REVUE DE STATISTIQUE APPLIQUÉE

GILDAS BROSSIER

Algorithmes d'ordonnancement des hiérarchies binaires et propriétés

Revue de statistique appliquée, tome 32, n° 3 (1984), p. 65-79

<http://www.numdam.org/item?id=RSA_1984__32_3_65_0>

ALGORITHMES D'ORDONNANCEMENT DES HIERARCHIES BINAIRES ET PROPRIETES

Gildas BROSSIER

UER Sciences et Techniques
Université de Haute-Bretagne

RESUME

Le résultat d'une classification hiérarchique est généralement représenté sous la forme d'un arbre hiérarchique, en général binaire, appelé aussi dendrogramme. Il existe alors 2^{n-1} représentations possibles de cet arbre, toutes équivalentes du point de vue de la classification.

Afin de faciliter l'interprétation de la hiérarchie, on ordonne les nœuds terminaux de celle-ci selon différents types de critères ou de contrraintes.

Nous présentons différents algorithmes d'ordonnancement et leurs propriétés.

ABSTRACT

The result of a hierarchical clustering is generally represented as a, usually binary, hierarchical tree, also called a dendrogram. There are 2^{n-1} possible representations of this tree which are equivalent from a clustering point of view.

To make this representation more easy to interpret, we order the terminal nodes of the tree under some different constraints and criteria.

Different algorithms are proposed with their properties

I. INTRODUCTION

Les méthodes de classification hiérarchique consistent à transformer une matrice initiale de distance ou plus généralement de dissimilarité en une matrice vérifiant l'inégalité ultramétrique. Selon les cas il est aussi possible d'avoir en donnée initiale une matrice de similarité ou de proximité. Le résultat est représenté sous la forme d'un arbre hiérarchique que certains appellent un dendrogramme.

Généralement cet arbre est binaire, c'est-à-dire que chaque nœud se divise en deux branches et deux seulement. Nous nous limiterons ici à ce cas qui est de loin le plus fréquent.

Toute hiérarchie binaire admet 2^{n-1} représentations, toutes équivalentes, ces représentations diffèrent entre elles par l'ordre des nœuds terminaux de l'arbre. Si la hiérarchie n'était pas binaire, elle admettrait plus de 2^{n-1} représentations.

Il s'agit de choisir parmi toutes ces représentations celle qui est la meilleure au sens d'une contrainte additionnelle dans le but de faciliter l'interprétation.

Mots-Clés : **Classification hiérarchique, ordres, ordonnancement.**

Cette contrainte additionnelle peut être perçue comme une variable "supplémentaire" qui jouerait un rôle de variable "explicative" de la hiérarchie.

Ce problème a déjà fait l'objet de plusieurs publications séparées, étudiant chacune un cas particulier (voir références). Nous proposons ici d'étudier les algorithmes proposés et de les étendre aux autres cas possibles.

Nous présentons sur un exemple l'application de ces algorithmes dans différents cas.

II. NOTATIONS ET RAPPELS

Les données sont l'ensemble E des éléments à classer et la matrice U des distances ultramétriques obtenue par une méthode de classification hiérarchique.

Si T est un ordre sur les éléments de E, alors l'ordre T est dit compatible avec l'ultramétrique U si on peut associer à celle-ci un arbre hiérarchique, plan, sans croisement, dont les éléments terminaux sont rangés dans l'ordre T.

Cette définition découle d'une autre plus générale. Si D est une matrice de distance quelconque sur les éléments de E, alors un ordre total T est compatible avec D si :

$$\forall x, y, z \in E \quad x \underset{T}{<} y \underset{T}{<} z \Leftrightarrow (d(x,y) < d(x,z) \text{ et } d(y,z) < d(x,z))$$

Autrement dit un ordre est compatible avec une matrice de distance si tous les triplets sont compatibles. Il est intéressant de noter qu'en général une matrice de distance D n'admet pas d'ordre compatible et que si elle en admet un, il est unique (à la condition que $\forall x, y, z\ t\ \ d(x,y) \neq d(z,t)$.

A l'inverse, une matrice de distance ultramétrique admet au moins 2^{n-1} ordres compatibles, et à chaque ordre compatible est associée une représentation de la hiérarchie.

On notera $\mathcal{O}_u$, l'ensemble des 2^{n-1} ordres compatibles avec l'ultramétrique U. On supposera que la hiérarchie est binaire et donc que U n'admet que 2^{n-1} ordres compatibles.

Le problème est donc de chercher un ordre T appartenant à $\mathcal{O}_u$ et qui soit le meilleur en un sens que nous allons préciser.

III. LA VARIABLE EXTERNE

Pour donner les nœuds terminaux d'un arbre hiérarchique, on peut se fixer différents types d'objectifs. Par exemple, l'ordre sur les nœuds terminaux doit être aussi proche que possible d'un ordre donné. Cet ordre donné est ce que nous appelerons une variable supplémentaire pour la représentation de l'arbre, ou plus simplement la variable externe V.

Cette variable externe peut être de différents types. Elle peut être une variable ordinale, préordinale, ou être une variable liée à une partition, une va-

riable quantitative, mais elle peut être aussi une matrice de distance ou même une autre hiérarchie.

Selon le type de la variable, on associera des critères permettant de mesurer l'adéquation entre l'ordre recherché et la variable externe.

Le rôle de la variable externe est un apport supplémentaire d'information pour la lecture de la hiérarchie. En effet quelque soit la variable externe la classification est invariante par certaines permutations de ses éléments terminaux, seule varie sa représentation. Comme c'est la représentation qui est le support de l'interprétation il est intéressant de s'aider de variables supplémentaires, destinées à faciliter sa lecture. En effet interpréter une classification c'est nommer les différentes classes et chercher à comprendre pourquoi celles-ci se sont formées. C'est ce rôle d'aide à l'interprétation que peut jouer la variable externe.

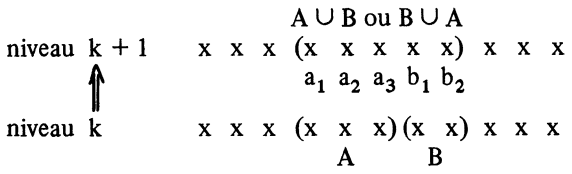
Avant d'étudier chaque cas, nous allons présenter deux algorithmes généraux permettant d'ordonner la hiérarchie et étudier à quelles conditions ils peuvent fournir une solution optimale.

IV. DEUX ALGORITHMES GENERAUX

Les deux algorithmes parcourent l'arbre une seule fois, de la base au sommet pour l'ascendant et du sommet à la base pour le descendant, en orientant chacun des $n-1$ nœuds. De cette façon, les 2^{n-1} ordres compatibles possibles sont accessibles.

a) Principe de l'algorithme ascendant

On part d'un ordre T quelconque. Pour passer au niveau supérieur, l'arbre fusionne deux éléments (ou deux classes) A et B. On a le choix alors entre l'ordre AB et l'ordre BA. Pour choisir entre ces deux ordres, on procède à un test dépendant du critère à optimiser. Le processus est ainsi itéré jusqu'au sommet de l'arbre.

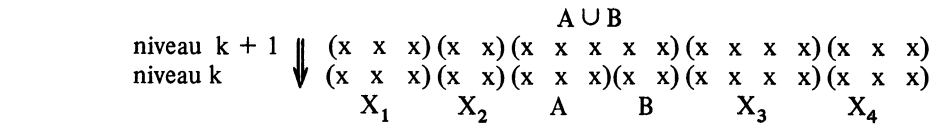


Remarquons que pour passer du niveau k au niveau $k+1$, il faut fusionner et ranger deux classes A et B dont tous les éléments ont été rangés aux niveaux inférieures. De sorte qu'on choisit entre l'ordre AB et l'ordre BA sans mettre en cause les ordres internes aux classes A et B.

A l'opposé, les classes externes aux classes A et B ne sont pas rangées entre elles, elles le seront à un niveau supérieur.

b) Principe de l'algorithme descendant

On part du sommet de l'arbre. Pour passer du niveau k + 1 au niveau k, la classe (A ∪ B) se divise en la classe A et la classe B. On procède alors à un test permettant de choisir entre l'ordre AB et l'ordre BA, le test dépendant du critère à optimiser. Le processus est itéré jusqu'aux nœuds terminaux de l'arbre pour obtenir l'ordre T recherché.



A l'inverse de l'algorithme précédent, quand on passe du niveau k + 1 au niveau k, les éléments internes aux classes A et B ne sont pas rangés. Ils le seront à un niveau inférieur. Mais les éléments externes aux classes A et B sont rangés (X_1, X_2, X_3, et X_4). Autrement dit toutes les classes d'un même niveau k sont ordonnées, mais les éléments les constituant ne sont pas ordonnés à l'intérieur des classes.

c) Propriétés des algorithmes

La première chose à vérifier est que l'ordre T résultant est compatible avec l'ultramétrique U. Ce résultat découle directement de la construction de T. En effet en mettant contigues à chaque niveau les classes A et B à fusionner ou à séparer, on est assuré d'obtenir une hiérarchie sans croisement et donc un ordre compatible.

Ensuite, nous devons remarquer qu'il y a unicité de la solution uniquement si les tests conduisent tous à des inégalités strictes (A avant B ou l'inverse). Il y aura 2^p solutions équivalentes au sens du critère s'il y a p tests qui donnent lieu à p égalités, c'est-à-dire p indéterminations (A avant B est aussi bon au sens du critère que B avant A).

D'un point de vue temps de calcul, les algorithmes nécessitent n – 1 étapes avec, à chaque étape, un test (généralement très simple).

Pour étudier les conditions d'optimalité de la solution, nous définissons les relations entre le test et le critère :

- On note C(T, V) le critère qui mesure l'adéquation entre un ordre T et la variable externe V. On supposera que le critère est à minimiser.
- On note T′ l'ordre qui se déduit de T en permutant deux classes contigues A et B sans changer l'ordre interne à A et à B.
- On dira que, $\mathcal{T}(k, T)$ test pour l'ordre T au niveau k, a la valeur vraie si $C(T, V) < C(T', V)$. De plus, le test $\mathcal{T}$ est dit indépendant de l'ordre interne (res. externe) aux classes A et B si le résultat du test ne dépend pas de cet ordre.

On a alors les théorèmes suivants :

Théorème 1

Si le test $\mathcal{T}$ associé au critère C est indépendant de l'ordre interne aux classes, alors l'algorithme descendant engendre un ordre qui est optimal au sens de ce critère.

$$C(T, V) = \min_{R \in \mathcal{O}_u} C(R, V)$$

Démonstration

Supposons que T ne soit pas optimal. Soit R un ordre différent de T et optimal au sens du critère. Les deux ordres étant différents, il y a au moins un nœud où les deux ordres diffèrent. Soit k le premier nœud (en descendant). Comme $\mathcal{C}(k, T)$ est vrai par construction et que le test est indépendant de l'ordre interne, $\mathcal{C}(k, R)$ est faux. Donc l'ordre R' qui se déduit en R en permuttant les classes A et B du nœud k est tel que $C(R', V) < C(R, V)$.

Ce qui contredit l'optimalité de R.

Théorème 2

Si le test $\mathcal{C}$ associé au critère C est indépendant de l'ordre externe aux classes, alors l'algorithme ascendant engendre un ordre qui est optimal au sens du critère.

La démonstration est identique á celle du théorème 1.

Il faut remarquer que si l'hypothèse d'indépendance est nécessaire pour montrer l'optimalité de la solution, elle est aussi nécessaire à la construction de la solution. En effet, en raisonnant sur l'algorithme descendant, il n'est pas possible de déterminer la valeur du test si celui-ci dépend de l'ordre interne, car cet ordre n'est connu qu'à un niveau inférieur.

Nous allons maintenant voir, en fonction du type de la variable, quels sont les critères et les tests que l'on peut mettre en œuvre.

V. LA VARIABLE V EST DU TYPE ORDINAL, PREORDINAL OU QUANTITATIF

a) Les critères

Quand la variable V est du type ordinal, on utilise les deux critères classiques, le tau de Kendall, $\tau(T, V)$, ou le rho de Spearman $\rho(T, V)$. Rappelons que maximiser $\rho(T, V)$ revient au même que minimiser $||T - V||$ où T et V désignent les vecteurs rangs associés aux deux ordres.

Ces deux critères se généralisent au cas où V est un préordre. On cherche alors à minimiser $d_k(T, V)$ où d_k est la distance de Kendall des désaccords généralisés au cas d'un préordre :

- il y a désaccord si $T_i < T_j$ et $V_i > V_j$
  ou $T_j < T_i$ et $V_j > V_i$
- il n'y a pas désaccord si $V_i = V_j$

Remarquons que dans ce cas (ordre total et préordre), minimiser d_k revient au même que minimiser la distance de la différence symétrique sur les deux relations binaires T et V.

L'autre critère se réécrit de la même façon : minimiser $||T - V||$, avec T et V les vecteurs rangs associés.

Si la variable externe est quantitative, la distance de Kendall n'a plus de sens sauf à considérer la variable quantitative dans son aspect ordinal mais on est ramené au cas précédent. On cherchera donc à minimiser $||T - V||$ ou T est le vecteur rang de l'ordre recherché, et V désigne le vecteur associé à la variable externe.

b) Tests appliqués à chaque niveau

Si le critère C est la distance des désaccords de Kendall (cas d'un ordre ou préordre), on calcule $d_k(A \cup B, V)$ et $d_k(B \cup A, V)$ qui sont les nombres des désaccords par rapport à la variable V (en fait V est restreint aux éléments appartenant à $A \cup B$).

$d_k(A \cup B, V)$: il y a un désaccord si $i \in A$, $j \in B$ et $V_i > V_j$
ou $i \in B$, $j \in A$ et $V_i < V_j$

$d_k(B \cup A, V)$: il y a un désaccord si $i \in B$, $j \in A$ et $V_i > V_j$
ou $i \in A$, $j \in B$ et $V_i < V_j$

Dans les deux cas il n'y a pas désaccord si $V_i = V_j$ (cas du préordre). On choisit l'ordre AB ($\mathcal{C}(T)$ est vrai) si $d_k(A \cup B, V) < d_k(B \cup A, V)$, et l'ordre BA sinon $\mathcal{C}(T')$ est alors vrai).

T et T' ne diffèrent que par les intervalles A et B, il est immédiat que :

$$\mathcal{C}(T) \text{ est vrai} \Leftrightarrow d_k(T, V) < d_k(T', V)$$

D'autre part, on vérifie également que $\mathcal{C}(T)$ est indépendant à fois de l'ordre interne et de l'ordre externe aux classes A et B.

Si le critère C est de maximiser le ρ de Spearman ou plus généralement de minimiser $\|T - V\|$ alors on calcule $\bar{V}_A$ et $\bar{V}_B$ avec :

$$\bar{V}_A = \frac{1}{n_A} \sum_{i \in A} V_i \text{ et } \bar{V}_B = \frac{1}{n_B} \sum_{i \in B} V_i$$

n_A et n_B étant les cardinaux des classes A et B.

V_i désignant soit le rang de l'individu i si V est un ordre ou un préordre, soit la valeur de la variable quantitative V pour l'individu i. On choisit l'ordre AB ($\mathcal{C}(T)$ vrai) si $\bar{V}_A < \bar{V}_B$, l'ordre BA sinon.

On a la propriété suivante :

Propriété

Si $\mathcal{C}(T)$ est vrai alors $\|T - V\| < \|T' - V\|$ et réciproquement.

Démonstration

$$\|T' - V\|^2 = \|T\|^2 + \|V\|^2 - 2\sum_i T_i V_i,$$

donc

$$0.5(\|T - V\|^2 - \|T' - V\|^2) = \sum_{i \in A \cup B} T'_i V_i - \sum_{i \in A \cup B} T_i V_i$$

$$= \sum_{i \in A} (T'_i - T_i) V_i + \sum_{i \in B} (T'_i - T_i) V_i$$

Pour $i \in A$, $T'_i - T_i = n_B$, on a décalé les éléments de A de n_B rangs

$i \in B$, $T' - T_i = -n_A$, on a décalé les éléments de B de n_A rangs dans l'autre sens.

Donc :

$$0.5\,(\|T - V\|^2 - \|T' - V\|^2) = n_B \sum_{i \in A} V_i - n_A \sum_{i \in B} V_i$$

et $\|T - V\|^2 < \|T' - V\|^2 \Leftrightarrow \bar{V}_A < \bar{V}_B = \mathcal{C}(T)$ vrai et $\mathcal{C}(T')$ faux.

Il est évident que le calcul de $\bar{V}_A$ et de $\bar{V}_B$ est indépendant de l'ordre interne aux classes A et B et à fortiori de l'ordre externe.

Les tests associés aux deux critères vérifiant les deux conditions d'indépendance, les deux algorithmes, conduisent à un ordre optimal pour chacun des deux critères, que la variable externe soit ordinale, préordinale ou quantitative.

Si on veut utiliser un autre critère, il suffit de vérifier que le test associé vérifie une des deux conditions d'indépendance, et de choisir l'algorithme en conséquence.

L'optimalité avait déjà été démontrée par BROSSIER dans [2] pour le cas où la variable externe est un ordre, le critère le ρ de SPEARMAN, et l'algorithme descendant. D'autre part R. DEGERMAN avait énoncé dans [4] le résultat concernant le cas où la variable externe est ordinale, le critère le τ de Kendall et l'algorithme ascendant.

VI. LA VARIABLE V EST UNE MATRICE DE DISTANCE D

Ce problème fut le premier traité [1], [7]. En effet, la matrice des distances ultramétriques étant une approximation de la matrice des distances initiales, il est intéressant de choisir la représentation qui restitue au mieux les distances initiales, de façon à rester aussi proche que possible des données de base.

Les deux algorithmes proposés sont les suivants :

a) Algorithme ascendant de GRUVAEUS et WAINER [7]

Les auteurs présentent seulement un algorithme sans justifier d'un critère. A l'analyse cet algorithme procède à une série de tests locaux tendant à optimiser le critère suivant : rechercher un ordre T qui engendre une chaîne, sur les éléments terminaux, de longueur minimale au sens de la matrice de distance D.

De façon plus explicite, en notant $x_1, \ldots, x_n$ les éléments terminaux et si T, l'ordre cherché, est l'ordre $1, \ldots, i, \ldots, n$, la somme

$$\sum_{i=1}^{n-1} D(x_i, x_{i+1}) \text{ est minimale.}$$

Pour passer du niveau k au niveau k + 1, il faut fusionner les deux classes A et B. Les éléments internes à A et à B ont été rangés à un niveau inférieur. Si x_A et y_A sont les éléments extrémaux de la classe A et, x_B, y_B, ceux de la classe B, on va fusionner A et B en rangeant côte à côte les deux éléments extrêmaux les plus proches au sens de D (par exemple, ici, y_A et y_B).

niveau k + 1 . . . . $(x_A \; . \; y_A \; y_B \; . \; x_B)$ . . .

⇑

niveau k . . . . $(x_A \; . \; y_A)(y_B \; . \; x_B)$ . . .

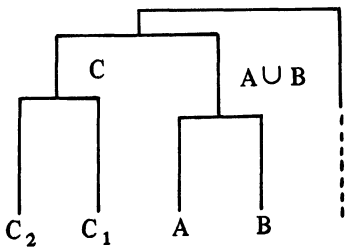
Cet algorithme diffère donc de l'algorithme ascendant, précédemment énoncé, car les classes A et B sont en fait orientées l'une par rapport à l'autre parmi 4 possibilités :

$$x_A - y_A\, y_B - x_B \,,\; x_A - y_A\, x_B - y_B \,,\; y_A - x_A\, x_B - y_B \,,\; y_A - x_A\, y_B - x_B$$

Le nombre de 2^{n-1} tests reste quand même vrai car aux premiers niveaux, quand chaque classe ne contient qu'un élément, il n'y a pas lieu de faire un test.

b) L'algorithme descendant, BROSSIER [1]

Il est du même type que celui énoncé au § IV, seule la nature du test change. En effet, on considère la classe C_1 qui est la plus proche (au sens de l'ultramétrique) du noeud AB pour orienter celui-ci.



Lors de la construction de la hiérarchie on a calculé, à partir des distances initiales entre les éléments, les distances inter-classes à chaque niveau de l'agrégation. Ce sont ces distances inter-classes $D(A, C_1)$ et $D(B, C_1)$ déjà calculées que l'on va utiliser. Donc si $D(A, C_1) < D(B, C_1)$, on rangera la classe A à côté de la classe C_1.

Le critère, dans ce cas, est de maximiser pour l'ordre T le nombre des triplets (x, y, z) compatibles avec la distance initiale D.

c) Propriétés de ces deux algorithmes

Dans les deux cas l'optima n'est pas atteint. Cependant, si la matrice D admet un ordre compatible T et que celui-ci soit compatible avec l'ultramétrique, on peut montrer que l'algorithme conduit à cet ordre T qui est optimal [1]. L'algorithme a) utilise toujours, pour mesurer la distance entre deux classes, la distance du lien

simple (min d(x, y), $x \in A$ et $y \in B$) ce qui est un inconvénient si la hiérarchie a été construite en utilisant une autre notion de distance.

A l'opposé, l'algorithme b) utilise la distance qui a servi à construire la hiérarchie. Ceci est aussi un inconvénient si la variable externe n'est pas la matrice des distances initiales, car il est nécessaire de calculer les distances entre classes.

d) Autres critères

On peut, comme au § V, chercher à minimiser un critère de moindres carrés et utiliser les algorithmes du § IV. Si T est l'ordre recherché, on associe à T la matrice carrée de la différence des rangs, R, définie par $R_{ij} = |T_i - T_j|$.

D étant la matrice de distance donnée, on cherche R minimisant $\|D - R\|$. Malheureusement le test associé à ce critère dépend à la fois de l'ordre interne aux classes A et B, et de l'ordre externe. Donc ni l'algorithme ascendant, ni l'algorithme descendant ne fourniront un ordre optimal.

Si la donnée initiale n'est pas la matrice des distances mais un tableau rectangulaire, individu x variables quantitatives, on peut utiliser celui-ci comme variable externe ainsi que le propose Y LE FOLL [8]. L'algorithme est celui du §b, seul change le test :

On oriente le noeud AB de façon à rendre positif le produit scalaire :

$$\langle (A \cup B, C), (A, B) \rangle$$

L'idée de l'auteur est de lier la représentation de l'arbre à la représentation géométrique des éléments dans R^p. Le fait de choisir l'ordre des noeuds de façon à rendre positif le produit scalaire ci-dessus tend à rechercher un certain "alignement" de ces éléments terminaux.

L'auteur ne parle ni du critère, ni de l'optimalité.

VII. LA VARIABLE V EST UNE HIERARCHIE OU UNE PARTITION

a) Hiérarchie

Dans ce cas, on a comme données, deux ultramétriques U et V sur le même ensemble E. On cherche deux ordres $T_u \in \mathcal{O}_u$ et $T_v \in \mathcal{O}_v$ tels que T_u et T_v soient les plus proches possibles.

Pour ce faire, l'approche la plus efficace est celle proposée par DIDAY dans [5]. Le critère à maximiser est de rechercher $T_u \in \mathcal{O}_u$ et $T_v \in \mathcal{O}_v$ comportant les plus grandes séquences communes. En effet, en général $\mathcal{O}_u \cap \mathcal{O}_v = \emptyset$ et il n'existe pas d'ordre compatible commun aux deux hiérarchies.

L'algorithme est le suivant :

- on part d'un élément quelconque x.
- on cherche l'ensemble $M_u(x)$, des éléments à distance minimale de x au sens de U.
- on fait de même avec la distance V, soit $M_v(x)$.

– si $M_u(x) \cap M_v(x) \neq O$, on choisit y dans cet ensemble, on le range à côté de x et on recommence.

Sinon, ou bien on a terminé
ou bien on remet en cause le choix précédent (x) et on recommence.

L'optima n'est en général pas atteint sauf à parcourir toutes les possibilités ce qui serait en général trop long. Cependant, si $\mathcal{O}_u \cap \mathcal{O}_v \neq \emptyset$, l'algorithme conduit à un ordre commun aux deux hiérarchies.

L'algorithme a été étudié et mis en oeuvre par E. GAUD dans [8].

b) Partition

Il n'existe pas dans ce cas d'algorithme vraiment efficace. En effet un algorithme ascendant ou descendant devient vite sans intérêt du fait du trop grand nombre de cas possibles à parcourir.

Deux solutions sont possibles :

– considérer la partition comme une hiérarchie à deux niveaux et appliquer l'algorithme précédent.
– ordonner de façon arbitraire les classes de la partition pour la transformer en un préordre, et appliquer un des algorithmes précédents qui sont rapides et conduisent à un optima.

Il faut remarquer que l'on peut, si la partition n'a que deux classes, la considérer comme un préordre sans changer la nature du problème et obtenir une solution optimale.

VIII. UN EXEMPLE D'APPLICATION

Les données concernent un ensemble de 18 pays répartis dans le monde entier et choisis pour le rôle important qu'ils jouent dans leur continent respectif.

Il s'agit de : U.S.A, Canada, R.F.A, Royaume-Uni, France, Italie, URSS, Japon, Corée du Sud, Inde, Pakistan, Indonésie, Nigéria, Egypte, Zaire, Argentine, Brésil et Mexique.

Les 17 variables relèvent des différents domaines :

– *démographie* : densité (hab/km), taux de croissance de la population (%), taux de mortalité infantile (%), espérance de vie (années), taux de population urbaine (%).
– *culturel* : taux de scolarisation en 1^{er} et $2^{è}$ degré (%), en $3^{è}$ degré (%), poste de radio par ménage (%).
– *militaire* : proportion des dépenses militaires dans le P.I.B. (%)
– *économique* : P.I.B. par habitant ($/hab), structure de la population travaillant dans l'agriculture (%), dans l'industrie (%), dans les services (%), taux d'inflation (%), consommation d'énergie (TEC/hab), commerce extérieur (% du P.I.B.).

Les données sont extraites de "l'état du monde 1981" et concerne 1980. Les variables ayant été centrées, réduites, on a calculé le tableau des distances eucli-

diennes entre ces différents pays. L'ultramétrique a été obtenue à partir de ce tableau par l'algorithme du diamètre minimum (complete linkage).

La classification obtenue est sans surprise et s'interprète aisément selon le degré et le mode de développement des pays. Cependant, il est intéressant de regarder les rôles que peuvent jouer différentes variables dans la formation des classes en essayant d'ordonner la hiérarchie en fonction de ses variables.

V2	V1	T	
2556	1	1	. USA
251	8	2	. Canada
1082	2	3	. URSS
1038	3	4	. Japon
824	4	5	. RFA
515	6	6	. R.U.
648	5	7	. France
392	7	8	. Italie
56	13	9	. Corée
207	9	10	. Brésil
115	11	11	. Mexique
61	12	12	. Argentine
7	18	13	. Zaire
126	10	14	. Indre
55	14	15	. Nigéria
52	15	16	. Indonésie
21	16	17	. Pakistan
18	17	18	. Egypte

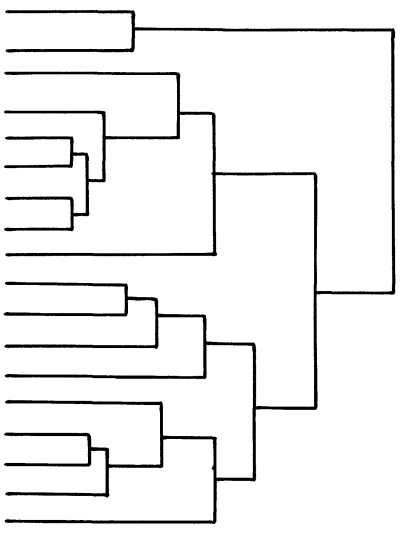


Figure 1. – hiérarchie ordonnée en fonction de la variable ordinale V1, associée à la variable quantitative V2, P.I.B. global de chacun des pays. T ordre optimal obtenu par l'algorithme.

Une possibilité est de ne retenir que la richesse en prenant pour variable externe le P.I.B. de chacun des pays (variable V2, en milliard de $). Dans un premier temps, nous avons rangé les pays par P.I.B. décroissant, obtenant ainsi la variable ordianle V1. La hiérarchie ordonnée sous la contrainte de V1 est la suivante :

L'ordre de la variable V1 n'étant pas compatible avec la hiérarchie, on obtient un ordre T qui approche l'ordre V1. Il faut remarquer que sur cet exemple, on obtient le même ordre T si on remplace la variable ordinale V1 par la variable quantitative V2 (P.I.B. de chacun des pays).

La richesse brute est visiblement un facteur important mais il ne saurait tout expliquer. Le simple exemple du couple USA-Canada est représentatif de l'influence d'autres variables.

On va alors s'intéresser à la notion d'industrialisation des différents pays par le biais de la variable % du P.I.B. réalisé dans l'agriculture. On sait que ce pourcentage est élevé dans les pays en voie de développement et a tendance à décroître au fur et à mesure du développement économique. On a regroupé les différents pays en 4 classes obtenant la variable préordinale V3 à 4 modalités :

A % compris entre 0 et 10
B % compris entre 10 et 20
C % compris entre 20 et 50
D % supérieur à 50

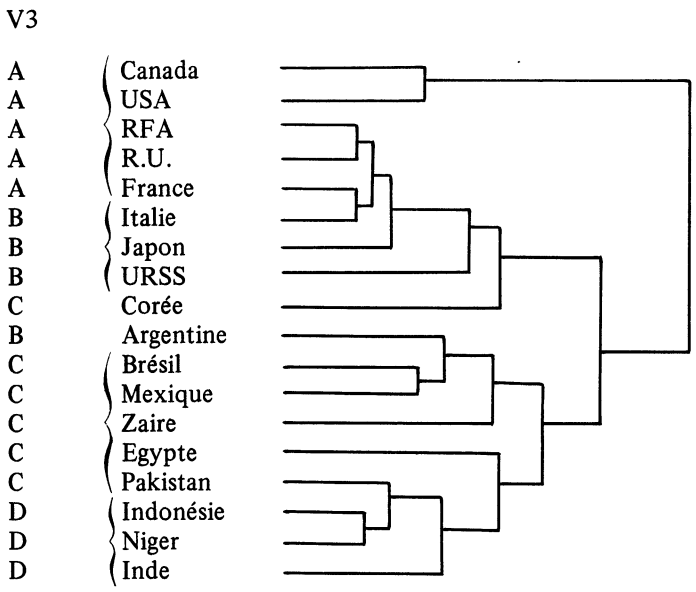


Figure 2. – hiérarchie ordonnée selon la variable préordinale % du P.I.B. réalisé dans l'agriculture.

La encore, il n'existe pas de solution exacte, en effet la Corée du Sud et l'Argentine sont mal ordonnées. Il faut noter aussi que 4 tests conduisent à une indétermination, il y a 2^4 solutions optimales équivalentes dans ce cas. On peut les retrouver aisément sur le graphique en permutant les classes dont tous les pays appartiennent à une même modalité de V3.

On constate un très large accord entre cette variable et la hiérarchie, bien que les classes du préordre ne coincident pas exactement avec les classes de la hiérarchie.

Pour compléter cette lecture de la hiérarchie, on peut chercher à la représenter sous une contrainte géographique. On considère alors la variable V4 qui est la partition en 5 régions du monde : Europe, Asie, Afrique, Amérique du Sud et Centrale, Amérique du Nord.

Dans ce cas, on obtient une solution exacte bien que la partition V4 n'appartienne pas à la hiérarchie.

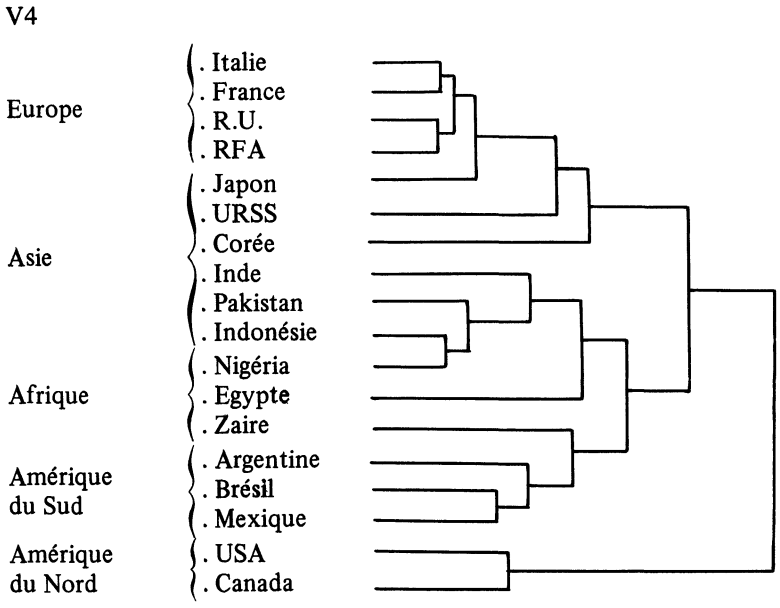


Figure 3. – hiérarchie ordonnée selon la variable appartenance géographique.

On peut utiliser comme variable externe le tableau de distance initiale. Il ne s'agit plus alors de chercher à comprendre la classification en la comparant à une variable particulière mais de la représenter de façon la plus fidèle possible. On obtient la hiérarchie suivante :

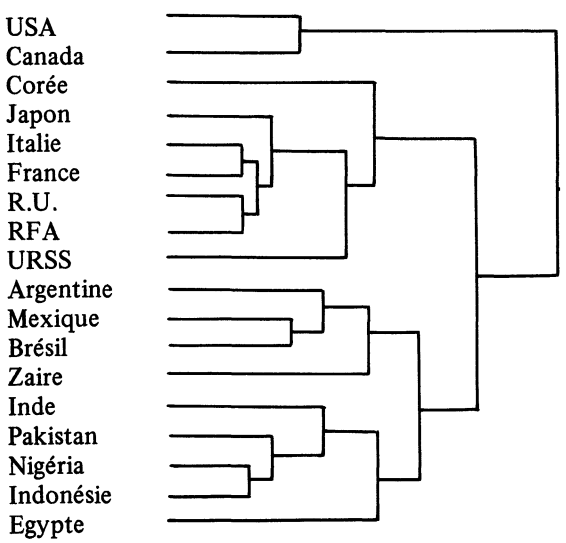


Figure 4. – hiérarchie ordonnée selon la matrice de distance initiale.

Le dernier exemple d'ordonnancement concerne la comparaison des résultats de deux algorithmes de classification hiérarchique différents :

– agrégation selon la moyenne (Average linkage) et
– agrégation selon le diamètre minimum (complete linkage).

On est ici dans le cas où deux ultramétriques ont plusieurs ordres compatibles en commun. Ceci nous permet de comparer facilement les résultats des deux classifications. Quelques différences apparaissent concernant la structure de la hiérarchie (effet de chaine plus marquée pour l'algorithme de la moyenne) et les classes elles-mêmes : par exemple le Zaire est rattaché soit à la classe (Mexique, Brésil, Argentine), soit à la classe (Pakistan, Indonésie, Nigéria, Inde).

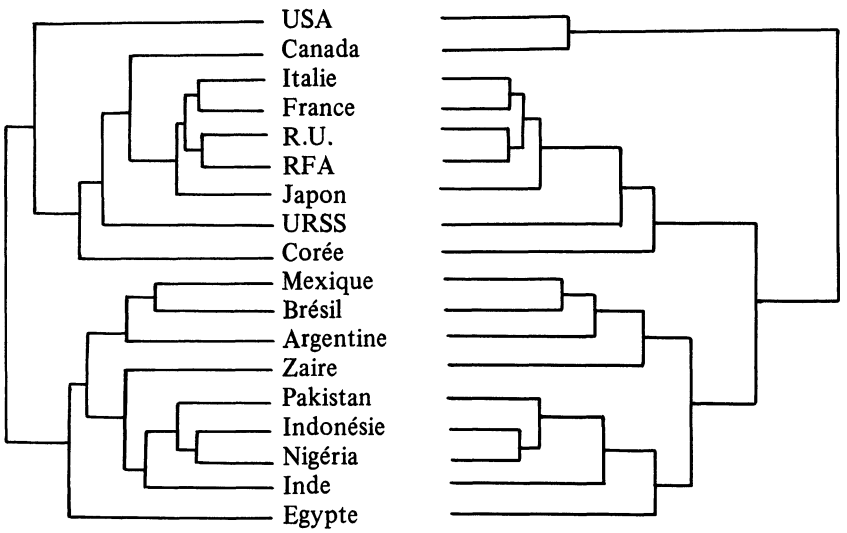


Figure 5. – comparaison de deux hiérarchies.

IX. CONCLUSION

L'utilisation des variables externes permet d'aborder la lecture d'une hiérarchie sous différents angles. On peut ainsi faciliter l'interprétation des classes et apprécier l'importance de telle ou telle variable. Il peut être par-exemple intéressant de prendre comme variable externe, le premier axe d'une analyse factorielle pour faciliter l'interprétation conjointes des résultats.

Si pour trois types de variables nous n'avons pas de solution optimale, celà est peut-être à rattacher au fait que ces types de variables sont du point de vue combinatoire beaucoup plus complexes que les variables du type ordre, préordre, quantitatif.

Il serait peut-être possible en augmentant la complexité des algorithmes de parvenir à une solution optimale, mais je pense que ce type de méthode n'est intéressant que si le temps de calcul nécessaire à l'ordonnancement reste faible par rapport à celui nécessaire au calcul de la hiérarchie.

C'est pour celà que nous n'avons pas envisagé des algorithmes du type aller-retour nous limitant aux algorithmes fonctionnant en n étapes.

X. REFERENCES

[1] G. BROSSIER (1980). – Représentations ordonnées des classifications hiérarchiques, *Statistique et analyse des données*, 2, pp. 31-44.

[2] G. BROSSIER (1982). – Classification à partir de matrices carrées non symétriques, *Statistique et analyse des données*, vol. 7, n° 2, pp. 22-40.

[3] G. BROSSIER (1983). – *Ordonnancement des hiérarchies.* 3e journées internationales. Analyse de données et informatique. INRIA. Versaille.

[4] R. DEGERMAN (1982). – Ordered binary trees constructed through an application of Kendall's tau, *Psychometrika*, vol. 47, n° 4.

[5] E. DIDAY (1982). – *Croisement, ordres et ultramétriques*, Rapport de recherche n° 144. INRIA.

[6] E. GAUD (1983). – *Représentation d'une préordonnance*, Thèse de 3e cycle, Université de Provence.

[7] G. GRUVAEUS et H. WAINER (1972). – Two additions to hierarchical cluster analysis. *Br. J. math. Statist. psychol.*, 25.

[8] Y. LE FOLL (1983). – Sur le choix des aînés et des benjamins nœuds en vue du tracé optimum d'un arbre, *Cahiers Analyse des données*, vol. VIII, n° 2.