive concepts. A special Issue on Knowledge Acquisition and Induction. Policy Analysis and Information Systems, 3.

[MDS 81] MICHALSKI R. S., DIDAY E., STEPP R. E. (1981), A recent advance in data analysis : Clustering Objects into classes characterized by conjunctive concepts. In : Kanal L. N. and Rosenfeld A. (Eds.) : *Progress in pattern recognition*. North-Holland, 33-56.

[VDL 00] VERDE R., DE CARVALHO F.A.T., LECHEVALLIER Y. (2000), A Dynamical Clustering Algorithm for Multi-Nominal Data. In : H.A.L. Kiers, J.-P. Rasson, P.J.F. Groenen and M. Schader (Eds.) : *Data Analysis, Classification, and Related Methods*, Springer-Verlag, Heidelberg, 387-394.

[VDL 01] VERDE R., DE CARVALHO F.A.T., LECHEVALLIER Y. (2001), A dynamical clustering algorithm for symbolic data. Tutorial *Symbolic Data Analysis*, GfKl Conference, Munich.