

# STATISTIQUE ET ANALYSE DES DONNÉES

GILDAS BROSSIER

## **Représentation ordonnée des classifications hiérarchiques**

*Statistique et analyse des données*, tome 5, n° 2 (1980), p. 31-43.

[http://www.numdam.org/item?id=SAD\\_1980\\_\\_5\\_2\\_31\\_0](http://www.numdam.org/item?id=SAD_1980__5_2_31_0)

© Association pour la statistique et ses utilisations, 1980, tous droits réservés.

L'accès aux archives de la revue « Statistique et analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/legal.php>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

REPRESENTATION ORDONNEE DES CLASSIFICATIONS  
HIERARCHIQUES

par

Gildas BROSSIER

UER des Sciences et Techniques. Université de Haute-Bretagne.  
Rennes II

I - INTRODUCTION

Le résultat d'une classification hiérarchique sur un ensemble E à analyser peut prendre différentes formes: soit un ensemble de partitions ordonnées totalement par la relation d'inclusion, soit une matrice de distance (ou de similarité) vérifiant l'inégalité ultramétrique, soit un graphe arborescent reliant les différents éléments de E, du type arbre de longueur minimale, soit enfin un arbre hiérarchique appelé aussi dendrogramme. Ces différentes représentations sont équivalentes et on sait aisément passer de l'une à l'autre.

Cependant, la forme la plus utilisée, parce qu'elle est la plus facilement interprétable, est l'arbre hiérarchique sous sa forme habituelle (voir par exemple la figure 1).

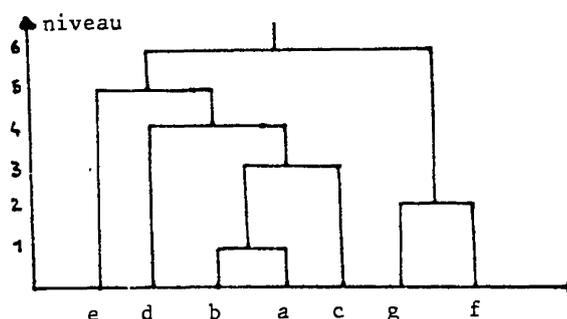


figure 1

Cette représentation appelle trois remarques:

- Seul l'axe vertical du graphique est porteur d'informations. En effet la distance entre deux éléments  $x$  et  $y$  est égale au niveau auquel ces deux éléments fusionnent: par exemple  $\delta(d,c)=4$ .
- Par conséquent l'axe horizontal n'a aucune signification. Les objets  $y$  sont alignés dans un ordre arbitraire dû en général à la fantaisie du programme de classification.

Par exemple, sur l'axe horizontal de la figure 1, "d" est beaucoup plus proche de "e" que de "c" alors que  $\delta(d,e)=5$  et  $\delta(d,c)=4$  (et donc "d" est en réalité plus proche de "c" que de "e").

c) Un arbre de classification hiérarchique correspondant à une matrice de distances ultramétriques admet plusieurs représentations possibles liées à l'ordre des objets sur l'axe horizontal. Toutes ces représentations étant bien sûr équivalentes.

Par exemple sur la figure 1 si nous permutons la classe (b,a) et la classe (c) nous obtenons une autre représentation de la même classification hiérarchique et donc l'interprétation qui en est faite ne devrait pas être modifiée. D'ailleurs, tenant compte de cette réalité, JAMBU dans [3], a suggéré de représenter un arbre de classification comme étant un mobile suspendu.

Donc si la relation naturelle initiale entre les éléments de E était une distance vérifiant l'inégalité ultramétrique il n'y aurait pas lieu de s'occuper de cet axe horizontal. Mais en fait, la matrice de distance ne vérifie jamais cette inégalité, et donc, sa représentation par un arbre de classification n'est qu'une approximation de la structure liant les objets.

Dans ces conditions, le problème se pose de choisir parmi toutes les représentations possibles d'un arbre de classification celle qui, au sens de la relation d'ordre qu'elle induit sur l'axe horizontal, est la meilleure au regard des distances initiales.

Ainsi l'arbre de la figure 1 peut se mettre sous la forme de celui de la figure 2, qui lui est rigoureusement équivalente, mais où les points ont été réarrangés de façon à faire apparaître un ordre "naturel" qui facilite l'interprétation et n'apparaissait pas sur la figure 1.

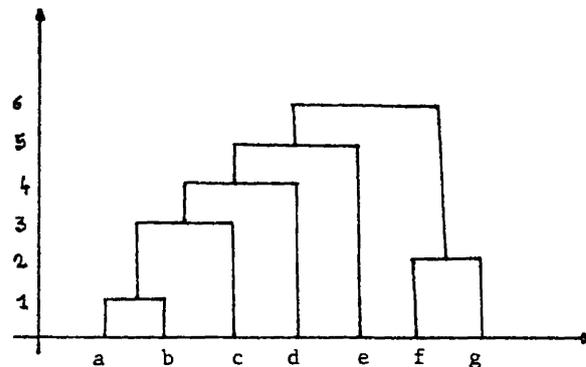


figure 2

C'est ce que font S. SATTAN et A. TVERSKY dans [8] quand, présentant des exemples illustrant une méthode de classification, ils représentent leurs arbres de façon à faire apparaître un ordre interprétable. Toutefois ils ne disent rien sur la façon dont ils obtiennent cet ordre.

## II - ORDRE COMPATIBLE AVEC UNE MATRICE DE DISTANCE

Dans la suite nous considérerons que deux ordres sont équivalents s'ils sont symétriques l'un par rapport à l'autre (par exemple, l'ordre 5,4,3,2,1 est équivalent à l'ordre 1,2,3,4,5). Ainsi le terme ordre désignera indifféremment un ordre et son symétrique, et nous noterons  $x_0y_0z_0...t_0$  l'ordre  $x,y,z,...,t$  ou l'ordre  $t,...,z,y,x$ .

*Définition.*

Soit  $D$  une matrice de distance sur  $E$ , et  $O$  un ordre sur  $E$ ,  $D$  et  $O$  seront dits compatibles si et seulement si:

$$\forall x, y, z \in E \quad x \notin y \notin z \iff d(x, y) \leq d(x, z) \text{ et } d(z, y) \leq d(z, x)$$

Cette définition est équivalente à la suivante:

$$x \notin y \notin z \iff d(x, z) = \max(d(x, y), d(y, z))$$

On en déduit les propriétés suivantes:  $a, b \in \{x, y, z\}$

*Propriété 1.*

Soit  $D$  une matrice de distance ne comportant pas deux distances égales, alors s'il existe un ordre  $O_D$  compatible avec  $D$ ,  $O_D$  est unique (à son symétrique près).

Pour le démontrer il suffit de considérer deux ordres  $O$  et  $O'$  et un triplet qui les différencie.

*Propriété 2.*

Soient  $D$  et  $O_D$  compatibles, alors  $O_D$  est tel que ses points extréma, noté  $a$  et  $b$  vérifient:

$$d(a, b) = \max(d(x, y)), (x, y) \in E \times E$$

Démonstration: si non, soient  $u$  et  $v$  tels que  $d(u, v) > d(a, b)$  alors les triplets  $(a, u, b)$  et  $(a, v, b)$  sont incompatibles avec  $D$ .

*Propriété 3.*

Soient  $D$  et  $O_D$  compatibles, " $a$ " un point extrémum de  $O_D$ , alors  $O_D$  est engendré par les valeurs croissantes de  $d(a, x)$ .

La propriété 3 nous donne un moyen simple de bâtir  $O_D$ , quand il existe. Cependant, pour une matrice  $D$  quelconque il n'existe pas toujours un ordre  $O_D$  compatible avec  $D$ .

Nous allons maintenant nous intéresser au cas particulier où  $D$  est ultramétrique.

### III - ORDRES COMPATIBLES AVEC UNE DISTANCE ULTRAMETRIQUE

On sait que lorsqu'une distance vérifie l'inégalité ultramétrique, tous les triangles sont isocèles, et la base est le plus petit des côtés. De ce fait on peut mettre la définition de compatibilité sous la forme simplifiée suivante:

*Définition.*

Un ordre et une distance ultramétrique sont compatibles, si et seulement si, pour tout triplet, le sommet du triangle ainsi formé, est un point extrémum de l'ordre (restreint au triplet).

Ce qui nous amène à énoncer la propriété suivante qui va nous permettre de construire des représentations ordonnées des classifications hiérarchiques.

*Propriété:*

Pour une matrice de distance ultramétrique  $\Delta$ , les deux propositions suivantes sont équivalentes:

- 1) l'ordre  $O_\Delta$  est compatible avec la distance ultramétrique.

- 2) la matrice ultramétrique est représentable par un arbre hiérarchique plan (sans croisement) dont les éléments sont ordonnés selon l'ordre  $O_{\Delta}$ .

Démonstration:

Soit un triplet d'éléments de E vérifiant l'inégalité ultramétrique, on peut toujours, en les appelant x, y et z, mettre cette inégalité sous la forme:

$$(1) \delta(x,y) = \delta(y,z) > \delta(x,z)$$

La distance  $\delta$  peut alors se représenter par l'arbre "en mobile" de la figure 3.

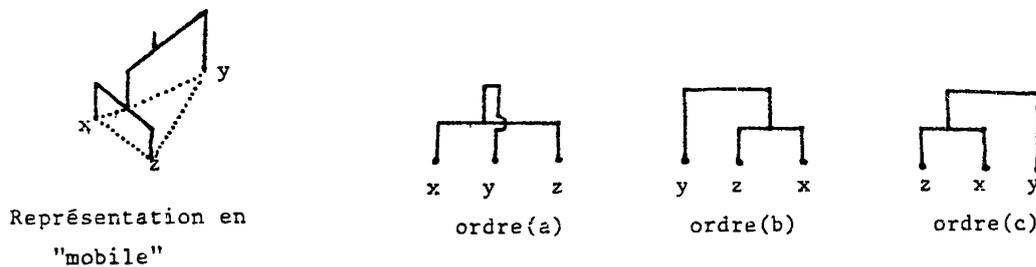


figure 3

Les trois points (x,y,z) admettent trois ordres possibles (en considérant toujours que deux ordres symétriques sont équivalents).

- a)  $x \theta y \theta z$
- b)  $y \theta z \theta x$
- c)  $z \theta x \theta y$ , parmi lesquels seuls b) et c) sont compatibles avec 1)

1°) si l'ordre n'est pas compatible il ne conduit pas à une représentation hiérarchique sans croisement : en effet l'ordre a) qui n'est pas compatible, oblige, après avoir fusionnés les points x et z qui sont les plus proches, à un croisement pour représenter les distances  $\delta(x,y)$  et  $\delta(y,z)$ .

2°) si un ordre est compatible il conduit à un arbre sans croisement: en effet, on agrège au premier niveau deux éléments qui sont toujours contigus, car d'après la définition de compatibilité, le sommet du triangle isocèle, qui s'agrège au deuxième niveau, est un point extrémum.

#### IV - DENOMBREMENT DES ORDRES COMPATIBLES AVEC UNE DISTANCE ULTRAMETRIQUE

Cette identité entre la notion de représentation d'un arbre, et la notion de compatibilité d'un ordre et d'une distance ultramétrique va nous permettre par un calcul simple de dénombrement de situer les contraintes respectives.

Si on a n éléments:

- a) Il existe  $n!$  ordres possibles sur n éléments
- b) Tous ces ordres étant symétriques deux à deux, nous n'avons que  $n!/2$  ordonnances possibles.

- c) Il existe  $n!(n-1)!/2^{n-1}$  hiérarchies possibles sur  $n$  objets:  $n(n-1)/2$  façons de choisir le premier couple à agréger, puis  $(n-1)(n-2)/2, \dots$  etc.
- d) Chaque arbre admet  $2^{n-1}$  représentations possibles. En effet, chacun des  $(n-1)$  noeuds admet 2 représentations.
- e) Compte tenu de la symétrie, chaque arbre, ou donc, chaque matrice de distance ultramétrique admet  $2^{n-2}$  ordres compatibles.
- f) A l'inverse chaque ordre admet  $(n-1)!$  arbres compatibles avec lui.

Il est donc intéressant de noter que si une matrice de distance non ultramétrique mais dont tous les éléments sont distincts, n'admet pas toujours un ordre compatible, et que s'il existe il est unique, à l'inverse une matrice de distance ultramétrique admet toujours des ordres compatibles, et il y en a  $2^{n-2}$ .

Il s'agit donc de trouver parmi toutes les représentations possibles d'un arbre, celle qui va nous donner un ordre significatif au regard des distances initiales.

Supposons que l'on ait trois éléments avec la matrice de distance non ultramétrique suivante:

	a	b	c
a	0	1	3
b	1	0	4
c	3	4	0

Quelque soit la méthode de classification ascendante hiérarchique utilisée nous allons obtenir la hiérarchie suivante avec sa matrice ultramétrique associée:

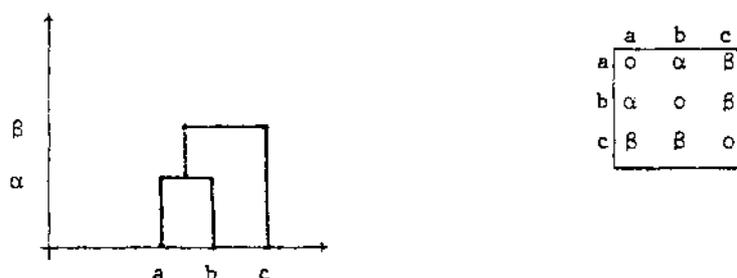


figure 4

Les valeurs de  $\alpha$  et de  $\beta$  dépendant de la méthode utilisée: par exemple  $\alpha=1, \beta=3$  pour le Single Linkage,  $\alpha=1, \beta=3,5$  pour l'average Linkage,  $\alpha=1, \beta=4$  pour le complete Linkage, ...etc.

Il est évident que la représentation suivante (figure 5) de l'arbre, rigoureusement équivalente à la précédente, mais utilisant l'autre ordre compatible, représente mieux la matrice de distance initiale si l'on interprète l'axe horizontal. En effet "a" y est plus proche de "c" que ne l'est "b". (En fait nous avons choisi ici un exemple extrême où les données sont unidimensionnelles).

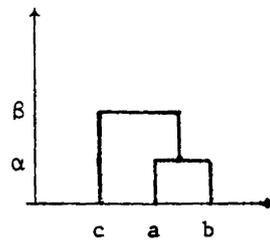


figure 5

V - LES DIFFERENTS PROBLEMES

Le problème peut alors se présenter sous les trois formes suivantes:

- a) Etant donné une matrice de distance (ou de similarité) sur les  $n$  éléments de  $E$  et un arbre de classification, rechercher un ordre compatible avec l'arbre et représentant au mieux les distances (ou similarités) initiales.
- b) Etant donné une matrice de distance (ou de similarité) sur  $E$  rechercher une relation d'ordre sur les éléments de  $E$  puis un arbre hiérarchique compatible avec cet ordre et représentant au mieux les données.
- c) Avec les mêmes données qu'en b) rechercher simultanément une relation d'ordre et une classification hiérarchique compatibles entre elles.

Nous allons proposer un algorithme simple apportant une solution possible à la problématique a) qui nous paraît être la plus intéressante, et nous évoquerons quelques éléments de solution aux problématiques b) et c).

VI - ORDONNANCEMENT DES NOEUDS TERMINAUX D'UN ARBRE HIERARCHIQUE

En données on a donc, d'une part, la matrice  $D_{n \times n}$  des distances (ou similarités) entre les  $n$  éléments de  $E$  et, d'autre part, une classification hiérarchique obtenue par un algorithme quelconque et représentée par son arbre hiérarchique ou sa matrice de distance ultramétrique  $\Delta_{n \times n}$ . On utilise un algorithme descendant qui parcourt l'arbre de son sommet jusqu'à ses noeuds terminaux. A chaque étape lorsqu'un noeud se présente on l'oriente dans un sens ou dans l'autre à l'aide d'une règle de décision liée aux distances entre noeuds.

Ayant  $n-1$  noeuds, et l'orientation du premier étant arbitraire on a  $n-2$  décisions à prendre correspondant aux  $2^{n-2}$  ordres compatibles possibles.

On peut schématiser la procédure par le schéma de l'étape  $i$ :



A l'étape  $i$ ,  $k$  éléments sont déjà ordonnés. Pour passer à l'étape  $i+1$ , afin d'obtenir l'ordre sur les  $k+1$  points il suffit d'ordonner les 2 éléments nouveaux.

Le test de décision:

On peut imaginer tous les tests possibles cependant un calcul simple est préférable sachant que:

- il n'y a que deux possibilités
- en général, on n'optimise pas globalement par une suite de  $n-2$  optimisations locales

La règle choisie est celle de la comparaison au plus proche voisin illustrée par le schéma suivant:

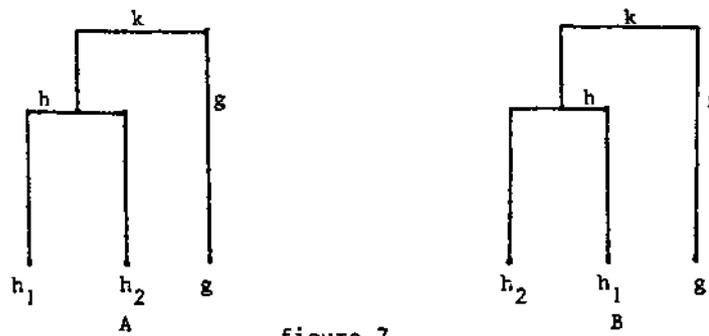


figure 7

On oriente le noeud  $h$  qui se subdivise en  $h_1$  et  $h_2$  en fonction des distances de  $h_1$  et  $h_2$  avec leur plus proche voisin qui est le noeud  $g$  avec lequel ils fusionnent dans le noeud  $k$ .

Le noeud  $k$  ayant déjà été orienté, par exemple  $h$  à gauche et  $g$  à droite, on oriente le noeud  $h$  de façon à générer soit l'ordre  $h_1 h_2 g$  (solution A) soit l'ordre  $h_2 h_1 g$  (solution B).

Si  $d(h_1, g) < d(h_2, g)$  on a l'ordre B

$d(h_2, g) < d(h_1, g)$  on a l'ordre A

En cas d'égalité on peut trancher en comparant à la branche de gauche s'il y en a une.

Un cas particulier à examiner est celui où le noeud  $g$  s'est déjà divisé à un niveau supérieur au niveau  $h$ .

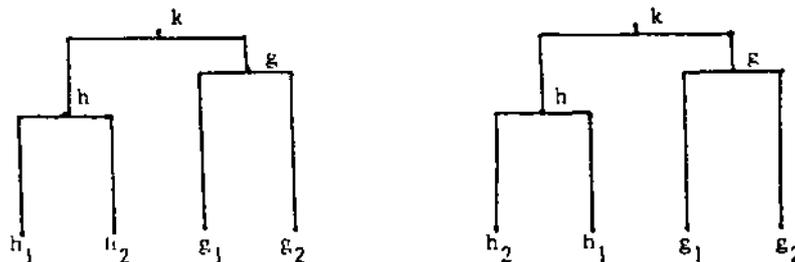


figure 8

Dans ce cas la branche  $g$  a été orientée à une étape précédente pour donner naissance à l'ordre  $hg_1g_2$  (par exemple) et on va orienter le noeud  $h$  non plus par rapport à  $g$  mais à  $g_1$ . En effet, la distance  $d(h_1, g)$  n'a aucun sens, ces 2 noeuds, n'étant pas au même niveau.

*Mise en oeuvre de l'Algorithme:*

Si on rajoute un tel algorithme à la suite d'un algorithme de classification hiérarchique, il faut prévoir un tableau supplémentaire. Le tableau ORDRE de taille  $n$ , qui contient la liste des noeuds dans l'ordre obtenu. C'est ce tableau qui est modifié à chaque étape de l'algorithme (on lui rajoute 1 élément à chaque étape) et qui constitue le résultat final.

D'autre part il faut pouvoir accéder aux distances entre deux noeuds d'un même niveau. Comme ces distances ont forcément été déjà calculées lors de la construction de l'arbre il est alors prudent de prévoir de les stocker sur disque afin de pouvoir y accéder sans nouveau calcul.

Au total il faut :  $n-2$  étapes nécessitant chacune la lecture de 2 distances sur disque et 1 comparaison.

## VII - PROPRIÉTÉ DE L'ALGORITHME

*Définition:*

On appelle méthode de classification ascendante hiérarchique (C.A.H.) homogène, toute C.A.H. telle que la distance de l'élément  $q$ , réunion de  $h$  et de  $l$ , à un élément  $i$  quelconque vérifie:

$$\min(d_{ih}, d_{il}) < d_{iq} < \max(d_{ih}, d_{il})$$

Intuitivement cela veut dire que le représentant  $q$  du groupe  $h, l$  reste dans le convexe formé par  $h$  et  $l$ .

Il est aisé de vérifier que cette condition est satisfaite par un certain nombre de méthodes parmi lesquelles on peut citer:

- Single Linkage, ou ultramétrique inférieure maxima
- Average Linkage, ou méthode de la moyenne
- Complete Linkage, ou diamètre minimum

*Proposition.*

Soit  $D$  une matrice de distance admettant un ordre compatible  $O_D$ ,

Soit  $\Delta$  une matrice de distance ultramétrique déduite de  $D$  par une C.A.H. homogène,

Alors

$\Delta$  admet  $O_D$  comme ordre compatible, et l'algorithme d'ordonnement conduit à l'ordre  $O_D$

*Démonstration.*

Montrons d'abord que  $\Delta$  et  $O_D$  sont compatibles.

Au niveau 0 de départ de la C.A.H., rangeons les  $n$  éléments à classer dans l'ordre  $O_D$  compatible avec la matrice  $D$  de dimension  $n \times n$ .

Au niveau  $k$  on suppose les  $n-k$  éléments à classer rangés selon l'ordre  $O_{D_k}$  et cet ordre  $O_{D_k}$  compatible avec la matrice de distance  $D_k$  entre les  $n-k$  éléments.

Pour passer au niveau  $k+1$ :

1. On agrège les 2 éléments  $h$  et  $l$  les plus proches. Ces deux éléments sont forcément contigus sinon  $O_{D_k}$  et  $D_k$  ne seraient pas compatibles.

2. On obtient l'élément  $q$  réunion de  $h$  et de  $l$ .

$O_{D_{k+1}}$  se déduit de  $O_{D_k}$  en remplaçant les deux éléments  $h$  et  $l$  par le seul élément  $q$ , les autres restant inchangés. Pour montrer que  $O_{D_{k+1}}$  est compatible avec  $D_{k+1}$  il suffit de montrer

que les inégalités sont respectées sur les seuls triangles mettant en jeu  $q$ , ce qui est immédiat si la C.A.H. est homogène. En effet les inégalités qui étaient vraies pour  $h$  et  $l$  le demeurent puisque quelque soit l'élément  $i$  on a :

$$\min(d_{ih}, d_{il}) \leq d_{iq} \leq \max(d_{ih}, d_{il})$$

On bâtit ainsi une suite de  $n-1$  ordres  $O_{D_i}$  compatibles avec  $n-1$  matrices de distance  $D_i$  de dimension  $(n-i) \times (n-i)$ .

Il en découle que l'on vient de construire, sans croisement, l'arbre hiérarchique correspondant à la matrice  $\Delta$ , à partir de l'ordre  $O_D$  et donc, d'après la propriété 4, cet ordre est compatible avec  $\Delta$ .

Montrons maintenant que connaissant  $D$  et  $\Delta$  l'algorithme d'ordonnement conduit à l'ordre  $O_D$ . Reconsidérons la suite que nous avons bâti:  $O_{D_i}$  et  $D_i$ ,  $i = 0, \dots, n-2$ .

- au niveau  $n-2$ : nous avons 2 éléments que l'on peut ranger selon l'ordre  $O_{D_{n-2}}$  compatible avec  $D_{n-2}$ .

- au niveau  $n-k$ : nous supposons que les  $k$  éléments sont rangés dans l'ordre  $O_{D_{n-k}}$  compatible avec la matrice  $D_{n-k}$ .

- au niveau  $n-(k+1)$ : l'algorithme va ordonner les  $k+1$  éléments selon un ordre  $O$ . Montrons que  $O$  coïncide avec  $O_{D_{n-(k+1)}}$ .

$O_{D_{n-(k+1)}}$  se déduit de  $O_{D_{n-k}}$  en séparant l'élément  $q$  en deux éléments  $h$  et  $l$  contigus et ordonnés. Les autres éléments restant dans le même ordre car l'arbre est sans croisement.

$O$  se déduit de  $O_{D_{n-k}}$  par l'algorithme de la même façon. De plus  $h$  et  $l$  sont ordonnés par rapport à leur plus proche voisin  $i$  de façon à ce que le triplet  $(h, l, i)$  soit compatible avec  $D_k$ . Or nous n'avons que deux possibilités pour  $O$ :  $O_1$  qui est retenu par l'algorithme et  $O_2$  qui est rejeté.

Si l'ordre  $O_{D_{n-(k+1)}}$  était l'ordre  $O_2$  il serait incompatible avec  $D_{k-1}$  pour le triplet  $(h, l, i)$ . Donc  $O$  et  $O_{D_{n-(k+1)}}$  sont identiques.

On reconstitue donc par l'algorithme la chaîne  $O_{D_k}, D_k, k = 0, \dots, n-2$  en aboutissant ainsi à l'ordre  $O_D$ .

#### VIII - CAS D'UN ORDRE SUIVI D'UNE HIERARCHIE.

Nous nous plaçons maintenant dans la problématique b), celle où l'on va d'abord ordonner les éléments de  $E$  avant de bâtir une hiérarchie compatible avec cet ordre.

Dans le cas particulier où l'ensemble  $E$  est muni à priori d'une structure d'ordre il existe des méthodes bien connues de classification sous contrainte (voir {2}, {5}, {6}).

Dans le cas général, qui nous intéresse, où  $E$  n'est muni d'aucune structure d'ordre à priori, il faut d'abord rechercher une ordonnance significative sur  $E$ . Une solution simple consiste à utiliser des méthodes de représentation spatiale des données telles que l'analyse factorielle ou le multidimensional scaling en se restreignant à un espace de dimension 1.

Il est aisé alors d'enchaîner avec une des méthodes classiques de hiérarchie sous contrainte. Cependant, cette approche n'est intéressante que si on veut privilégier la relation d'ordre par rapport à la classification.

Ce type de liaison entre un modèle de multidimensional scaling et classification hiérarchique a été développé en détail par Kruskal dans {4}.

## IX - RECHERCHE SIMULTANEE D'UNE HIERARCHIE ET D'UN ORDRE

Il est tentant de chercher à obtenir simultanément une hiérarchie et un ordre compatibles entre eux plutôt que d'obtenir le second sous la contrainte du premier. Une méthode possible consiste à obtenir la hiérarchie par une méthode divisive en y incluant une procédure de décision permettant d'ordonner les classes formées.

Les algorithmes descendants sont peu utilisés en classification car ils sont à priori long et coûteux. Cependant, comme le note Williams dans {10}, ils ne manquent pas d'intérêts. Le principal est que, en général, on est intéressé par des partitions contenant peu de classes, ce qui revient à attendre les dernières étapes d'une hiérarchie ascendante pour avoir un résultat intéressant. Or ce sont justement dans ces dernières étapes que les algorithmes ascendants perdent de leur efficacité, en terme d'optimisation.

A l'inverse avec un algorithme descendant, on peut s'arrêter dès que l'on a obtenu un nombre de classes suffisant, et c'est dans ces premières étapes que l'optimisation est la plus efficace. Il est aisé alors d'adjoindre après chaque division une procédure de décision analogue à celle définie en VI de façon à ordonner les classes au fur et à mesure de leur formation. La différence entre ces deux approches est que les deux procédures (ordre et classification) s'exécutent alternativement et non successivement.

## X - UN EXEMPLE D'APPLICATION

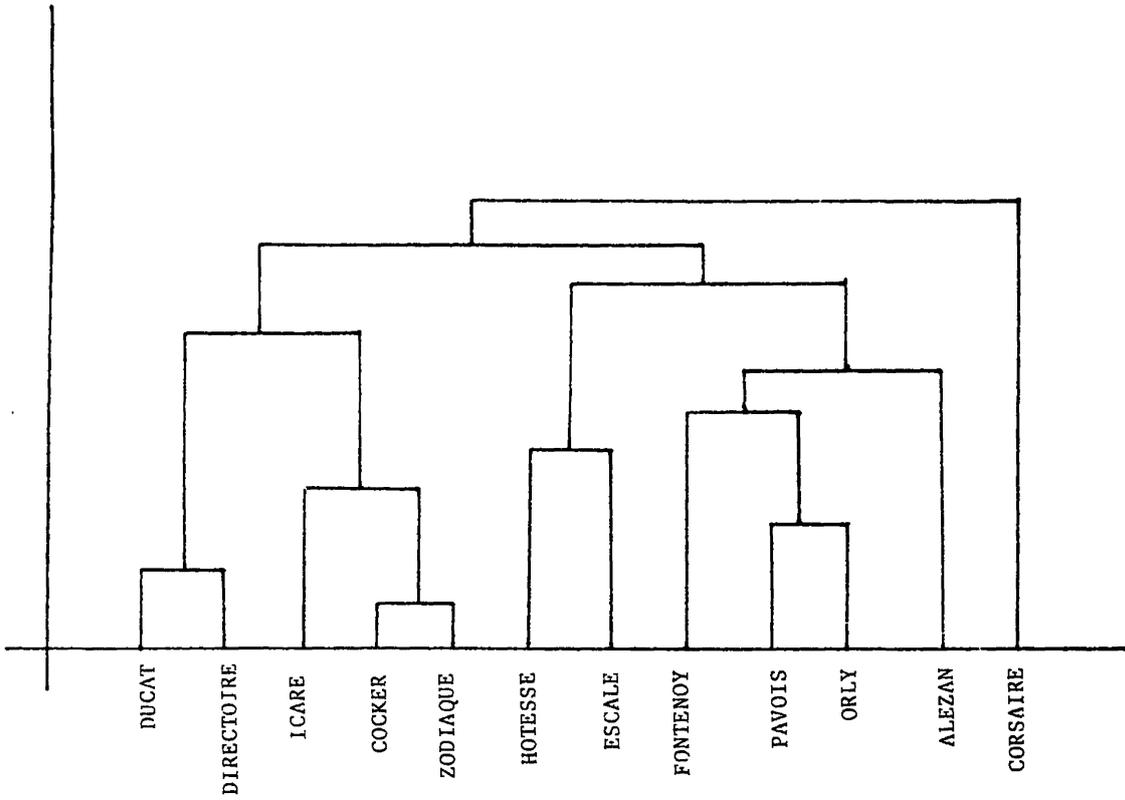
Nous avons choisi l'exemple bien connu, du choix d'un nom pour une nouvelle marque de cigarettes. L'exemple fut traité à l'origine par B. ROY {9} et repris ensuite par J.P. BENZECRI {1} ainsi que par J.F. MARCOTORCHINO et P. MICHAUD dans {7}. Ces analyses précédentes nous permettront de disposer de points de comparaisons pour les résultats.

"Il s'agit d'une enquête effectuée pour la régie Française des tabacs sur cent fumeurs pour déterminer le nom que l'on doit donner à une cigarette, brune, qui doit apparaître comme: luxueuse, de qualité supérieure, destinée à des fumeurs virils, raffinés, distingués et de niveau socio-économique élevée (sic).

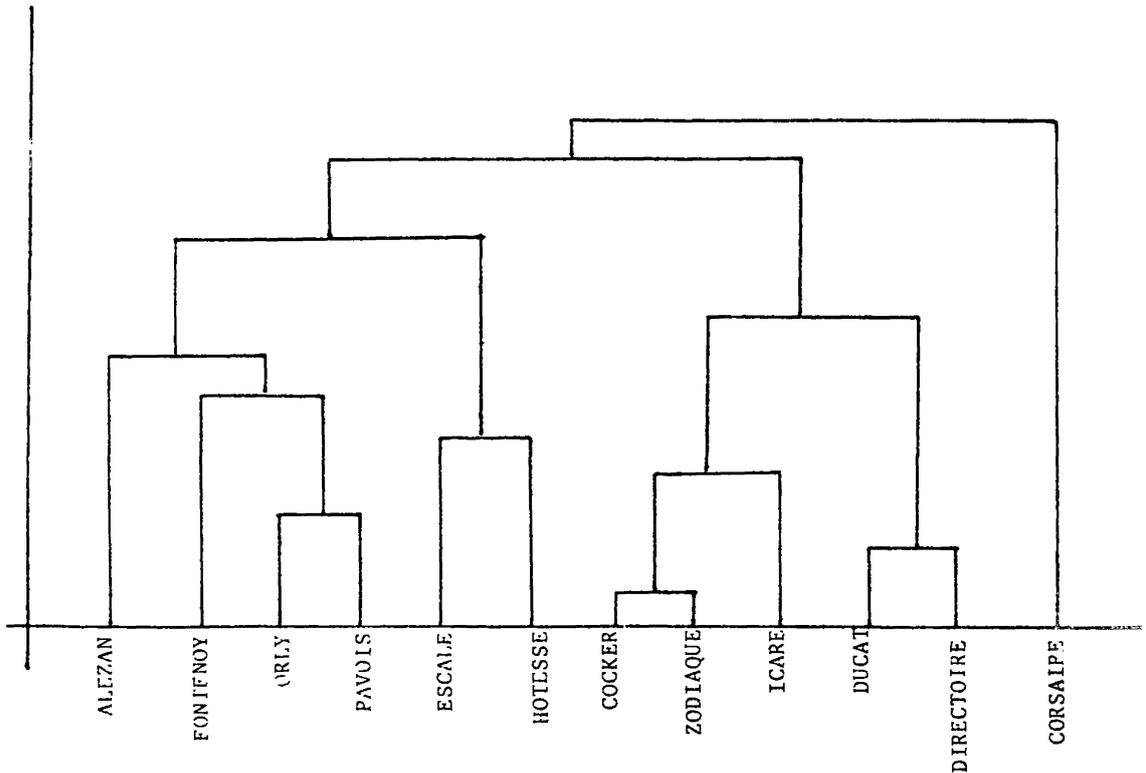
Le choix porte sur douze noms de marque: ALEZAN, COCKER, CORSAIRE, DIRECTOIRE, DUCAT, ESCALE, FONTENOY, ICARE, ORLY, PAVOIS, ZODIAQUE et onze attributs, auxquels on devait associer les noms des marques qui leur correspondaient le mieux: vieillot-désuet, nouveau-riche, sobre-élégant, cocasse-ridicule, racé, nièvre, distingué, vulgaire-commun, pour un homme, pour une femme, pour une petite nature."

1) B. ROY utilisant une méthode de graphe retiend deux noms ALEZAN et FONTENOY parmi lesquels la régie retiendra FONTENOY.

2) J.P. BENZECRI utilisant l'analyse des correspondances fournit une analyse détaillée de résultats facteurs par facteurs. Une analyse selon les critères fixés par la régie lui permet de retenir dans une première étape ALEZAN et FONTENOY avant de décider pour ALEZAN.



Agrégation Complete Linkage - distance  $\chi^2$  - Programme JAMBU  
figure 9



Le même arbre dont les noeuds terminaux ont été ordonnés  
figure 10

3) J.F. MARCOTORCHINO et P. MICHAUD considèrent les onze attributs comme onze juges sur les marques et recherchent un ordre global par agrégation d'échelles. Ils aboutissent à l'ordre suivant: ALEZAN, FONTENOY, ORLY, HOESSE, PAVOIS, DIRECTOIRE, CORSAIRE, ESCALE, DUCAT, COCKER, ICARE et ZODIAQUE. Ils considèrent de plus que cet ordre est stable et significatif au moins sur les premiers classés.

Quant à nous, nous avons effectué simplement une classification ascendante hiérarchique sur l'ensemble des marques. La méthode utilisée est celle du diamètre maximum, avec comme distance celle du  $\chi^2$  comme dans les analyses 2) et 3). Le programme utilisé est celui de JAMBU. Ceci nous fournit une classification, fort intéressante et sans surprise, des marques, représentée sur la figure 9.

Mais évidemment cette classification ne nous permet pas de choisir une marque plutôt qu'une autre. Ce qui n'a rien de surprenant car ce n'est pas bien sûr, l'objet des méthodes de classifications.

Maintenant si nous permutons les noeuds de l'arbre selon l'algorithme présenté au paragraphe VI nous obtenons la hiérarchie représentée figure 10. Il s'agit bien, du même arbre de classification que celui de la figure 9 mais l'ordre des éléments sur l'axe horizontal est maintenant interprétable. Cet ordre, très proche de celui obtenu par J.F. MARCOTORCHINO et P. MICHAUD, est même identique pour les premiers éléments, et, de plus, si on doit prendre une décision ce sera bien sûr ALEZAN puis FONTENOY, en accord avec les analyses précédentes.

Du même coup, l'interprétation de l'arbre en terme de classification s'en trouve grandement facilitée, ce qui reste quand même le but principal de l'algorithme. La classification y est en effet privilégiée par rapport à l'ordonnance puisque celle-ci se fait sous contrainte de compatibilité avec la hiérarchie et non l'inverse.

## XI - CONCLUSION

L'essentiel de notre propos est donc de présenter une procédure simple susceptible de faciliter l'interprétation des classifications hiérarchiques en ajoutant une information supplémentaire à la représentation de la hiérarchie par une interprétation désormais possible de l'axe horizontal du graphique.

Cette procédure, indépendante de l'algorithme d'agrégation, permet de choisir parmi les  $2^{n-2}$  représentations possibles, une qui soit significative au regard des distances initiales. Ainsi on facilite la lecture et l'interprétation des résultats, notamment par les non spécialistes.

## BIBLIOGRAPHIE

- { 1 } J.P. BENZECRI : l'analyse des correspondances, Dunod (1973)
- { 2 } A.D. GORDON : Classification in the presence of constraints. *Biometrics*, Vol.29, n°4, pp.821-822 (1973)
- { 3 } M. JAMBU : Classification automatique pour l'analyse des données.Tomes 1 et 2, Dunod, Paris (1978)
- { 4 } J. KRUSKALL : The Relationship between Multidimensional scaling and clustering. Dans: classification and clustering, Van Rysing, Academic Press (1977)
- { 5 } J.A. HARTIGAN : Clustering Algorithms. John Wiley, New York (1975)
- { 6 } Y. LECHEVALLIER : Classification optimale sous contrainte d'ordre total. IRIA (1978)
- { 7 } J.F. MARCOTORCHINO et P. MICHAUD : Optimisation en analyse ordinale des données.Masson, (1979)
- { 8 } S. SATTAN and A. TVERSKY : Additive similarity trees. *Psychometrika*,Vol.42, n°3 (1977)
- { 9 } B. ROY : Algèbre moderne et théorie des graphes, Dunod (1969)
- { 10 } W.T. WILLIAMS : Principles of clustering,Annual Review of Ecology, Vol.2 (1971)