



Recherche d'une partition optimale sous contrainte d'ordre total

Yves Lechevallier

► **To cite this version:**

Yves Lechevallier. Recherche d'une partition optimale sous contrainte d'ordre total. [Rapport de recherche] RR-1247, INRIA. 1990. inria-00075311

HAL Id: inria-00075311

<https://hal.inria.fr/inria-00075311>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1247

Programme 5
Automatique, Productique,
Traitement du Signal et des Données

RECHERCHE D'UNE PARTITION OPTIMALE SOUS CONTRAINTE D'ORDRE TOTAL

Yves **LECHEVALLIER**

Juin 1990



* RR - 1247 *

RECHERCHE D'UNE PARTITION OPTIMALE SOUS CONTRAINTE D'ORDRE TOTAL

Yves Lechevallier
INRIA

Domaine de Voluceau, Rocquencourt, B.P. 105
78153 Le CHESNAY Cedex - France

Résumé

Un des problèmes de la classification automatique est de trouver une partition qui soit optimale pour un critère donné. Dans le cas où les objets sont définis par une seule variable numérique la programmation dynamique trouve cette partition pour certains critères. Nous proposons plusieurs critères utilisables dans ce cadre. Plusieurs simulations et applications sont décrites.

OPTIMAL CLUSTERING ON ORDERED SET

Abstract

Cluster Analysis involves the problem of finding a partition which is optimal for a given criterion. For some criteria and if the objects are R-valued points, optimal partition can be derived from Dynamic Programming Algorithm. The purpose of this paper is to present many criteria which give an optimal partition via the DPA. Numerical simulations, and a comparative study are presented.

1.1. Position du problème

1.1.1. Introduction

Le but de la classification automatique non hiérarchique est de trouver la meilleure partition en k classes de l'ensemble E comprenant n objets. Cette recherche nécessite la définition d'un indice de distance d entre ces objets et le choix d'un critère W permettant d'évaluer chaque partition. Ainsi le problème de classification automatique se résume souvent à une optimisation du triplet (E, D, W) où D est le tableau des indices de distances entre les objets de E et W une fonction de valuation sur l'ensemble des partitions de E en k classes $P_k(E)$ dépendante des valeurs d_{ij} . Ainsi on recherche une solution P^* de $P_k(E)$ minimisant le critère W :

$$W(P^*) = \underset{P \in P_k(E)}{\text{Min}} W(P)$$

La complexité du problème naît de son caractère combinatoire. Le nombre de solutions est astronomique dès que le nombre de points à classer dépasse quelques dizaines..

Pour se fixer les idées rappelons quelques chiffres significatifs. On désigne par n le nombre d'objets et k le nombre de classes de la partition. Le nombre de dissimilarités $D(n)$ de n objets est $n(n-1)/2$, ce nombre correspond au cardinal du tableau des distances entre n objets.

Le nombre de partitions de n objets est égal à :

$$S(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^{k-i} \frac{1}{i!(k-i)!} i^n$$

et il est appelé nombre de Stirling du deuxième ordre.

Le nombre de partitions de n objets est égal à :

$$B(n) = \sum_{k=1}^n S(n, k) = e^{-1} \sum_{k=1}^{\infty} k^n / k! \text{ est appelé nombre de Bell.}$$

Le nombre de hiérarchies de partitions est encore plus important et il est égal à :

$$H(n) = n! (n-1)!/2^{n-1}$$

Dans le tableau suivant nous donnons quelques exemples :

n	D(n)	B(n)	H(n)
4	6	15	18
10	45	115 975	2 571 912 000
15	105	1 382 958 545	6.98×10^8
20	190	5.17×10^{13}	5.64×10^{29}
40	780	1.57×10^{35}	3.02×10^{82}
50	1225	1.85×10^{47}	3.27×10^{112}

et $S(100,5) = 10^{68}$ ou $S(60,2) = 10^{18}$

Aussi la nécessité de trouver des méthodes capables de fournir automatiquement une bonne classification sans évaluer toutes les solutions possibles apparaît dès qu'il s'agit de classer plus d'une dizaine d'objets. (En effet, si un ordinateur pouvait évaluer un million de partitions à la seconde, il faudrait plus de 8 jours pour déterminer la partition optimale de 20 objets en 5 classes et plus de 2444 siècles si le nombre est égal à 30).

Face à cette impossibilité d'utiliser la méthode optimale, l'utilisateur a le problème du choix d'une heuristique. La description de l'objectif permet, souvent, de faire un choix judicieux de la méthode et l'algorithme les mieux adaptés à son problème.

La procédure d'énumération complète étant inefficace il faut utiliser les méthodes d'optimisation existantes. Mais dans ce cas pour trouver cette solution optimale il faut, soit réduire l'ensemble des partitions, soit définir des critères de valuation de partition.

La réduction de l'ensemble des partitions est utilisé par les quatre approches suivantes :

- la programmation dynamique (Jensen 1969)
- la programmation en nombre entier (Garfinkel et Nemhauser en 1969).
- procédure de séparation et d'évaluation (Scott 1969)
- procédure de coloration de graphes (Hubert 1974).

Mais si le critère W choisit est le suivant :

$$W(P) = \sum_{k=1}^k w(P_i)$$

avec P une partition en k classes (P_1, \dots, P_k)

$$W(P_i) = \text{Max}_{\substack{x \in P_i \\ y \in P_i}} d(x, y)$$

alors les partitions optimale peuvent être obtenues par la construction de l'arbre de longueur minimum. Ceci se démontre facilement si l'on connaît l'algorithme de Prim (1957), il suffit de démontrer que les partitions optimales en $2, \dots, N$ classes sont emboîtées ce qui permet de construire une hiérarchie.

1.1.2. Exemple d'utilisation de la programmation dynamique

Jensen a proposé un algorithme de programmation dynamique qui recherche la partition optimale au sens d'un critère W défini sur l'ensemble des partitions et qui doit être additif sur l'ensemble des classes.

$$W : P(E) \rightarrow \mathbf{R}^+$$

avec

$$W(P) = \sum_{k=1}^K W(P_k)$$

Soit $P(Q)$ l'ensemble des parties de Q et $P_K(Q)$ l'ensemble des partitions en K classes de l'ensemble Q CE alors la solution optimale en K classes peut être obtenue ainsi :

$$\text{Min}_{P \in P_K(Q)} W(P) = \text{Min}_{C \in P(Q)} \{W(Q_{Q-C}^{k-1}) + w(C)\}$$

où Q_{Q-C}^{k-1} est la solution optimale de l'ensemble $Q-C$ en $K-1$ classes

avec pour $K=2$ $W(Q_{Q-C}^1) = w(Q-C)$

L'amélioration apportée par la programmation dynamique est importante quand le nombre de classes croit, mais cette méthode nécessite le stockage de

la valuation optimale pour toutes les classes possibles de $P(E)$ soit 2^n ce qui limite l'efficacité de cette procédure d'ou le commentaire désabusé de Jensen. "En clair, dans les problèmes extrêmement importants de classification, nos plus puissants ordinateurs peuvent être insuffisants pour l'utilisation pratique de la programmation dynamique et de l'énumération complète, contraignant l'analyste à se satisfaire d'une approche agrégative plus pratique".

Cependant Scott et Symons (1977) montrent que si les données sont dans un espace à p dimension et si on prend le critère de la variance alors il suffit d'examiner $(2^{p-2}) C_n^p$ partitions, mais ce résultat reste inapplicable car l'on ne dispose pas d'une procédure permettant d'énumérer ces partitions intéressantes. Aussi Fisher en réduisant l'espace de représentation à un seul paramètre et en utilisant le critère de la variance donne une procédure dont la complexité est de l'ordre de n^2 .

En conclusion cette approche de recherche de la solution optimale peut être très efficace lorsqu'il n'y a qu'un seul paramètre pour décrire les objets ainsi peut elle être utilisée dans les problèmes de codage de variables quantitatives et le chapitre suivant permettra d'explorer ces possibilités.

1.1.3. Programmation en nombre entier

Soit $E = \{e_1, \dots, e_n\}$ et désignons par Q_m une classe de E et $w(Q_m)$ sa valuation.

L'objectif est de trouver K classes formant une partition de E et minimisant le critère W suivant :

$$W(P) = \sum_{k=1}^K w(P_k)$$

Soit m le cardinal de l'ensemble $\{Q_1, \dots, Q_m\}$ des parties de E , ce nombre est égal à 2^n .

$$\text{Si on note } \begin{aligned} \delta_i^m &= 1 \text{ si } e_i \in Q_m \\ \delta_i^m &= 0 \text{ sinon} \end{aligned}$$

Alors, l'objectif précédent peut s'écrire comme la recherche d'un ensemble de K classes (P_1, \dots, P_K) de $P(E)$ qui rend minimum le critère W :

$$W(P) = \sum_{m=1}^M w(P_m) y_m \text{ et } P(P_1, \dots, P_k)$$

ou $y_m = 1$ si la classe Q_m est une classe de P
 $y_m = 0$ sinon

avec la contrainte

$$\sum_{m=1}^M \delta_i^m y_m = 1 \text{ pour } i = 1, \dots, n$$

car P doit être une partition de E . Malheureusement la programmation en nombre entier est inutilisable si n est plus grand que 20 car la valeur m égale à 2^n devient rapidement très grande quand n croit.

1.1.4. Procédure de coloration de graphes

Dans le cadre de la recherche des partitions de diamètre minimum Hubert (1974) définit l'équivalence de ce problème et de la recherche de sous-graphe complets maximaux dans un graphe partiel. Puis Hansen et Delattre (1978) développent une méthode optimale basée sur la coloration des graphes pour rechercher les partitions de diamètre minimum et ont démontré que les valeurs de l'écart pour les partitions d'un ensemble E sont égales au poids des arêtes de l'arbre de longueur minimum du graphe correspondant ils ont développé une méthode de classification optimale bicritère.

1.2. Recherche de la solution optimale sous contrainte d'ordre total

1.2.1. Classification automatique et programmation dynamique

Dans le cadre tracé par Fisher la programmation dynamique peut être très efficace car elle permet de trouver les partitions optimales avec un algorithme d'une complexité de n^2 (n étant le nombre d'objets). Cet algorithme a cette propriété si les éléments que l'on veut classer ne sont mesurés que par un seul paramètre et que le critère soit la variance intraclasse. Nous généraliserons cette approche en l'appliquant à des critères adaptés à l'analyse d'un tableau de contingence.

Dans la première partie de ce chapitre on reprendra l'approche de Fisher en démontrant différemment les propriétés optimales de cette méthode ceci pour nous permettre dans la deuxième partie de faire une généralisation.

1.2.2. Approche de ce problème par Fisher

Le découpage optimal au sens du critère de la variance d'une variable quantitative peut se faire par l'algorithme de programmation dynamique développé par Fisher. Ce résultat ne peut être obtenu que grâce à ce que Fisher nomme la propriété d'ordre (P.O).

Définition de la propriété d'ordre [P.O]

Soit un ensemble $E = \{e_1, e_2, \dots, e_n\}$. On dit qu'une partition P de k classes $P_1 \dots P_k$ possède la propriété d'ordre [P.O] si pour deux éléments e_i et e_j de E classés dans la même classe P_ℓ alors :

$$\forall e_k \in E \text{ tel que : } q_i < q_k < q_j \text{ alors on a } e_k \in P_\ell$$

Ainsi la partition P respecte l'ordre donné par le paramètre Q .

Notations et Formulation du problème

Les éléments sont représentés sur la droite réelle. Nous cherchons la partition qui minimise la variance intraclasse ce qui s'écrit :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^K \omega_i \delta_j^i (x_i - \bar{x}_j)^2$$

où n est le nombre d'élément de E et K est le nombre de classes, non vides, de la partition que l'on désire obtenir.

Si ω_i est le poids de l'élément e_i de E et si x_i est la valeur de l'élément e_i de E alors :

$$\delta_j^i = 1 \text{ si l'élément } e_i \text{ a été classé dans la classe } j$$

$$\delta_j^i = 0 \text{ sinon}$$

$$\text{et } \bar{x}_j = \frac{\sum_{i=1}^n \omega_i \delta_j^i x_i}{\sum_{i=1}^n \omega_i \delta_j^i} \quad (\bar{x}_j \text{ est la moyenne de la classe } j)$$

Pour simplifier la démonstration nous supposons que si $k > i$ alors $x_k > x_i$. Nous devons aussi nous assurer que $x_i \neq x_j$ si $i \neq j$. Sinon il faut regrouper e_i et e_j en un même élément en lui affectant le poids $w_i + w_j$.

Démonstration que l'algorithme proposé par Fisher permet d'obtenir la partition optimale

Fisher démontre que pour qu'une partition P puisse être optimale avec le critère de la variance intraclasse elle doit vérifier la propriété d'ordre [P.O]. (C'est-à-dire qu'elle doit être composée de K segments de droite). Puis il définit une relation de récurrence permettant de parcourir l'ensemble des partitions vérifiant la propriété d'ordre.

Pour démontrer que l'algorithme, développé par Fisher, donne bien une des meilleures solutions nous avons divisé notre démonstration en deux parties :

- 1ère partie : montrer que la partition optimale doit vérifier la propriété d'ordre [P.O].
- 2ème partie : définir un algorithme efficace qui, en parcourant l'ensemble des partitions vérifiant la propriété d'ordre [P.O] permet d'extraire dans cet ensemble la meilleure partition.

Théorème 1

La partition en k classes non vides, minimisant le critère de la variance intraclasse, doit vérifier la propriété d'ordre [P.O].

Démonstration

Supposons que la partition P , en k classes non vides minimisant le critère de la variance intraclasse, ne vérifie pas la propriété d'ordre [P.O]. Alors on peut construire une partition R en k classes non vides vérifiant la propriété d'ordre [P.O] qui améliore le critère de la variance intraclasse. Cette construction se fait ainsi :

Soient (P_1, \dots, P_k) les k classes de P , $(R_1 \dots R_k)$ les k classes de R ; on associe à chaque classe P_i de P son centre de gravité noté g_i

La partition R est construite ainsi :

On affecte x_i à R_j si $(x_i - g_j)^2 < \underset{\substack{j \neq \ell \\ \ell = 1, \dots, k}}{\text{Min}} (x_i - g_\ell)^2$

En cas d'égalité on choisit le plus petit indice.

Vérifions que R améliore le critère W de la variance intraclasse

Soit Q la partition issue de l'intersection de deux partitions P et R en k classes, on note $C_{ij} = P_i \cap R_j$ d'où $Q = (C_{11}, \dots, C_{kk})$

alors $W(P) = \sum_{i=1}^k \sum_{j=1}^k \sum_{e_h \in C_{ij}} (x_h - g_i)^2$ g_i centre de gravité de P_i

et $W(R) = \sum_{i=1}^k \sum_{j=1}^k \sum_{e_h \in C_{ij}} (x_h - g_j^*)^2$ g_j^* centre de gravité de R_j

par construction de R on peut écrire :

$$\sum_{i=1}^k \sum_{j=1}^k \sum_{e_h \in C_{ij}} (x_h - g_j)^2 < \sum_{i=1}^k \sum_{j=1}^k \sum_{e_h \in C_{ij}} (x_h - g_i)^2 = W(P) \quad [1]$$

et d'après le théorème de Huygens on a

$$W(R) = \sum_{j=1}^k \sum_{e_h \in R_j} (x_h - g_j^*)^2 < \sum_{j=1}^k \sum_{e_h \in R_j} (x_h - g_j)^2 = \sum_{i=1}^k \sum_{j=1}^k \sum_{e_h \in C_{ij}} (x_h - g_j)^2$$

et en comparant cette inégalité avec l'inégalité [1] on a :

$$W(R) = \sum_{j=1}^k \sum_{e_h \in R_j} (x_h - g_j^*)^2 < \sum_{i=1}^k \sum_{e_h \in P_i} (x_h - g_i)^2 = W(P)$$

Ainsi $W(R) < W(P)$.

donc R améliore le critère. Ainsi il est toujours possible à partir d'une partition ne vérifiant pas la propriété d'ordre de construire une partition vérifiant cette propriété et dont la variance interclasse est plus petite.

Rappel des démonstrations faites par Fisher

Il suffit de démontrer que l'affectation des points de E au centre de gravité le plus proche découpe notre droite en k segments.

Si $e_h \in E$ a été classé dans P_j ceci peut s'exprimer par

$$\begin{aligned} \forall_i \neq j \quad (x_h - g_i)^2 &> (x_h - g_j)^2 \text{ et aussi par} \\ \forall_i \neq j \quad |x_h - g_i| &> |x_h - g_j| \end{aligned}$$

R ne vérifie pas la propriété d'ordre cela veut dire qu'il existe un élément e_q de E vérifiant $g_j^* < x_q < x_h$ et que cet élément n'est pas classé dans R_j mais e_h est dans R_j .

Démontrons que cette affirmation est fautive : l'élément x_q est classé dans la classe R_i alors le centre de gravité g_i^* de cette classe ne peut se situer que dans les trois cas suivants :

Cas 1 : $g_j^* < g_i^* < x_h$

Cas 2 : $g_i^* < g_j^*$

Cas 3 : $g_i^* > x_h$

Cas 1 : impossible car la règle d'affectation de e_h implique que

$$|x_h - g_j^*| < |x_h - g_i^*|$$

Cas 2 : on a

$$x_q - g_i^* = (x_q - g_j^*) + (g_j^* - g_i^*) \text{ et comme } g_i^* < g_j^*$$

ceci entraîne

$$(x_q - g_j^*) < (x_q - g_i^*)$$

ce qui est impossible d'après la règle d'affectation de e_q

Cas 3 : on a

$$(g_i^* - x_q) = (g_i^* - x_h) + (x_h - x_q) \text{ et } x_h > x_q$$

ceci entraîne $(g_i^* - x_q) > (g_i^* - x_h)$

et comme e_h est classé dans R_j on a

$$(g_i^* - x_h) > (x_h - g_j^*)$$

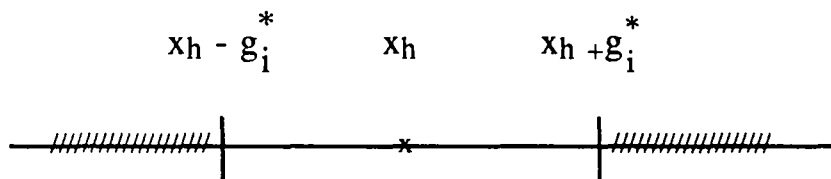
d'où

$$(g_i^* - x_q) > (g_j^* - x_q) \text{ ce qui est impossible.}$$

Démonstration géométrique

Si $|g_j^* - x_h| < \text{Min}_{i \neq j} |g_i^* - x_h|$ les autres centres de gravité g_i doivent se trouver

dans les zones hachurées ci-dessous :



Théorème 2

L'algorithme proposé par Fisher permet d'obtenir la partition optimale au sens de la variance interclasse.

Démonstration

Recherche de la partition optimale en deux classes :

L'énumération complète de l'ensemble des partitions en 2 classes non vides respectant la propriété d'ordre [P.O] se fait ainsi :

- On construit la première partition $(\{e_1\}, \{e_2...e_n\})$ et on calcule le critère de la variance pour la classe $\{e_2, \dots, e_n\}$ (nous noterons cette valeur $F_1(2)$). La construction de la $j^{\text{ème}}$ partition se fait ainsi :

$(\{e_1, \dots, e_j\}, \{e_{j+1}, \dots, e_n\})$. On calcule la variance de la classe $\{e_{j+1}, \dots, e_n\}$ soit $F_1(j+1)$.

Et on continue ainsi de suite jusqu'à la partition $(\{e_1, \dots, e_{n-1}\}, \{e_n\})$.

Nous avons construit $n-1$ partitions en deux classes respectant la propriété d'ordre [P.O]. Cet ensemble de partitions est l'ensemble des partitions en deux classes respectant la propriété d'ordre. Donc, d'après le théorème précédent, la partition optimale est celle qui minimise le critère de la variance intraclasse parmi ces $n-1$ partitions.

Recherche d'une partition optimale en k classes :

On notera $w(B)$ la variance de la classe $B \subset E$. Le critère de la variance intraclasse d'une partition P s'écrit alors :

$$W(P) = \sum_{j=1}^k w(B_j)$$

où $B_1 \dots B_k$ sont les classes de P avec $B_j = \{e_1 \dots e_j\}$

Propriété fondamentale

Si la partition $P = (\{e_1, \dots, e_i\}, B_2, \dots, B_k)$ est optimale pour k classes alors la partition (B_2, \dots, B_k) est une partition optimale en k-1 classes de l'ensemble $\{e_{i+1}, \dots, e_n\}$.

Démonstration

Supposons que cette propriété n'est pas vérifiée alors la partition (A_2, \dots, A_k) optimale en k-1 classes de cet ensemble différente de (B_2, \dots, B_k) et elle vérifie :

$$\sum_{j=2}^k w(A_j) < \sum_{j=2}^k w(B_j)$$

Alors la partition Q construite ainsi :

$$Q = (B_1, A_2 \dots A_k) \text{ vérifie}$$

$$W(Q) = w(B_1) + \sum_{j=2}^k w(A_j) < \sum_{j=1}^k w(B_j) = W(P)$$

donc $(B_1, A_2 \dots A_k)$ est la partition optimale en k classes de l'ensemble $\{e_1, \dots, e_n\}$ ce qui est contraire à l'hypothèse.

Conclusion :

Ainsi la partition optimale en k classes appartient bien à l'ensemble des partitions suivantes : $(\{e_1\}, 0_2), \dots, (\{e_1, \dots, e_{n-k+1}\}, 0_{n-k+2})$.

Où $0_2, \dots, 0_{n-k+2}$ sont les partitions optimales, des sous ensembles de E , construites à l'étape k-1. Ainsi pour l'étape k il faut rechercher la partition C_i en k-1 classes optimales du sous ensemble $\{e_i, \dots, e_n\}$ en parcourant l'ensemble suivant :

$$(\{e_i, \dots, e_j\}, 0_{j+1}), \dots, (\{e_i, \dots, e_{n-k+1}\}, 0_{n-k+2})$$

Cette recherche peut être faite par la programmation dynamique. L'algorithme proposé par Fisher et développé dans Hartigan permet de résoudre ce problème.

1.2.3. Exemple d'application

L'exemple décrit dans le livre de Chandon et Pinson peut être traité par l'algorithme proposé par Fisher.

L'exemple comprend 6 objets $\{x_1, \dots, x_6\}$

1ère étape

La première étape consiste à calculer l'inertie C_{ij} de la classe $\{x_i, \dots, x_j\}$. Cet ensemble comprend $n(n+1)/2$ éléments dont n éléments n'ont qu'un seul point.

C	1	2	3	4	5	6
1	0	0,50	8,67	22,75	41,80	67,33
2	0	0	4,50	12,67	27,75	41,20
3			0	2,00	8,00	14,75
4				0	2,00	4,67
5					0	0,50
6						0

Tableau des C_{ij}

$F_{j+1}(k-2)$ est la solution optimale d'un découpage en $k-2$ classes de l'intervalle $[x_{j+1}, x_n]$

$$F_j(k-1) = \underset{j \leq i \leq N-1}{\text{Min}} \{C_{ji} + F_{i+1}(k-2)\} = \underset{j \leq i \leq N-1}{\text{Min}} F_{k-1}(j, i)$$

A partir de la relation de récurrence $W_k(i, j) = C_{1i} + C_{i+1, j} + F_{j+1}(k-2)$ cet algorithme calcule de proche en proche, toutes les partitions optimales de k classes en recherchant la solution vérifiant :

$$W_k^* = \underset{1 \leq i \leq n}{\text{Min}} W_k(i) = \underset{\substack{1 \leq i \leq n \\ i+1 \leq j \leq n-1}}{\text{Min}} W_k(i, j)$$

Aussi la recherche de la solution optimale en deux classes nécessite le calcul des ces 5 valeurs.

Au départ $F_j(0) = 0$ pour $j = 1, N$

Pour $k=2$ nous avons

i	Solutions en 2 classes	$W_2(i)$	$F_i(1)$
2	(1) (2,3,4,5,6)	41,20	41,20
3	(1,2) (3,4,5,6)	15,25	14,75
4	(1,2,3) (4,5,6)	13,34	4,67
5	(1,2,3,4) (5,6)	23,3	0,50
6	(1,2,3,4,5) (6)	44,8	0

Cette solution optimale est (1,2,3) (4,5,6) car

$$W_2^* = \min_{2 \leq i \leq 6} W_2(i) = 13,34$$

Après cette étape d'initialisation pour $k=3$ nous avons :

Pour i variant de 1 à $n-2$ et j variant de $i+1$ à $n-1$

i	j	Solution en 3 classes	Critère $W_3(i,j)$	Tableau $F_2(i,j)$
1	2	(1) (2) (3,4,5,6)	14,75	14,75
1	3	(1) (2,3) (4,5,6)	9,17	9,17
1	4	(1) (2,3,4) (5,6)	13,17	13,17
1	5	(1) (2,3,4,5) (6)	26,75	26,75
2	3	(1,2) (3) (4,5,6)	5,17	4,67
2	4	(1,2) (3,4) (5,6)	3,00	2,5
2	5	(1,2) (3,4,5) (6)	8,5	8,0
3	4	(1,2,3) (4) (5,6)	9,17	0,5
3	5	(1,2,3) (4,5) (6)	10,67	2
4	5	(1,2,3,4) (5) (6)	29,75	0

d'où la solution optimale en 3 classes est la partition (1,2) (3,4) (5,6) et les solutions optimales $O_2^{i,2}$ des segments $[x_i, x_n]$ en deux classes sont :

i	O_2^{i+1}	$F_i(2) = \min_j F_2(i,j)$
1	$O_2^2 = (2,3) (4,5,6)$	$9,17 = \min \{14,75, 9,17, 13,17, 26,75\}$
2	$O_2^3 = (3,4) (5,6)$	$2,5 = \min \{4,67, 2,5, 8,0\}$
3	$O_2^4 = (4) (5,6)$	0,5
4	$O_2^5 = (5) (6)$	0

Pour $k = 4$ on obtient le tableau suivant :

i	j	Solution 4 classes	Critère $W_4(i,j)$	Tableau $F^3(i,j)$
1	2	(1) (2) O_2^3	2,5	2,5
1	3	(1) (2,3) O_2^4	5,0	5,0
1	4	(1) (2,3,4) O_2^5	12,67	12,67
2	3	(1,2) (3) O_2^4	1	0,5
2	4	(1,2) (3,4) O_2^5	2,5	2
3	4	(1,2,3) (4) O_2^5	22,75	0

d'où la solution optimale en 4 classes est la partition (1,2) (3) (4) (5,6) car

$$W_4^* = \min_i W_k(i) = \min_i \{5,0,2,5,22,75\}$$

Le tableau résumant les solutions optimales est le suivant :

Nombre de classes	Critère	Solution optimale en k classes
1	67,33	(1,2,3,4,5,6)
2	13,33	(1,2,3) (4,5,6)
3	3,0	(1,2) (3,4) (5,6)
4	1,0	(1,2) (5) (7) (9,10)
5	0,5	(1,2) (5) (7) (9) (10)
6	0	(1) (2) (3) (4) (5) (6)

1.2.4. Amélioration algorithmique

La démonstration précédente prouve que l'algorithme de Fisher permet de trouver la population optimale d'un découpage d'une variable quantitative. Mais le stockage de la matrice C_{ij} représentant la variance de la classe $\{e_i, \dots, e_j\}$ limite les possibilités de l'algorithme développé par Fisher car il nécessite la mise en mémoire de $n(n-1)/2$ valeurs d'où la limitation, donnée par Hartigan, une centaine d'éléments. Cette limite a été reprise par H. Spath en 1980.

Cet algorithme nécessite $(n+(n-1)/2)(k-3) + (n-1)$ calculs du critère et le stockage de $n*(n+1)/2$ valeurs de la matrice C_{ij} plus n valeurs pour le tableau F et un tableau de $n*(k-2)$ valeurs stockant les solutions optimales.

Nous proposons une modification de l'algorithme qui évite le stockage de la matrice C_{ij} . Ainsi l'utilisation de cette méthode est possible pour plusieurs milliers de points au lieu de la centaine. Le stockage de la matrice C_{ij} peut être évité en calculant à chaque étape les valeurs de la matrice C_{ij} nécessaire à l'actualisation de la série récurrente $W_k(i, j)$. Ce calcul est court s'il existe trouver une relation simple entre C_{li} et C_{i+1j} avec C_{li+1} et C_{i+2j} . Comme C_{ij} est l'inertie de la classe $\{e_j, \dots, e_j\}$ nous avons

$$C_{li} + C_{i+1j} = \sum_{m=1}^i (\overline{x_m} - \overline{x_1})^2 + \sum_{m=i+1}^j (\overline{x_m} - \overline{x_2})^2$$

où $\overline{x_k}$ est le centre de gravité de la classe k . Nous avons aussi

$$C_{li} + C_{i+1j} = \sum_{m=1}^i x_m^2 - n_1 \overline{x_1}^2 + \sum_{m=i+1}^j x_m^2 - n_2 \overline{x_2}^2$$

où n_k est l'effectif de la classe k d'où

$$C_{li} + C_{i+1j} = \sum_{m=1}^j x_m^2 - n_1 \overline{x_1}^2 - n_2 \overline{x_2}^2$$

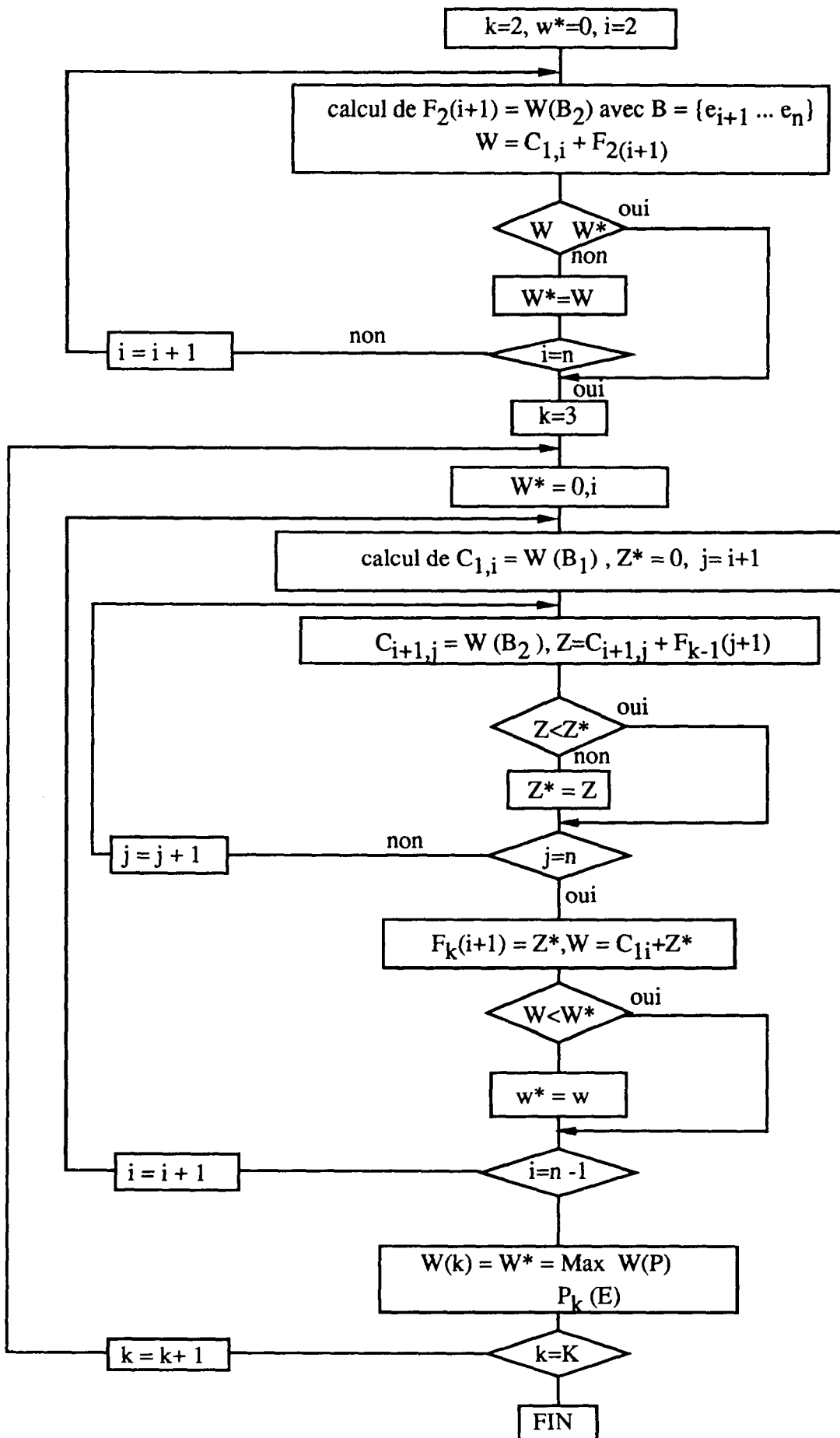
Comme $T = \sum_{m=1}^j x_m^2$ est constant, la recherche du minimum de $C_{li} + C_{i+1j}$ en

faisant varier i revient à chercher le maximum de $\left(\sum_{m=1}^i x_m\right)^2 / n_1 + \left(\sum_{m=i+1}^j x_m\right)^2 / n_2$

Si on note $A_i = \sum_{m=1}^i x_m$ et $B = \sum_{m=1}^n x_m$ alors

$$C_{li+1} + C_{i+2j} = T - (A_i + x_{i+1})^2 / (n_1+1) - (B - A_i - x_{i+1})^2 / (n_2-1)$$

ce qui permet de calculer rapidement à partir de la connaissance de A_i et de x_{i+1} la valeur $C_{li+1} + C_{i+2j}$, d'où l'organigramme suivant :



1.2.5. Généralisation de Vinod

Les éléments sont maintenant des points dans l'espace euclidien de dimension p et le critère est celui de la variance intraclasse. Pour résoudre ce problème Vinod généralise la propriété d'ordre de Fisher de la manière suivante :

Définition de la propriété d'ordre généralisée [G.P.O.] :

Soit l'ensemble $E = \{e_1, \dots, e_n\}$, on associe à cet ensemble une matrice des distances D

$$D = (d_{ij})_{\substack{i=1,n \\ j=1,n}}$$

Les valeurs de la $j^{\text{ème}}$ colonne peuvent être ordonnées ainsi

$$d_{jj} < d_{ji_1} \dots < d_{ji_{n-1}}$$

Nous avons pour chaque élément e_j un ordre θ_j (pour l'élément e_j nous avons l'ordre $j, i_1, i_2, \dots, i_{n-1}$). Une partition P vérifie la propriété d'ordre généralisée si pour chaque ordre θ_j elle vérifie la propriété d'ordre [P.O]. C'est à dire que la partition P doit respecter pour chaque élément e_j son ordre θ_j associé.

Cette généralisation ne donne qu'une condition suffisante. C'est à dire que la partition optimale appartient à l'ensemble des partitions vérifiant cette condition d'ordre généralisée mais cet ensemble peut être vide. Rao en 1971 donne l'exemple suivant.

Soient quatre points dans le plan :

$$x_1 = (1,2), x_2 = (2,3), x_3 = (1,1) \text{ et } x_4 = (1,0)$$

d'où le tableau des carrés des distances suivant :

	x_1	x_2	x_3	x_4
x_1	0	2	1	4
x_2	2	0	5	10
x_3	1	5	0	1
x_4	4	10	1	0

La partition optimale au sens du critère de la variance est $\{\{x_1, x_2\}, \{x_3, x_4\}\}$ mais cette partition ne vérifie pas la condition d'ordre généralisée donnée par Vinod. Dans ce cas il n'y a pas de partition vérifiant cette condition.

Nous allons donner quelques exemples de tableaux de distances où il existe une partition qui vérifie la condition d'ordre de Vinod.

Distances ultramétriques

Définition :

Une distance ultramétrique est une distance satisfaisant à l'inégalité suivante :

$$\forall x, y, z \quad d(x, y) \leq \text{Max} \{d(x, z), d(z, y)\} \quad \blacksquare$$

Avec un seuil tel que $\alpha < \text{Max}_{i,j} d(x_i, x_j)$ et $B(x, \alpha)$ la boule fermée de centre x et de rayon α on définit sur $E \times E$ la relation d'équivalence suivante

$$xRy \iff B(x, \alpha) = B(y, \alpha)$$

Proposition :

Si d est une distance ultramétrique alors cette relation d'équivalence induit sur E une partition.

Démonstration :

Il faut montrer que pour tout $z \in B(y, \alpha)$ alors $z \in B(x, \alpha)$. Supposons que $z \in B(y, \alpha)$ et $z \notin B(x, \alpha)$ alors on a $d(x, z) > \alpha$

comme $y \in B(x, \alpha)$ on a $d(x, y) \leq \alpha$ et comme $z \in B(y, \alpha)$ on a $d(y, z) \leq \alpha$ d'où $d(x, z) \leq \text{Max} \{d(x, y), d(y, z)\} \leq \alpha$

donc $z \in B(x, \alpha)$ ce qui est contradictoire à l'hypothèse. \blacksquare

Théorème :

Soit d une ultramétrique sur $E \times E$ alors il existe, au moins, une partition de E qui vérifie la condition d'ordre de Vinod.

Démonstration :

Soit la relation d'équivalence suivante

$$xRy \iff B(x, \alpha) = B(y, \alpha)$$

alors nous avons vu, par la proposition précédente, qu'elle induit sur E une partition P en K classes. Nous allons démontrer que cette partition P vérifie la condition d'ordre de Vinod. Pour démontrer cela il faut que cette partition respecte tous les ordres θ_j associés aux éléments x_j de E . Il suffit de démontrer que pour une classe quelconque P_k de P et deux éléments x, z de P_k il n'existe pas d'éléments $y \notin P_k$ qui, pour un ordre donné θ_i , vérifie les conditions suivantes :

$$d(x_i, x) \leq d(x_i, y) \leq d(x_i, z) \quad \text{et} \quad d(x_i, x) < d(x_i, z)$$

1. Supposons que $x_i \in P_k$

Dans ce cas $d(x_i, y) \leq d(x_i, z) \leq \alpha$ puisque d est une ultramétrie. Comme $d(y, x_i) \leq \alpha$ alors $y \in B(x_i, \alpha) = B(z, \alpha)$ donc, d'après la proposition précédente, $y \in P_k$ ce qui est en contradiction avec l'hypothèse.

2. Supposons que $x_i \notin P_k$

Dans ce cas nous avons $\alpha < d(x_i, x) \leq d(x_i, y) \leq d(x_i, z)$ et comme d est une ultramétrie nous avons

$$d(x_i, x) \leq \text{Max} \{d(x_i, z), d(x, z)\} = d(x_i, z)$$

et

$$d(x_i, z) \leq \text{Max} \{d(x_i, x), d(x, z)\} = d(x_i, x)$$

d'où $d(x_i, x) = d(x_i, z)$ ce qui est en contradiction avec l'hypothèse.

■

Remarque :

Comme il existe au moins une partition de E qui vérifie la condition d'ordre de Vinod, alors la solution optimale appartient à cet ensemble. Ainsi, cet algorithme permet de trouver les partitions optimales, au sens du critère de la variance, d'une matrice de distances ultramétriques.

1.2.6. Généralisation

Définitions et énoncé du problème

Soient un ensemble E de N éléments : $E = \{e_1, \dots, e_n\}$ et un ordre θ total sur E , ainsi l'ensemble $\{e_{\theta_1}, \dots, e_{\theta_n}\}$ est totalement ordonné. Pour simplifier les démonstrations nous supposons que l'ensemble $E = \{e_1, \dots, e_n\}$ est totalement ordonné.

Soient w une application de $\mathcal{P}(E)$ dans \mathbf{R}^+ , cette application donne une évaluation de toutes les parties de E et un critère W qui est défini par

$$W(P) = \sum_{i=1}^k w(P_i) \text{ où } P_i \in \mathcal{P}(E)$$

On appelle $Q(E)$ l'ensemble des parties de E vérifiant la propriété $[P_1]$ suivante :

$\forall e_i, e_j \in C$ avec $i < j$ et si $i < h < j$ alors $e_h \in C$

$Q(E) = \{C \in \mathcal{P}(E), C \text{ vérifiant la propriété } P_1\}$

On appelle $Q_k(E)$ l'ensemble des partitions de E en K classes non vides vérifiant la propriété P_1 .

Problème :

Rechercher une partition P de $Q_k(E)$ optimale au sens du critère W .

Nous dirons qu'une partie de E est **admissible** si elle vérifie la propriété P_1 ($Q(E)$ est l'ensemble des parties admissibles de E), et qu'une partition de E est **admissible** si toutes ces classes vérifient la propriété P_1 . $Q_k(E)$ est l'ensemble des partitions admissibles en k classes non vides de E et cet ensemble est non vide si $k < n$.

Nous notons $F = \{\{e_1\}, \{e_1, e_2\}, \dots, \{e_1, \dots, e_n\}\}$ qui est aussi
 $F = \{C \in Q(E) / e_1 \in C\}$

et $Q_k^*(A)$ une solution optimale au sens du critère W , cette solution est une partition admissible de $A \in Q(E)$ en k classes non vides.

$$Q_k^*(A) = \underset{P \in Q_k(A)}{\text{Min}} W(P) \text{ avec } A \in Q(E)$$

Théorème 1

L'ensemble $Q_{k+1}(A)$ avec $A \in F$ est l'ensemble des partitions de $(k+1)$ classes non vides de A dont les k premières classes forment une partition appartenant à $Q_k(B)$ avec $B \subset A$ et $B \in F$ et la dernière classe est $(A-B)$.

Démonstration

Comme $A \in F$ alors A s'écrit $\{e_1, \dots, e_i\}$ avec $1 \leq i \leq n$

Soit $Q \in Q_{k+1}(A)$, comme la dernière classe Q_{k+1} de Q vérifie la propriété P_1 alors il existe un indice ℓ tel que $Q_{k+1} = \{e_\ell, \dots, e_i\}$. Ainsi (Q_1, \dots, Q_k) est une partition en k classes non vides de l'ensemble $\{e_1, \dots, e_{\ell-1}\}$ et ces k classes vérifient la propriété P_1 donc, par définition cette partition (Q_1, \dots, Q_k) appartient à $Q_k(\{e_1, \dots, e_{\ell-1}\})$. Si on note $B = \{e_1, \dots, e_{\ell-1}\}$ alors Q est bien formé d'une partition $H \in Q(B)$ et de l'ensemble $A-B$ donc

$$Q_{k+1}(A) = \{(H, A-B) / H \in Q_k(B) \text{ et } B \subset A, B \in F\}$$

Soit $Q = (H, A-B)$ avec $H \in Q_k(B)$ et $B \subset A$ et $B \in F$, donc Q est une partition en $(k+1)$ classes non vides de A . Comme A et $B \in F$ il existe deux indices i et j , $j < i$ avec $A = \{e_1, \dots, e_i\}$ et $B = \{e_1, \dots, e_j\}$.

$A-B = \{e_{j+1}, \dots, e_i\}$ vérifie la propriété P_1 donc Q est une partition en $(k+1)$ classes non vides de A avec ses k premières classes vérifiant, la propriété P_1 et sa $(k+1)$ ème classe vérifiant aussi la propriété P_1 donc $Q \in Q_{k+1}(A)$.

Théorème 2

Si les partitions $Q_k^*(A)$ avec $A \in F$ sont connues alors

$$Q_{k+1}^*(A) = \text{Min } W(P), P \in \{(H, A-B) / H = Q_k^*(B) \text{ et } B \subset A, B \in F\}$$

est une solution optimale admissible de $Q_{k+1}(A)$.

Démonstration :

Il faut démontrer qu'une solution optimale admissible de $Q_{k+1}(A)$ est un élément de l'ensemble V défini par :

$$V = \{(H, A-B) / H = Q_k^*(B) \text{ et } B \subset A, B \in F\}.$$

Comme $A \in F$ alors A s'écrit $\{e_1, \dots, e_i\}$ avec $i = 1, \dots, n$

$$V = \{(Q_k(\{e_1, \dots, e_j\}), \{e_{j+1}, \dots, e_i\}) / 1 \leq j < i-1\}$$

Si on note Q la solution optimale de $Q = \text{Min}_{P \in V} W(P)$

alors sa valeur $W(Q)$ est obtenue par

$$W(Q) = \text{Min}_{j=1, i-1} W(Q_k^*(\{e_1, \dots, e_j\})) + w(\{e_{j+1}, \dots, e_i\})$$

Ainsi une solution optimale admissible de $Q_{k+1}(A)$ est un élément de V .

1.2.6.3. Algorithme DECOPT

Le théorème 2 démontre que les solutions optimales en $(k+1)$ classes sont formées par les solutions optimales en k classes, ceci est conforme au principe de la programmation dynamique, d'où l'algorithme suivant :

Initialisation

Pour $k = 1$ alors $Q_1(A)$ est formé d'un seul élément, c'est l'ensemble $A = \{e_1, \dots, e_i\}$

Pour $k = 2$ comme $Q_1(A)$ est formé d'un seul élément nous pouvons construire $Q_2(A)$ comme ceci

$$Q_2(A) = \text{Min}_{j=1, i-1} \{w(\{e_1, \dots, e_j\}) + w(\{e_{j+1}, \dots, e_i\})\}$$

Et la solution optimale d'un découpage en deux classes de l'ensemble E est donné par $Q_2(E)$

Etape k

Supposons $\{Q_k(A), A \in F\}$ connues nous allons construire l'ensemble $\{Q_{k+1}(A), A \in F\}$ ainsi

$$Q_{k+1}(A) = \text{Min}_{j=1, i-1} \{W(Q_k(\{e_1, \dots, e_j\})) + w(\{e_{j+1}, \dots, e_i\})\}$$

Et la solution optimale d'un découpage en $(k+1)$ classes de l'ensemble E est donné par $Q_{k+1}(E)$.

Remarque :

La complexité de cet algorithme est $(k-1)n^2$ pour $k > 2$. Car à chaque étape nous recherchons pour tous les éléments de F la solution optimale. Cette solution est parmi, les n solutions possibles. Le cardinal de l'ensemble F est n et à l'étape k nous avons la solution en $(k+1)$ classes.

Pour $k = 2$ nous n'avons que $(n-1)$ cas possibles. L'organigramme de cet algorithme s'écrit ainsi :

Algorithme

Etape 1

Soit l'ensemble $E = \{e_1, \dots, e_n\}$ composé de n éléments. A chaque élément e_i est associé un numéro d'ordre θ_i . On construit, à partir des numéros d'ordre θ_i , les m coupures possibles rangées suivant l'ordre total croissant $\{X_1, \dots, X_m\}$. On donne une valuation w sur tous les éléments de $P(E)$.

On lit les deux paramètres K et L_{MIN}

où K est le nombre de classes

L_{MIN} est le nombre minimum d'éléments dans une classe.

Etape 2

On calcule les tableaux $card$, t et tk de la manière suivante :

```
FAIRE pour  $i = 1, n$ 
     $card(i) = \text{cardinal } \{e_j \in E / \theta_j \leq X_i\}$ 
     $tk(2,i) = i$ 
SI  $(n - card(i) \geq L_{MIN})$  ALORS
     $T(i) = w(\{e_\ell \in E / \theta_\ell > X_i\})$ 
SINON  $T(i) = -1$ 
FIN
```

Etape 3

Recherche de la solution optimale en deux classes :

```
FAIRE pour  $k = 2$ 
    Rechercher la valeur  $w(2)$  tel que
     $W(2) = \text{Min}_{j=1,n} \{w(\{e_\ell \in E / \theta_\ell > X_i\}) + t(j)\}$ 
    sous les conditions que  $T(j) \neq -1$  et  $card(i) \geq L_{MIN}$ 
    et mettre dans  $v(2)$  l'indice  $j$  de la valeur minimale  $W(2)$ 
FIN
```

Etape 4

Rechercher les solutions optimales de 3,4,...,K classes :

```
FAIRE pour  $k = 3$  à  $K$ 
    FAIRE pour  $i = 1, n$ 
    Rechercher  $tk(1,i) = \text{Min}_{j=i+1,n} \{w(\{e_\ell \in E / X_i < \theta_\ell \leq X_j\}) + t(j)\}$ 
    sous les conditions que  $T(j) \neq -1$  et  $card(j) - card(i) \geq L_{MIN}$ 
    SI les conditions précédentes sont jamais vérifiées
        mettre  $tk(1,j) = -1$ 
    SINON mettre dans  $tk(k,i)$ 
        l'indice de la valeur minimale de  $tk(1,i)$ .
```

SI (tk(1,j) = -1) ALORS tk(1,i)=0.

FIN

Puis rechercher W(k) :

$$W(k) = \text{Min}_{i=1,n} \{w(\{e_\ell \in E / \theta_\ell \leq X_i\}) + tk(1,i)\}$$

sous les conditions que tk(1,i) ≠ -1 et card(i) ≥ LMIN

SI les conditions précédentes sont jamais vérifiées aller à l'étape 5.

FAIRE pour i = 1,n

t(i) =tk(1,i)

FIN

Etape 5

Etape finale

Le tableau W(k), k = 2, K contient pour les solutions optimales de 2 à K classes, la valeur du critère.

Pour obtenir les coupures il faut utiliser les tableaux tk et v de la manière suivante :

Pour la partition en k classes les valeurs des coupures sont :

lire la valeur v(k), cet indice correspond à l'indice de la coupure X(v(k)) de la première classe.

i₀ = W(k)

FAIRE pour m = K-1 à 1

la lire la valeur i de tk(m, i₀) qui est l'indice de la coupure X(i) de (k-m+1)^{ième} classe.

Mettre i dans i₀

FIN

Remarque :

Les tableaux tk et v ne sont utiles que si les valeurs des coupures des partitions optimales sont demandées.

On remarque que la complexité de l'algorithme dépend du calcul de $w(\{e_\ell \in E / \theta_\ell \leq X_{i+1}\})$.

Si ce calcul se déduit facilement, comme en 1.2.3, de la valeur $w(\{e_\ell \in E / \theta_\ell \leq X_i\})$ alors l'algorithme devient performant et utilisable pour de grands tableaux de données. Sinon il faut mémoriser les valeurs $w(\{e_\ell \in E / X_i \leq \theta \leq X_j\})$ pour i = 1,n et j = 1,n ce qui limite son utilisation sur de grands tableaux.

2. UTILISATION DE LA PARTITION OPTIMALE OBTENUE SOUS CONTRAINTE D'ORDRE LOCAL

Dans ce chapitre nous allons présenter plusieurs applications, de la méthode décrite dans le chapitre précédent.

La première application est celle proposée par Fisher (1958). Avec quelques simulations de distributions gaussiennes nous verrons les limites de cette stratégie.

La deuxième application est le découpage d'une variable quantitative en fonction d'une variable qualitative. Cette stratégie est intéressante si on désire retrouver, par l'analyse factorielle des correspondances, des relations non linéaires entre les variables. Elle permet, aussi, de rendre homogène un tableau de données en discrétisant toutes les variables continues et, ainsi, de pouvoir utiliser les méthodes de classification applicables sur des variables qualitatives.

La troisième application permet le découpage d'une variable totalement ordonnée (par exemple la profondeur, le temps...) en fonction d'un ensemble de variables explicatives. Cette stratégie a pour objectif de rechercher les zones de rupture ou de changement d'états, dans ce cas la valeur de la coupure est un indicateur important de l'analyse.

La quatrième application a un but similaire à la troisième application. Dans ce cas l'approximation est un modèle polynomial avec une continuité de l'approximation aux coupures.

2.1. Découpage d'une variable quantitative

2.1.1. Introduction

Nous nous plaçons ici dans le cadre défini par Fisher. En utilisant notre algorithme nous pouvons, sans difficulté, découper des populations de plusieurs milliers d'individus.

Le critère minimisé est celui de l'inertie intraclasse. Dans ce paragraphe nous analysons les possibilités de cette méthode en simulant des distributions gaussiennes.

Après nous donnerons des stratégies d'utilisation de cette méthode dans le cadre de l'analyse factorielle. Ces stratégies montreront l'intérêt de cette

méthode dans la transformation des variables quantitatives en variables qualitatives. Cette transformation permet, en outre, de rechercher les liaisons non linéaires entre ces variables.

2.1.2. Simulation des distributions gaussiennes

Les buts de cette simulation sont de pouvoir :

- vérifier la stabilité des bornes obtenues,
- analyser l'évolution du critère d'inertie en fonction du nombre de classes demandé,
- analyser les variations des bornes en fonction des effectifs des composants du mélange.

Une série simulée est déterminée par les quatre paramètres suivants :

- le nombre de lois gaussiennes,
- l'effectif de chaque loi gaussienne,
- la moyenne de chaque loi gaussienne,
- l'écart-type de chaque loi gaussienne.

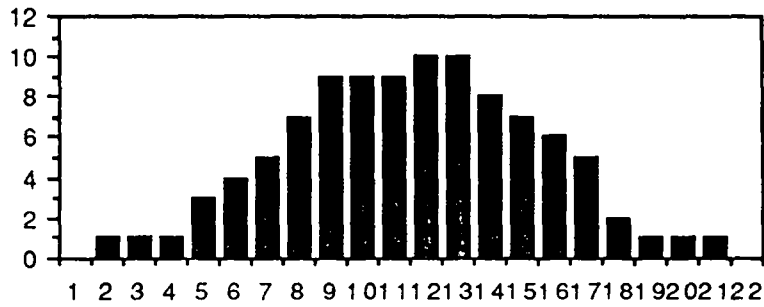
Pour mesurer la stabilité des bornes obtenues nous allons prendre, pour chaque expérience, 20 échantillons d'un mélange donné. Les composantes de ce mélange seront de même effectif et cet effectif sera de 20, 50, 100, 200, 500 éléments. Nous décrivons un mélange par la formule suivante

$$\sum_{i=1}^k n_i N(m_i, \sigma_i)$$

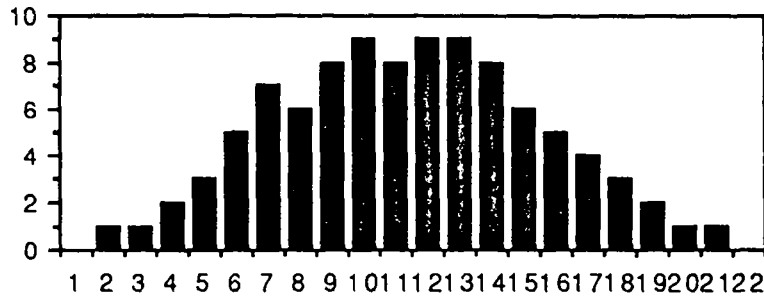
avec n_i l'effectif, m_i la moyenne, σ_i l'écart-type du $i^{\text{ème}}$ composant

Pour cette étude trois type de mélanges ont été retenus.

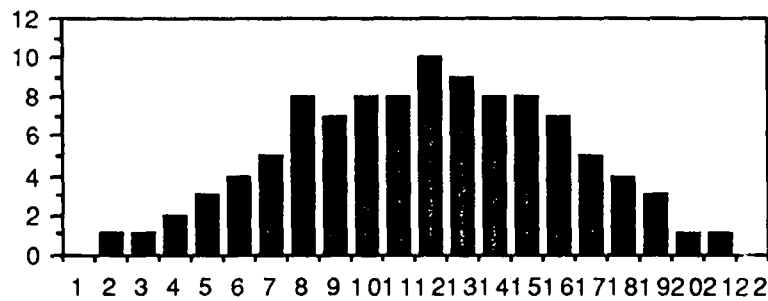
Le premier mélange est constitué de composants très proches. L'écart entre les moyennes de deux composants voisins est égal à un écart-type. Nous avons les trois échantillons suivants que noterons M_{12} , M_{13} et M_{14}



$$M_{12} = n N(1,1) + n N(2, 1)$$

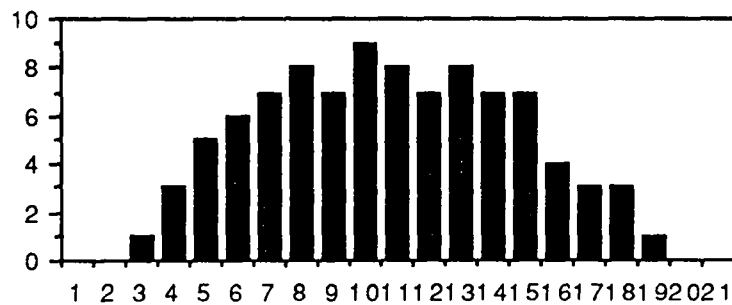


$$M_{13} = n N(1,1) + n N(2,1) + n N(3,1)$$

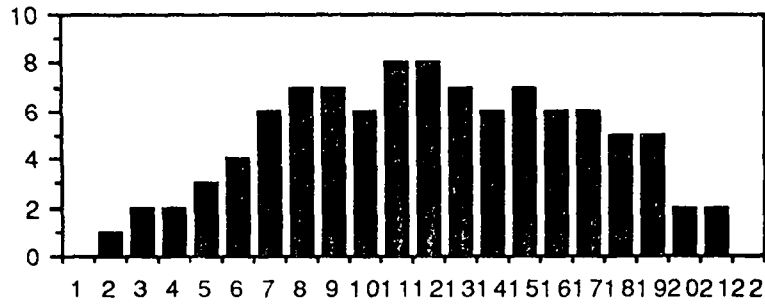


$$M_{14} = n N(1,1) + n N(2,1) + n N(3,1) + n N(4,1)$$

Le deuxième type de mélange est constitué de composants dont la différence des moyennes entre deux composants voisins est égale au double de l'écart-type. Nous avons les trois échantillons suivants que nous noterons M_{22} , M_{23} et M_{24} .

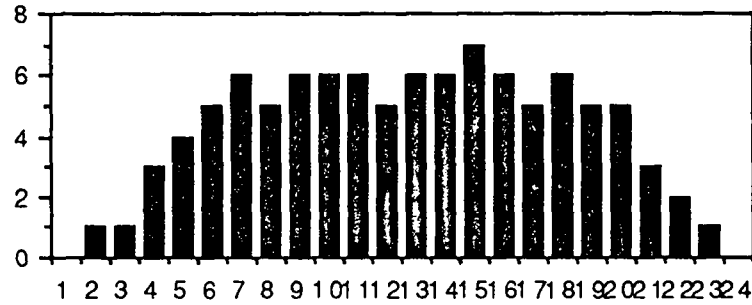


$$M_{22} = n N(1, 0.5) + n N(2, 0.5)$$



■ densité M23

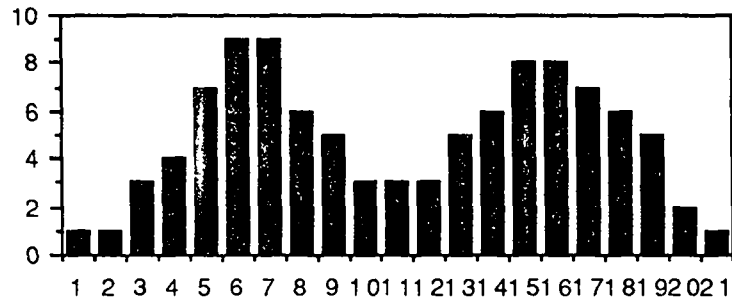
$$M_{23} = n N(1, 0.5) + n N(2, 0.5) + n N(3, 0.5)$$



■ densité M24

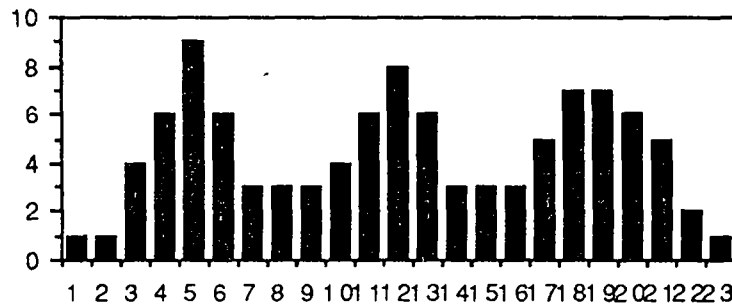
$$M_{24} = n N(1, 0.5) + n N(2, 0.5) + n N(3, 0.5) + n N(4, 0.5)$$

Le troisième mélange est constitué de composants éloignés entre eux car la différence entre les deux moyennes de composants voisins est égale à quatre fois l'écart-type. Nous avons les trois échantillons suivants que nous noterons M_{32} , M_{33} et M_{34} .



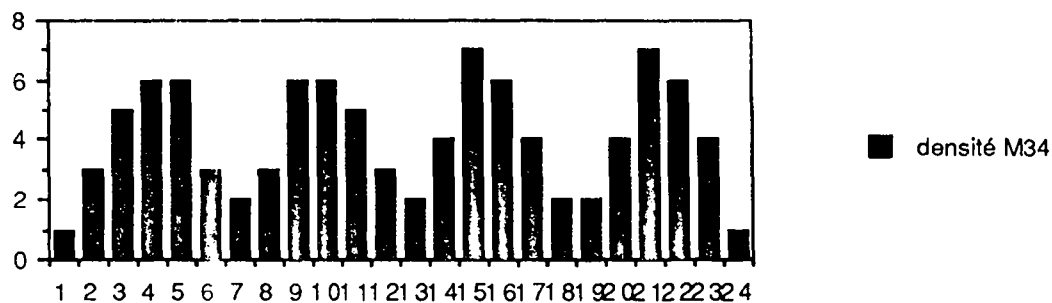
■ densité M32

$$M_{32} = n N(1,1) + n N(5,1)$$



■ densité M33

$$M_{33} = n N(1,1) + n N(5,1) + n N(9,1)$$



$$M_{34} = n N(1,1) + n N(5,1) + n N(9,1) + n N(13,1)$$

Les solutions optimales minimisent le critère de l'inertie intraclasse (notée W). Afin de juger la qualité de ces solutions de manière indépendante de l'effectif de l'échantillon (noté N) et du nombre de classe (noté k) et de l'inertie totale de l'échantillon (notée T) nous avons choisi les quatre critères suivants :

C₁ : Pourcentage d'inertie expliquée

C'est le rapport (T-W)/T. Ce rapport est égal à 0 si l'échantillon constitue une seule classe et à 1 si chaque élément de l'échantillon forme sa propre classe.

$$C_1 = (T-W)/T$$

C₂ : Augmentation relative de l'inertie expliquée

Soient W_k l'inertie intra-classe de la partition associée au découpage en k classes et W_{k+1} l'inertie intra-classe de celle en (k+1) classes. L'augmentation relative de l'inertie expliquée est le rapport

$$C_2 = (T-W_{k+1})/(T-W_k)$$

C₃ : Critère de Fisher :

Le critère de Fisher est égal à :

$$C_3 = \frac{N-k}{k-1} \frac{(T-W_k)}{W_k}$$

qui dans le cadre de l'analyse de la variance suit une loi de Fisher à (k-1, N-k) degrés de liberté. En dehors de ce cadre on peut cependant remarquer que le premier terme (N-k)/(k-1) décroît quand k->N et le deuxième terme (T-W_k)/W_k croît quand k->N.

C₄ : Critère Hartigan

Hartigan propose un critère, utilisant le même principe que le critère de l'augmentation de l'inertie par rapport au critère du pourcentage d'inertie expliquée. Il s'écrit de la manière suivante :

$$C_4 = \frac{(k-1)}{k} \frac{W_{k-1}}{W_k}$$

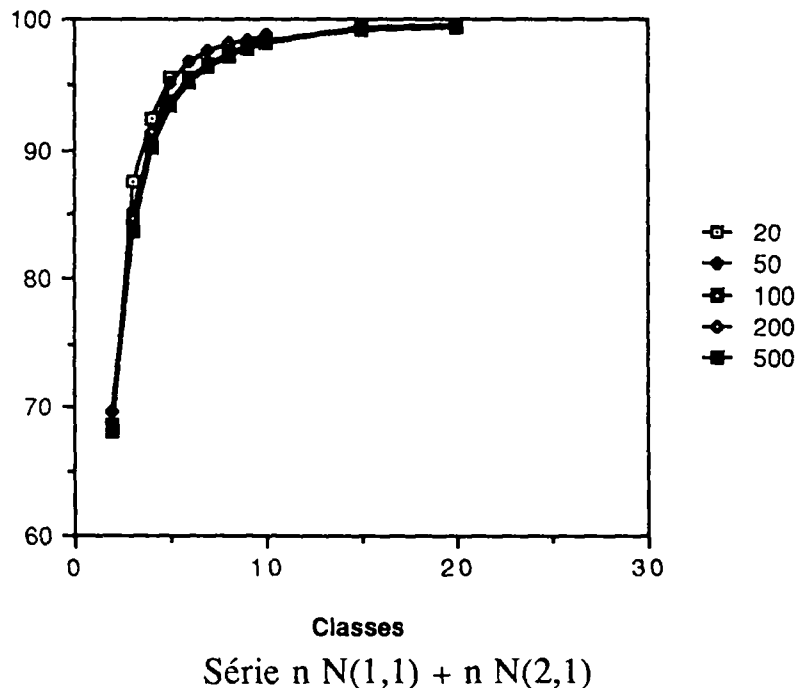
De plus pour chaque simulation nous allons comparer les résultats du découpage optimal par rapport à deux autres découpages, souvent utilisés en Analyse des Données, qui sont le découpage à intervalles égaux et le découpage à effectif égaux.

A partir de ces trois types de mélanges nous allons vérifier la variation du critère et la stabilité des bornes en fonction de l'effectif de l'échantillon. Dans cette étude l'effectif de l'échantillon est proportionnel au nombre de composants.

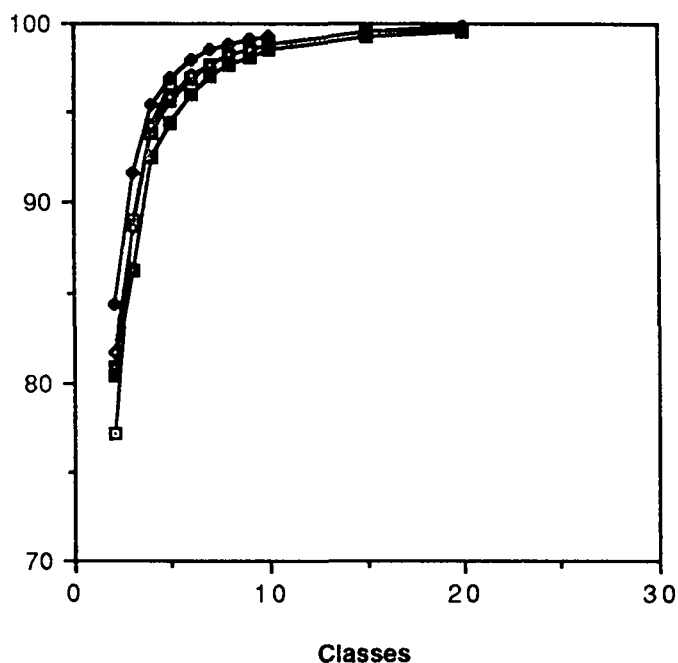
Les trois tableaux suivants donnent l'évolution du pourcentage d'inertie expliquée en fonction de l'effectif d'un composant (noté n) et le nombre de coupures demandé (noté k).

Le pourcentage d'inertie expliquée est assez stable en fonction de l'effectif de l'échantillon et la valeur de la borne de la coupure aussi si le nombre de classes correspond au nombre de composants du mélange.

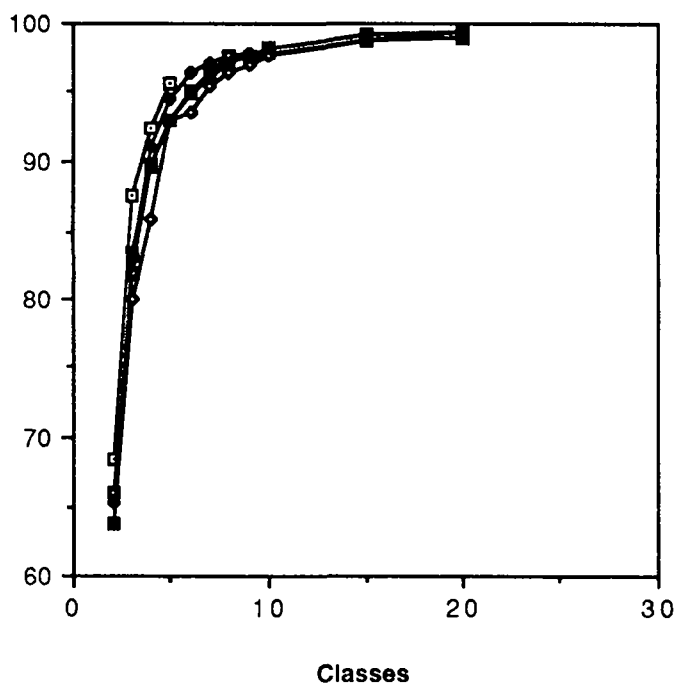
Les trois figures suivantes représentent les pourcentages d'inertie expliquée en fonction de l'effectif n de l'échantillon choisi et du nombre k de classes demandés. L'effectif pris est 20, 50, 100, 200 et 500.



Dans ces tableaux le nombre de classes varie entre 2 et 10 en pas de un, puis nous avons pris 15 et 20. Si l'effectif est faible le calcul du critère pour un nombre de classes élevé n'est pas effectué car il ne correspond plus à rien.



Série $n N (1, 0.5) + n N (2, 0.5)$

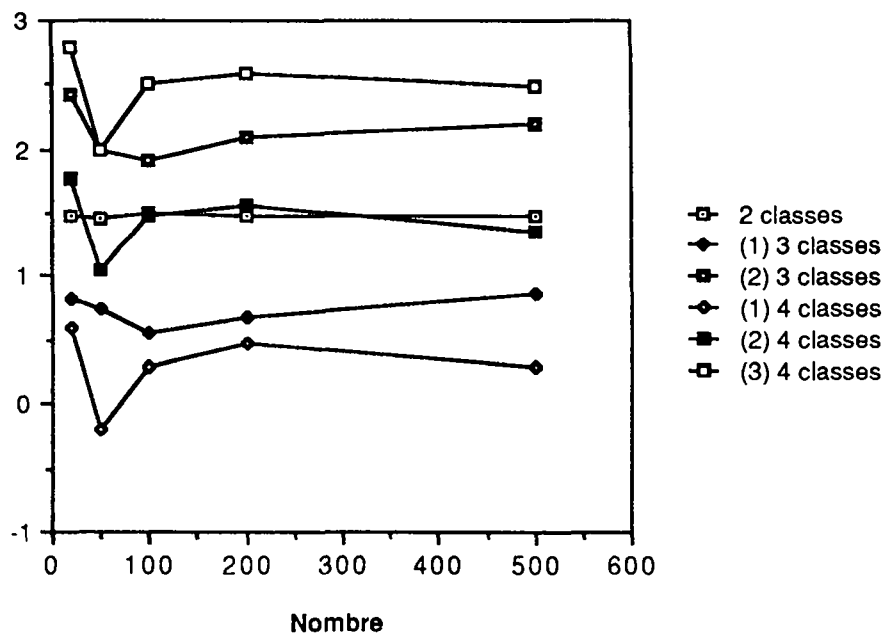


Série $n N (1,1) + n N (5,1)$

Les trois tableaux suivants donnent les bornes du découpage des trois séries. La valeur de la borne est la valeur inférieure de l'intervalle où doit s'effectuer le découpage ce qui explique que la valeur théorique est toujours plus grande que la valeur de la borne. Les bornes varient beaucoup dans la troisième série car les deux composants de cette série sont très séparés et les bornes se trouvent dans les zones presque vides.

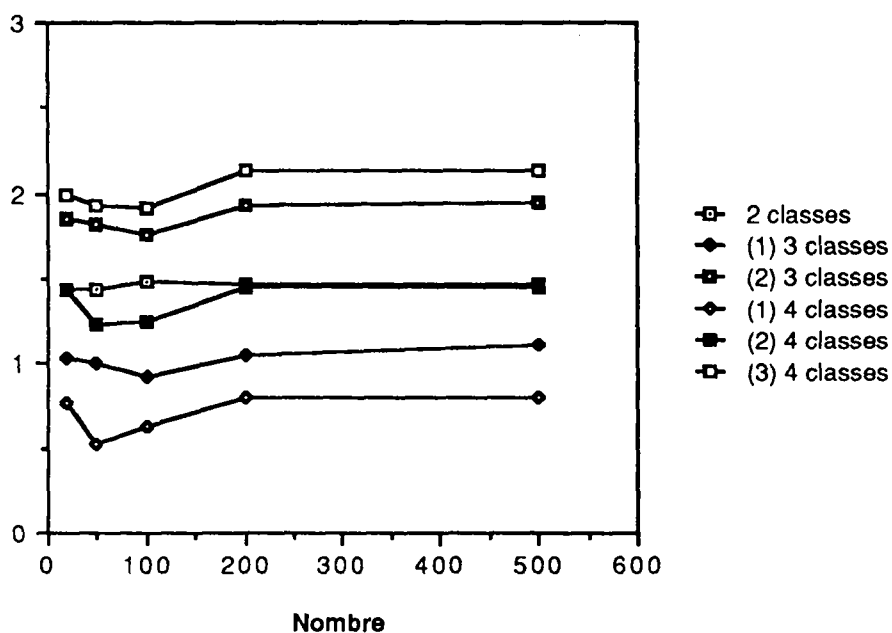
Lorsque le nombre de classes est égal à deux, la valeur théorique est de 1,5 dans les deux premiers tableaux et de 3 dans le dernier tableau.

	2 classes	3 classes		4 classes		
20	1.49	0.83	2.42	0.60	1.77	2.79
50	1.47	0.74	2.00	-0.20	1.06	2.00
100	1.50	0.56	1.92	0.29	1.49	2.50
200	1.49	0.69	2.09	0.48	1.56	2.60
500	1.48	0.86	2.21	0.30	1.36	2.49



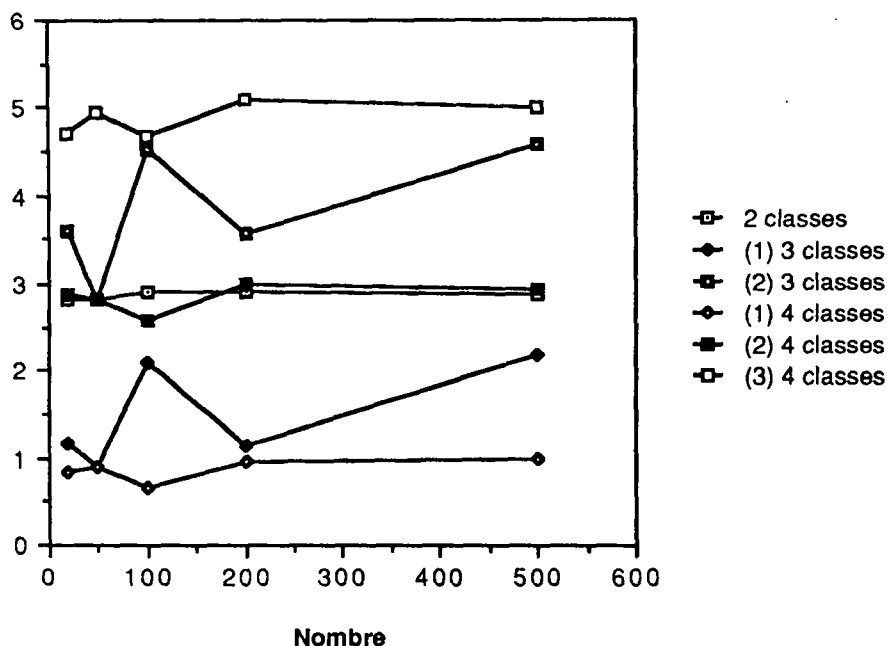
Bornes du découpage optimal pour la série $n N (1, 1) + n N (2, 1)$

	2 classes	3 classes		4 classes		
20	1.44	1.03	1.85	0.77	1.44	2.00
50	1.44	1.00	1.83	0.53	1.24	1.94
100	1.48	0.93	1.76	0.64	1.26	1.92
200	1.47	1.05	1.93	0.80	1.46	2.13
500	1.47	1.12	1.95	0.81	1.46	2.13



Bornes du découpage optimal pour la série $n N (1, 0.5) + n N (2, 0.5)$

	2 classes	3 classes		4 classes		
20	2.83	1.18	3.60	0.83	2.89	4.70
50	2.83	0.89	2.82	0.89	2.82	4.94
100	2.90	2.09	4.53	0.65	2.59	4.69
200	2.92	1.14	3.56	0.96	2.99	5.09
500	2.89	2.18	4.60	1.00	2.94	5.00



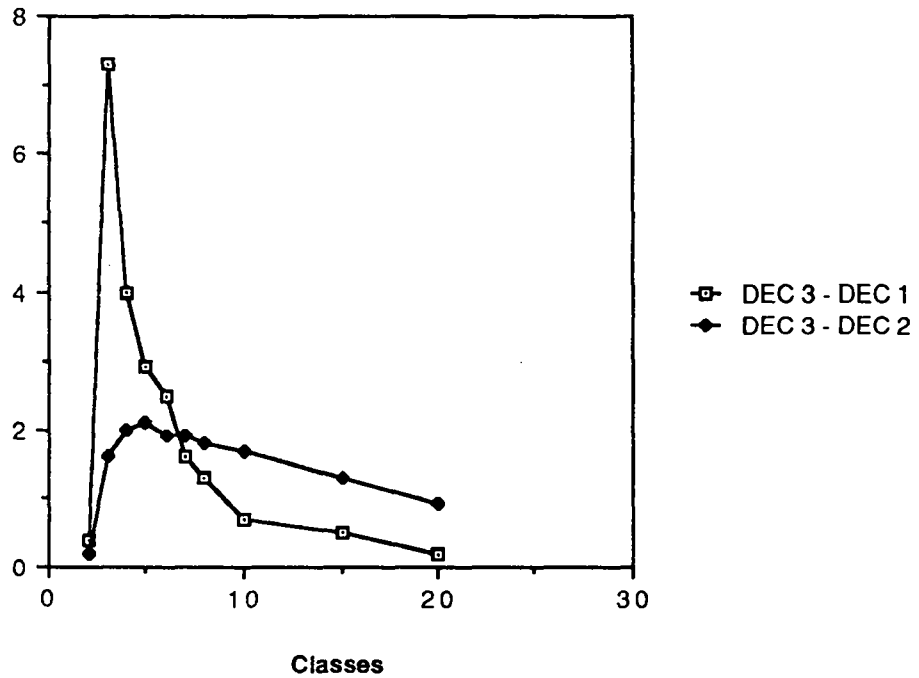
Bornes du découpage optimal pour la série $n N (1,1) + n N (5,1)$

En conclusion on obtient rapidement un pourcentage d'inertie expliquée assez élevé même lorsque les échantillons sont de faible effectif. Lorsque la valeur des coupures est connue (en deux classes) la valeur observée est proche de cette valeur théorique même avec un petit échantillon. On peut conclure que la méthode donne de bons découpages de variables quantitatives. Maintenant nous allons comparer ce découpage avec deux découpages très utilisés ; le découpage à intervalles égaux et le découpage à effectifs égaux.

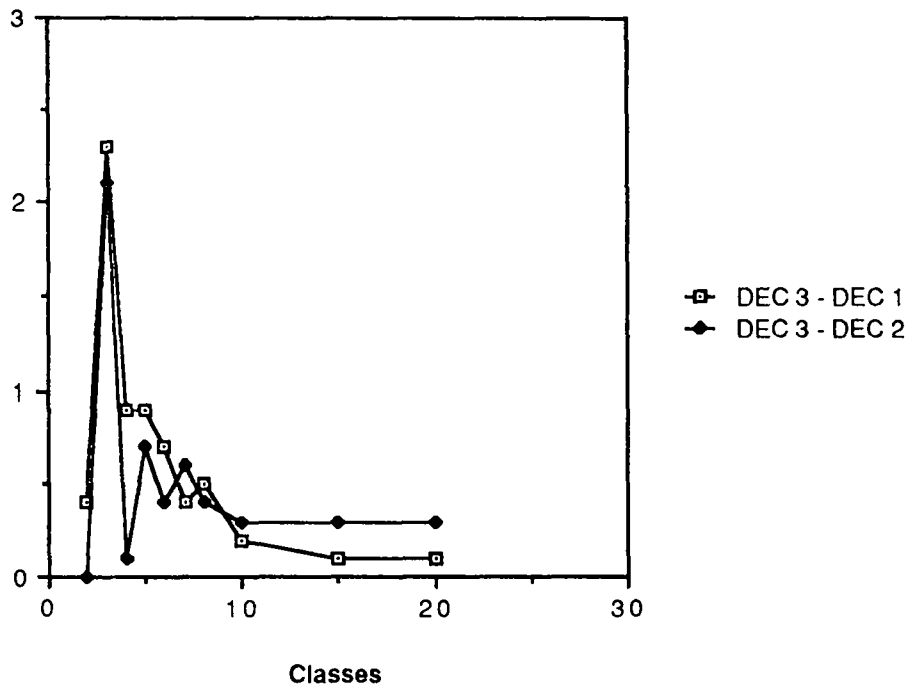
2.1.3. Comparaison de ce découpage avec deux autres découpages

Ce découpage peut être utilisé de la même manière que le découpage à effectifs égaux ou à intervalles égaux d'une variable quantitative afin de la rendre qualitative. En fonction du critère d'inertie nous pouvons mesurer le gain de ce découpage en fonction des deux découpages les plus usuels car ce découpage maximise le critère d'inertie. Ainsi, en supposant que le critère d'inertie est un bon critère de découpage, on pourra déterminer les cas où le découpage optimal doit être utilisé, sachant que ce découpage est bien plus coûteux que les deux autres. En reprenant les séries les distributions gaussiennes du paragraphe précédent nous pouvons établir les tableaux comparatifs suivants :

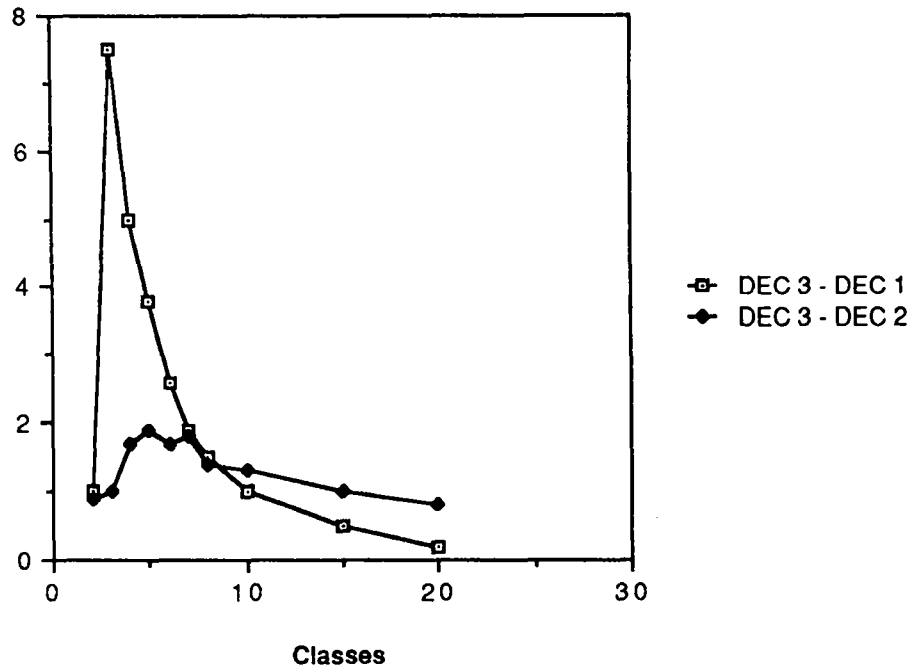
DEC1 : Découpage à Intervalles égaux
DEC2 : Découpage à Effectifs égaux
DEC3 : Découpage Optimal.



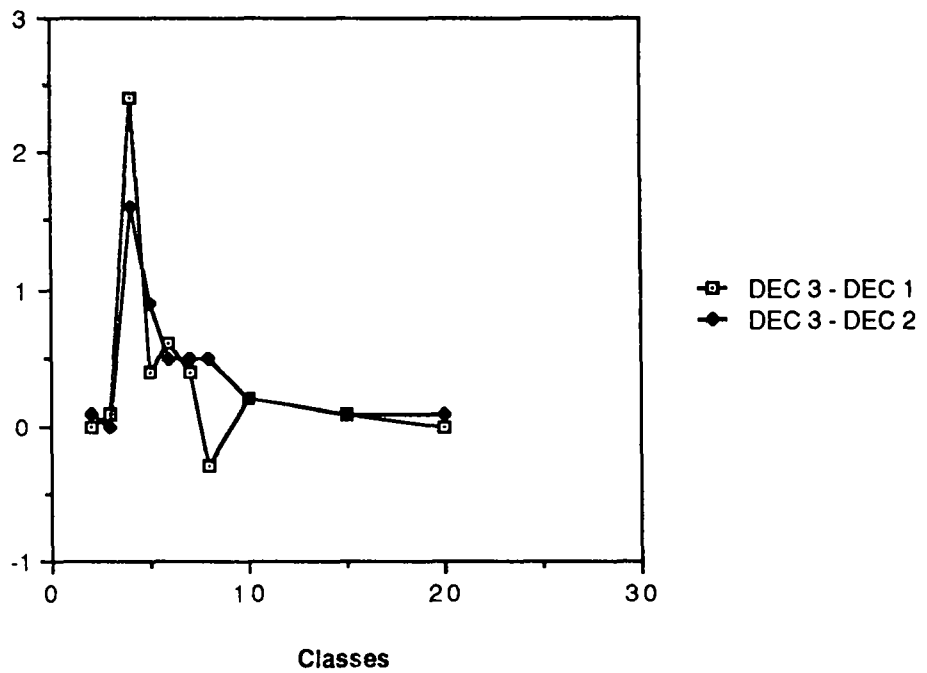
Série $M_{12} = 500 N(1, 1) + 500 N(2, 1)$



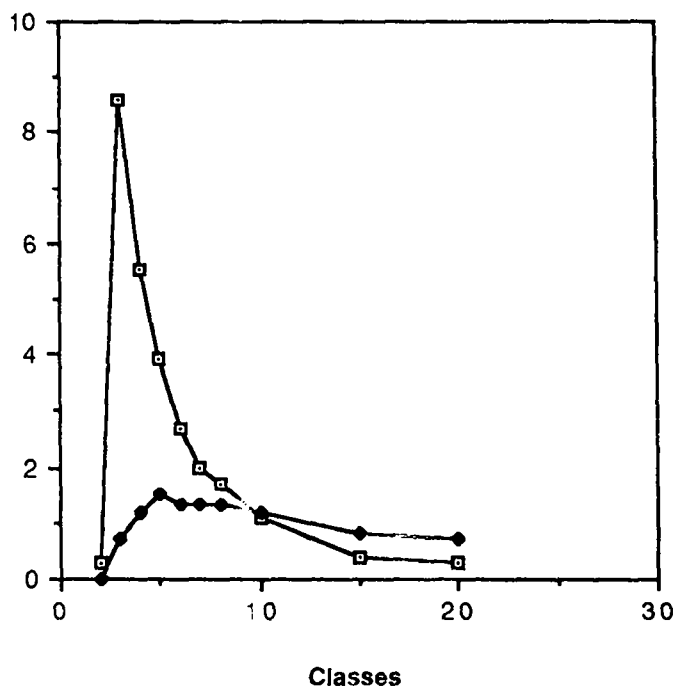
Série $M_{32} = 500 N(1, 1) + 500 N(5, 1)$



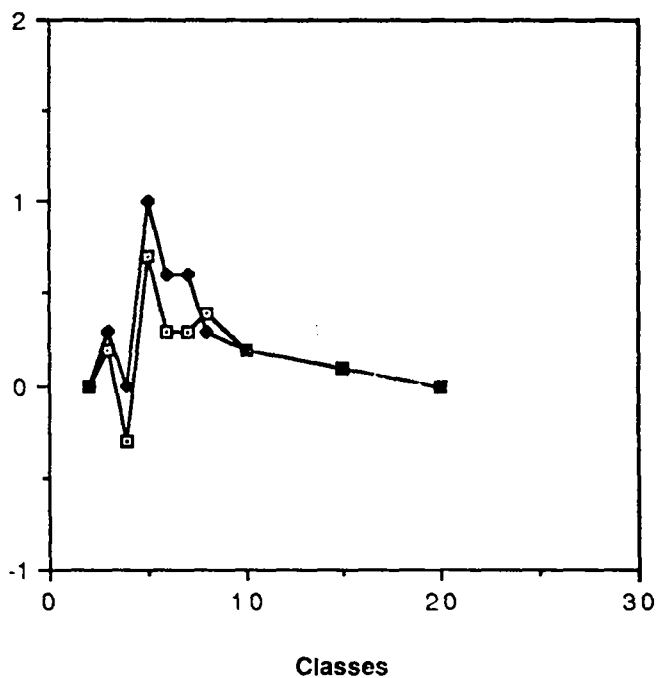
Série $M_{13} = 500 N (1, 1) + 500 N (2, 1) + 500 N (3, 1)$



Série $M_{33} = 500 N (1, 1) + 500 N (5, 1) + 500 N (9, 1)$



Série $M_{14} = 500 N (1, 1) + 500 N (2, 1) + 500 N (3, 1) + 500 N (1, 1)$

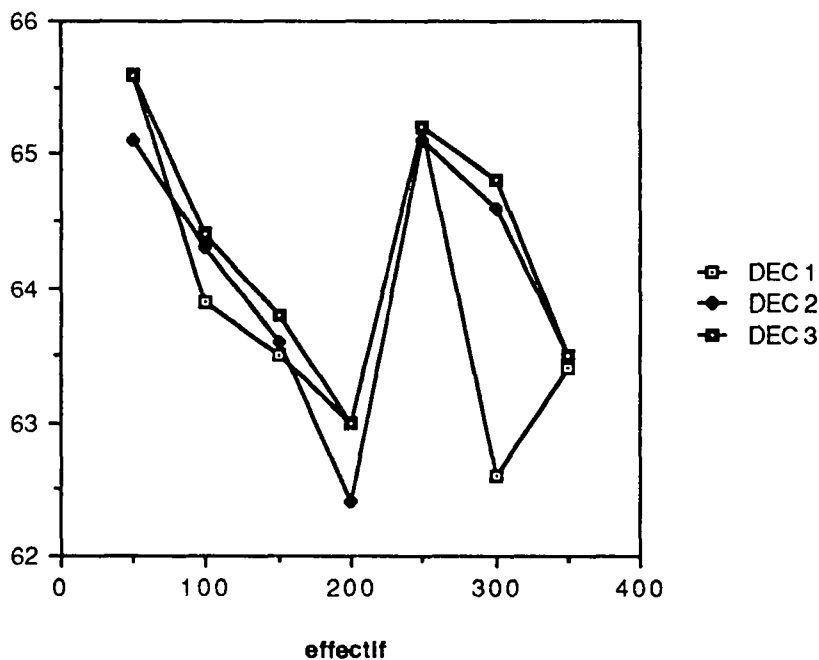


Série $M_{34} = 500 N (1, 1) + 500 N (5, 1) + 500 N (9, 1) + 500 N (3, 1)$

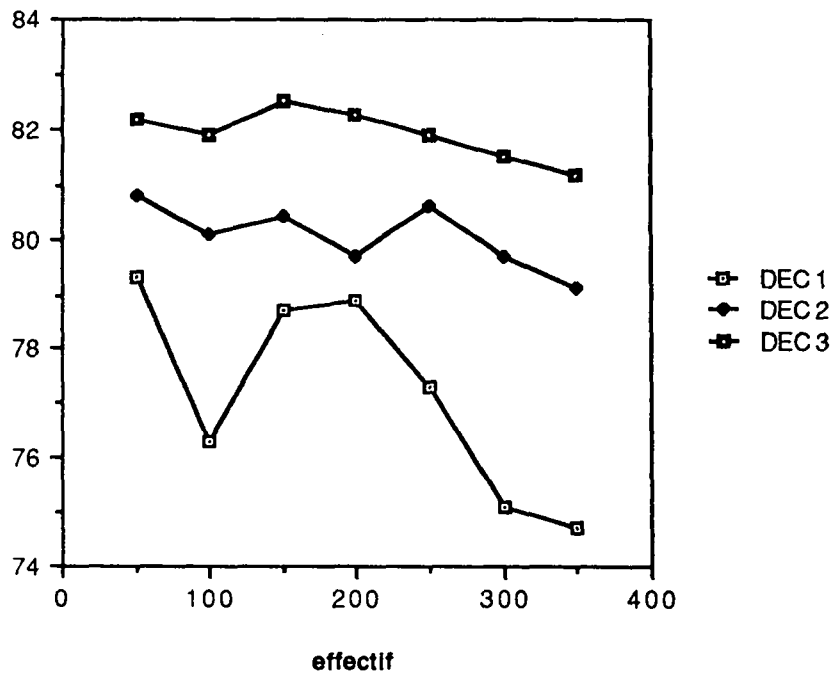
D'une manière générale on remarque que dans le cas d'une distribution gaussienne ayant des composants bien séparés le pourcentage d'inertie expliqué est très élevé, ce qui est normal et que l'écart entre les trois découpages diminue en fonction du nombre de classes.

Remarquons que le découpage optimal pour 2 classes ne donne pas une solution bien meilleure que celles obtenues par le découpage à intervalles égaux ou à effectifs égaux. Par contre l'écart se creuse entre 3 et 6 classes dans tous les cas que nous avons simulés. Mais à partir de 10 classes la différence est devenue faible. Dans les six tableaux précédents nous avons effectués des simulations avec des composants ayant les mêmes effectifs ce qui, a priori, favorise le découpage à effectifs égaux. Cependant nous obtenons des meilleurs résultats avec le découpage à intervalles égaux lorsque le nombre de classes dépasse 10 et ces résultats sont proches des résultats du découpage optimal. Le découpage optimal est intéressant à utiliser si le nombre de classes se situe entre 3 et 7, au delà le découpage à intervalle égaux est une bonne stratégie.

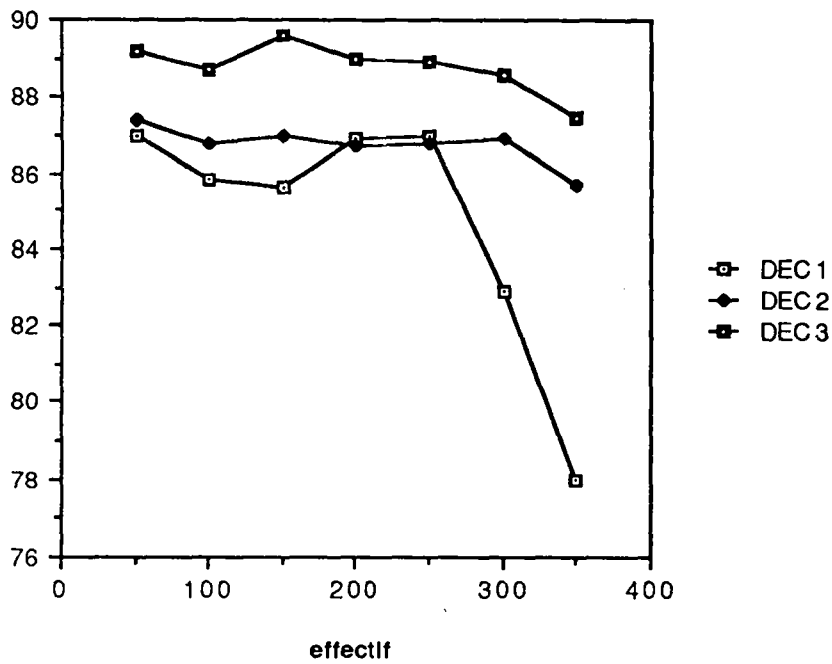
Nous allons maintenant faire varier les effectifs des composants du mélange de la manière suivante : $n N(1, 1) + (500-n) N(2, 1)$. L'axe horizontal représente la valeur n et l'axe vertical le pourcentage d'inertie expliquée.



découpage en 2 classes :



découpage en 3 classes :

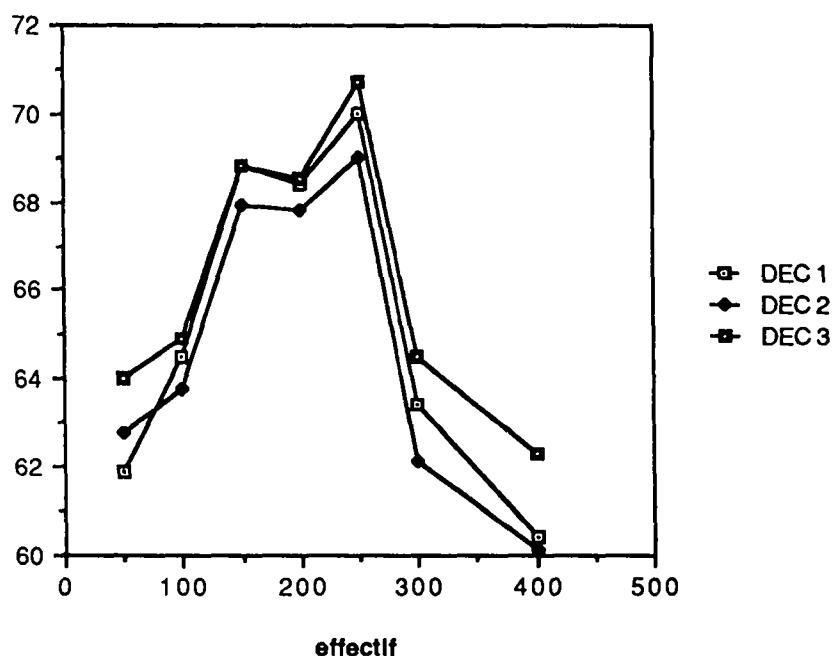


découpage 4 classes

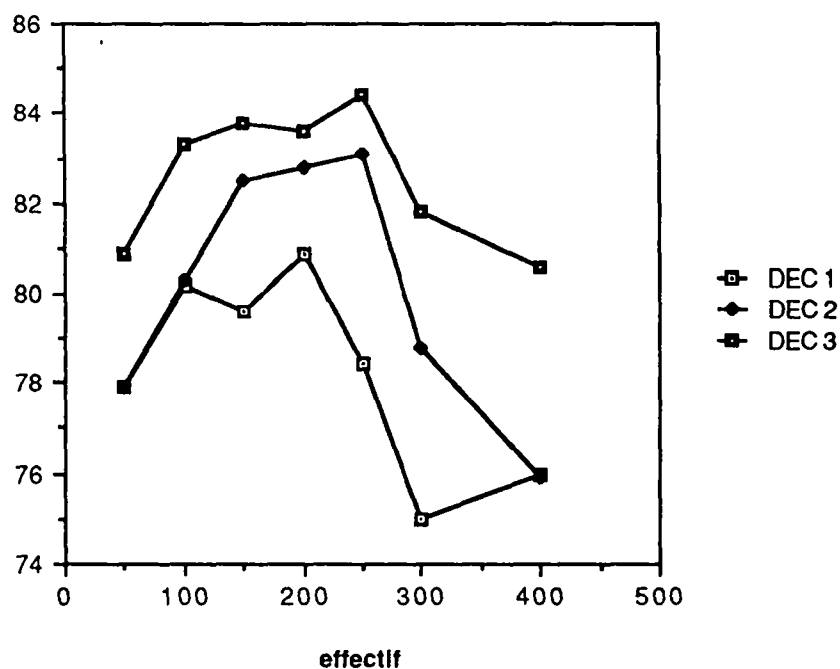
Si les deux composants sont assez proches nous remarquons que pour deux classes la variation de l'effectif des deux composants n'a pas d'influence sur le résultat. Dans ce cas les solutions proposées par les trois

méthodes de découpage sont très proche. Par contre cet écart devient sensible avec 3 à 7 classes.

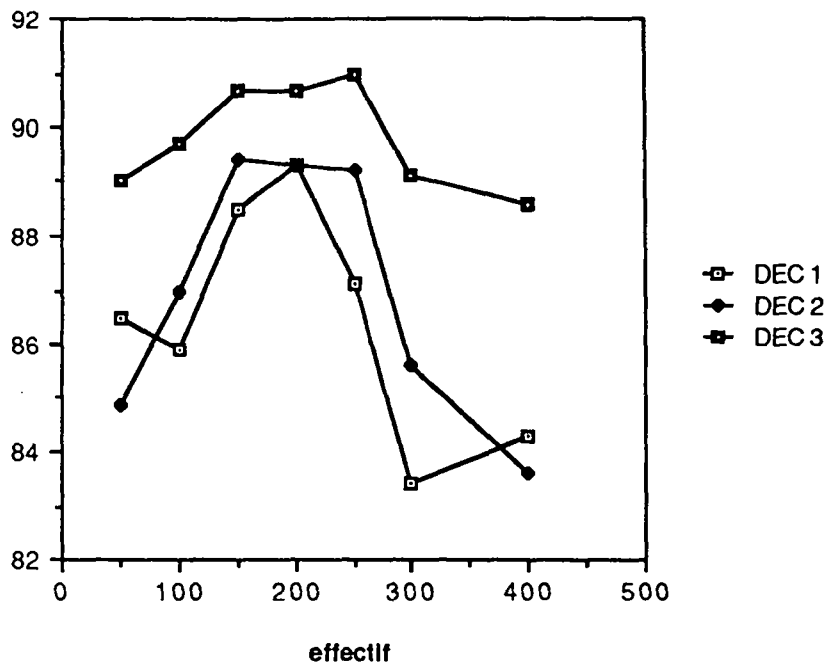
En prenant un mélange dont ses deux composants sont mieux séparés nous obtenons les résultats suivants :



Découpage en 2 classes de $n N(1, 1) + (400-n) N(1, 3)$



Découpage en 3 classes



Découpage en 4 classes

Si le nombre de classe est égal à deux les solutions proposées par les trois méthodes restent très proches. Par contre l'écart devient sensible entre 3 à 7 classes et cet écart est plus important que celui du mélange précédent ce qui est normal car la structure en classe est plus nette. En regardant la borne du découpage en deux classes nous remarquons que les trois méthodes obtiennent la valeur 2 lorsque la population est équilibrée (cette valeur est la valeur théorique). Avec une population non équilibrée la borne obtenue par le découpage à effectif égaux est toujours la plus extrême par rapport à la valeur 2.

2.2 Codage d'une variable quantitative sous contrainte d'une variable qualitative

2.2.1 Introduction

Le problème est de découper une variable quantitative en tenant compte d'une variable qualitative. Par exemple il s'agit de découper une variable de mesure en fonction d'un ensemble de classes a priori, ce découpage essayant d'établir le lien maximal entre la variable découpée et la partition formée par ces classes a priori. La mesure du lien entre cette variable

qualitative a priori et celle obtenue par le découpage peut être un χ^2 ou l'information de Shannon ou tout autre critère additif défini sur les intervalles du découpage.

2.2.2 Théorème

A chaque élément e_i de E est associé un couple (x_i, w_i) tel que

$$x_i \in X \text{ et } w_i \in \Omega \text{ où}$$

X est un ensemble totalement ordonné (par exemple \mathbb{R})

Ω est un ensemble fini (l'ensemble des formes a priori).

On note $Q_K(E)$ l'ensemble des partitions de E en K classes non vides vérifiant la propriété d'ordre $[P_1]$ sur l'ordre donné par X .

On définit un critère W , mesurant le lien entre un découpage de X et Ω , de la forme suivante :

$$W(P) = \sum_{i=1}^k w(B_i) \text{ où}$$

$P = (B_1, \dots, B_K)$ avec $P \in Q_K(E)$ et w une application de $P(E) \times \Omega$ dans \mathbb{R}^+ alors l'algorithme DECOPT, décrit dans le paragraphe 1.2.6.3, permet de trouver une partition Q de $Q_K(E)$ optimale pour ce critère W .

Démonstration :

Il faut vérifier que le principe de la programmation dynamique soit respecté. Le critère est additif par définition, il suffit que la valuation w d'une classe soit indépendante de la définition des autres classes de la partition Q . Comme par définition la valuation w est définie sur $P(E) \times \Omega$ le résultat de cette application sur un intervalle de X est indépendant des autres intervalles de X . Donc l'algorithme DECOPT permet de trouver une partition Q de $Q_K(E)$ optimale pour ce critère W .

2.2.3 Critère basé sur la quantité d'information

On associe aux deux ensembles une mesure de probabilité conjointe $P_{X \times \Omega}$ définie ainsi :

Si U est un intervalle de X et $a \in \Omega$ alors

$$P_{X \times \Omega}(U, a) = \frac{\text{Card}(\{e_i \in E / x_i \in U\} \cap \{e_i \in E / w_i = a\})}{\text{Card}(E)}$$

On définit les probabilités induites sur les deux ensembles X et Ω :

$$\forall U \text{ intervalle de } X \quad P_X(U) = \sum_{w \in \Omega} P_{X \times \Omega}(U, w)$$

$$\forall a \in \Omega \quad P_\Omega(a) = \int_X dP_{X \times \Omega}(U, a)$$

On appelle Information mutuelle au sens de Shannon la fonction suivante :

$$I(P) = \sum_{i=1}^k \sum_{a \in \Omega} P_{X \times \Omega}(p_i, a) \log \left[\frac{P_{X \times \Omega}(p_i, a)}{P_X(p_i) P_\Omega(a)} \right]$$

nous recherchons une partition optimale Q telle que

$$I(Q) = \text{Max}_{P \in Q_K(E)} I(P)$$

Le critère I s'écrit aussi :

$$I(P) = \sum_{i=1}^K w(P_i) \text{ où } P = (P_1, \dots, P_K) \in Q_K(E)$$

Si on pose

$$w(P_i) = \sum_{X \times \Omega} P_{X \times \Omega}(P_i, a) \log \left[\frac{P_{X \times \Omega}(P_i, a)}{P_X(P_i) P_\Omega(a)} \right]$$

Par définition, $P_{X \times \Omega}$ ne dépend que de l'intervalle P_i et de Ω , il en est de même de P_X et P_Ω alors w est une application de $P(E) \times \Omega$ dans \mathbb{R}^+ .

Donc nous pouvons utiliser l'algorithme DECOPT pour résoudre ce problème.

2.2.4 Critère basé sur le khi2

Le problème est de trouver une partition Q de $Q_k(E)$ qui maximise le χ^2 de contingence entre la probabilité conjointe $P_{X \times \Omega}$, et le produit des probabilités marginales des deux ensembles X et Ω . Le produit des deux probabilités marginales sera considéré comme la probabilité théorique, la probabilité observée sera la probabilité conjointe.

Nous appellerons intensité de dépendance entre la probabilité conjointe $P_{X \times \Omega}$ et la probabilité produit $P_X \cdot P_\Omega$ la valeur Φ^2 tel que

$$\Phi^2(Q) = \sum_{i=1}^k \sum_{a \in \Omega} \frac{[P_{X \times \Omega}(P_i, a) - P_X(P_i) \cdot P_\Omega(a)]^2}{P_X(P_i) \cdot P_\Omega(a)}$$

Notre algorithme permet de trouver Q de $Q_K(E)$ qui maximise l'intensité de dépendance entre un découpage en K intervalles disjoints et non vides de X et l'ensemble Ω des formes a priori.

$$\Phi^2(Q) = \text{Max}_{P \in Q_K(E)} \Phi^2(P)$$

Comme dans le paragraphe précédent il suffit de vérifier que la valeur

$$W(P_i) = \sum_{a \in \Omega} \frac{[(P_{X \times \Omega}(P_i, a) - P_X(P_i) \cdot P_\Omega(a))]^2}{P_X(P_i) P_\Omega(a)}$$

ne dépend que de P_i et Ω ce qui se vérifie facilement.

2.2.5 Détermination du nombre de classes

Le critère d'inertie le critère du χ^2 est croissant en fonction du nombre de classes il est impossible d'utiliser ce critère pour la détermination du nombre de classes. De plus le découpage optimal permet, par construction, d'obtenir des valeurs élevées aussi la transformation du critère du χ^2 en probabilité est inutilisable. Nous proposons plusieurs coefficients calculés à partir de Φ^2 de Pearson et assez indépendants du nombre de classes.

Le Φ^2 de Pearson est équivalent au χ^2 calculé sur le tableau des probabilités plutôt que sur le tableau de contingence. La valeur de Φ^2 est égale à

$$\Phi^2 = \frac{\chi^2}{n} \text{ où } n \text{ est l'effectif de la population.}$$

Le Φ^2 est aussi appelé intensité de dépendance.

Si on note L_m le plus petit nombre de modalités de l'une ou de l'autre variable alors cette valeur est comprise entre 0 et $L_m - 1$. Aussi Cramer propose son coefficient V

$$V = \sqrt{\frac{\Phi^2}{L_{m-1}}}$$

qui est en mesure d'association normée et qui est égale à un en cas d'association parfaite entre les deux variables. Il est recommandé pour cette propriété par Goodman et Kruskal (1954).

D'autres mesures d'association normées sont proposées ; le C de Pearson :

$$C = \sqrt{\frac{\frac{\chi^2}{2}}{\chi^2 + n}}$$

mais il ne peut pas servir de base de comparaison entre plusieurs tableaux de contingence de dimension différentes, aussi nous proposons le coefficient de contingence normé par

$$C_m = \frac{C}{\text{Max}} \text{ avec } \text{Max} = \sqrt{\frac{L_{m-1}}{L_m}}$$

Nous avons aussi le coefficient T de Tschuprow :

$$T = \sqrt{\frac{\Phi^2}{\text{lib}}}$$
 où lib est le nombre de degrés de liberté du tableau de contingence.

Cependant ce coefficient ne peut atteindre son maximum si et seulement si les deux variables ont le même nombre de modalités. Ce coefficient est un indice angulaire de dépendance entre deux variables qualitatives. G. Saporta (1975) montre que ce coefficient est le cosinus de l'angle entre les deux projecteurs centrés associés aux deux variables qualitatives. En conclusion nous proposons pour l'aide au choix du nombre de classes les coefficients de Cramer ou de Tschuprow. Dans le chapitre suivant nous allons, par simulation, montrer l'intérêt de ces coefficients.

2.2.6 Simulation

Pour ces simulations nous allons reprendre les 9 mélanges proposés en 2.1.2. La variable qualitative représente le numéro de classe a priori de

chaque élément de l'échantillon. Maintenant chaque élément est caractérisé par sa valeur et son numéro de classe a priori.

Nous obtenons les résultats suivants :

Valeur du critère χ^2 en fonction du nombre de classes et de la série.

Série ^K	2	3	4	5	6	7	8	9	10	15
M ₂₂	485	549	559	567	571	574	577	580	582	591
M ₁₂	132	180	190	198	204	209	215	219	222	222
M ₃₂	944	954	956	957	957	958	958	958	958	958
M ₂₃	931	1532	1600	1646	1670	1692	1702	1707	1713	1741
M ₁₃	492	598	657	675	690	699	707	715	723	755
M ₃₃	1416	2733	2773	2789	2794	2798	2800	2805	2806	2811
M ₂₄	1474	2404	3051	3177	3256	3315	3350	3370	3380	3424
M ₁₄	920	1263	1391	1440	1481	1506	1524	1534	1546	1592
M ₃₄	1897	3754	5444	5498	5532	5559	5573	5579	5585	5605

Le mélange M₁, le plus homogène, est représenté, par les séries M₁₂, M₁₃ et M₁₄. Le mélange M₂, le mélange intermédiaire, est représenté par les séries M₂₂, M₂₃ et M₂₄. Le mélange M₃, le mélange hétérogène est représenté par les séries M₃₂, M₃₃ et M₃₄. La valeur du khi2 conserve bien l'ordre de confusion des trois types de mélange, avec un khi2 faible le mélange est très hétérogène. Avec le nombre de classes égal à deux le khi2 associé au mélange M₁₂, ce facteur est égal à 4 pour 3 composants (M₁₃ et M₃₃) et n'est plus égal à deux pour 4 composants (M₁₄ et M₃₄).

En regardant l'évolution du critère des mélanges à 3 composants on remarque qu'entre deux et trois classes le khi2 associé au mélange M₁₃ croit de 20 %, celui du mélange M₂₃ de 65 % et celui de M₃₃ de 90 %.

L'évolution du critère associé aux mélanges ayant 4 composants est identique ainsi plus le mélange est hétérogène plus la variation du khi2 est importante. Ainsi le tracé de la courbe du taux d'accroissement du critère du khi2 en fonction du nombre de coupures nous permet de prendre une bonne décision. Cependant l'utilisation du critère de Tschuprow permet de déterminer de manière automatique le bon nombre de coupures. Le critère s'est révélé efficace pour les mélanges M₂ et M₃.

Il a été efficace pour le mélange M₁ uniquement avec les séries M₁₃ et M₁₄. Le test V de Cramer ne donne qu'un optimum local pour la solution

théorique car souvent la solution à 15 coupures est la solution retenue pour ce critère.

2.2.7 Comparaison des codages sur les données d'Iris

Ce tableau de données est composé de trois espèces d'Iris ; l'espèce SETOSA est représentée par 50 éléments, l'espèce VERSICOLOR par 50 éléments ainsi que l'espèce VIRGINIA. Ces 150 éléments sont caractérisés par quatre variables continues :

la longueur et la largeur du sépale, la longueur et la largeur du pétale.

Etant donné que les effectifs des trois classes a priori d'Iris sont égaux nous avons recommencé le codage en doublant l'espèce VERSICOLOR et en triplant l'espèce VIRGINIA. Ainsi la population 1 sera composée de 50 SETOSA, 50 VERSICOLOR et 50 VIRGINIA, la population 2 sera composée de 50 SETOSA, 100 VERSICOLOR et 150 VIRGINIA.

En utilisant le découpage optimal avec le critère du χ^2 nous obtenons sur ces deux populations les mêmes coupures. Le critère de Tschuprow nous propose un découpage en deux classes des variables associées au sépale et un découpage en trois classes des variables associées au pétale.

Valeurs du critère de Tschuprow en fonction du nombre de coupures.

Nb coupures	Population 1				Population 2			
	LoSe	laSe	LoPe	laPe	LoSe	laSe	LoPe	laPe
1	69,5	51,9	84,1	66,5	47,9	84,1	84,1	84,1
2	65,7	49,2	93,5	94,3	63,1	44,4	94,0	93,7
3	61,1	46,1	85,7	85,7	58,3	42,4	85,8	85,3
4	57,6	43,8	79,9	79,8	55,5	40,5	79,9	79,6
5	54,9	41,7	75,6	75,5	53,4	38,7	75,6	75,3
6	52,7	40,0	72,3	72,2	51,5	37,2	72,3	71,9
7	50,8	38,6	69,5	69,4	49,7	36,1	69,9	69,3

Nous pouvons remarquer que les variables associées au pétale sont beaucoup plus discriminantes que celles associées au sépale. Bien que nous avons réalisé une population 2 en augmentant de manière importante les deux classes d'Iris les plus proches nous pouvons remarquer que le critère de Tschuprow est assez voisin entre ces deux populations. Comme le critère de Tschuprow nous propose un découpage en deux classes des deux variables associées au sépale nous avons regroupé les VERSICOLOR et les VIRGINIA pour le calcul des mal classés.

Variabes	Bornes des coupures	Mal-classés
LaSe	5,4	12
laSe	3,3	25
LoPe	1,9 et 4,7	7
loPe	0,6 et 1,6	6

Afin de mesurer l'apport de ce découpage nous allons comparer ces résultats avec ceux obtenus par le découpage à effectifs égaux (DEC1), le découpage à intervalles égaux (DEC2) et le découpage minimisant la variance (DEC3). Ces découpages ne donnent pas les mêmes coupures sur les deux populations choisies.

Variables	Population 2			Population 1		
	DEC 1	DEC 2	DEC3	DEC 1	DEC2	DEC3
LoSe	6,1	6,1	6,0	5,8	6,1	5,9
laSe	3,0	3,2	3,0	3,0	3,2	3,1
LoPe	4,35 et 5,2	2,97 et 4,94	1,9 et 5,0	2,45 et 4,9	2,97 et 4,94	1,9 et 4,9
loPe	1,3 et 1,9	0,9 et 1,7	0,6 et 1,7	0,8 et 1,6	0,9 et 1,7	0,6 et 1,6

Ici le découpage en minimisant la variance est le plus proche du découpage optimisant le critère du χ^2 .

En conclusion on peut affirmer que si l'utilisateur a un objectif à atteindre dans son découpage et que cet objectif peut être caractérisé par une partition a priori le découpage optimisant le critère χ^2 donne une bonne solution et permet, en plus, de choisir le bon nombre de coupures.

De plus l'analyse factorielle des correspondances appliquées sur les quatre variables codées par le découpage optimal donne un premier plan factoriel ayant 84,18 % d'inertie expliquée. Cette même analyse factorielle appliquée sur ces quatre variables codées par un découpage à effectifs égaux donne un premier plan factoriel ayant 68,67 % d'inertie expliquée.

2.2.8 Critère de prédiction

Avec le critère du khi2 ou de l'information nous mesurons une relation fonctionnelle et symétrique entre la variable qualitative a priori et la variable continue découpée. La mesure de prédiction doit être une mesure causale entre deux variables. Elle est donc, par nature asymétrique. Par exemple si "X entraîne Y" est une relation causale donc cette relation n'est pas équivalente à "Y entraîne X". Par contre une relation fonctionnelle mesurée par le critère du khi2 est une relation du type "les variables X et Y sont associées".

Plusieurs mesures ont été proposées en 1954 par Godman et Kruskal. Elles sont basées sur la réduction d'erreur de classement dans la variable à prédire à partir de la connaissance de la variable prédictive. Si X est la variable prédictive et Y la variable à prédire alors le critère W de prédiction de X ou Y est égal à

$$W(Y/X) = \frac{Q(Y) - Q(Y/X)}{Q(Y)}$$

$Q(Y)$ est la probabilité d'erreur de classement dans Y sans connaissance de X et $Q(Y/X)$ est la probabilité d'erreur de classement dans Y en connaissant X.

Ces probabilités d'erreur dépendent de la stratégie de classement. Les deux variables X et Y sont des variables qualitatives, ainsi elles sont représentées sur l'ensemble Ω par deux partitions.

$$U = (U_1, \dots, U_k) \text{ et } P = (P_1, \dots, P_m)$$

Un individu w de Ω a une probabilité $p(P_i)$ d'appartenir à la classe P_i de P et $p(U_j)$ d'appartenir à la classe U_j de U. La première stratégie proposée pour Goodman et Kruskal est, en cas de non connaissance de X, d'affecter tous les individus dans la classe la plus grande. Dans ce cas

$$Q(Y) = 1 - p(P_m)$$

où P_m est un des éléments de $\{P_i \in P / \text{card}(P_i) \geq \text{card}(P_j) \text{ pour } j \neq i\}$

En connaissance de X des individus de la classe U_j sont maintenant mis dans la classe P_{mj} qui est la plus fréquente dans U_j .

Dans ce cas :

$$Q(Y/X) = \sum_{j=1}^K [1 - p(P_{mj} \cap U_j)]$$

où P_{mj} est un des éléments de

$$\{P_i \in P / \text{card}(P_i \cap U_j) \geq \text{card}(P_r \cap U_j) \text{ pour } r \neq i\}$$

La deuxième stratégie proposée par Goodman et Kruskal est une ventilation aléatoire des individus en fonction des probabilités des classes de la partition P associée à la variable Y. Dans ce cas la probabilité d'erreur de ce classement aléatoire est égale à

$$Q(Y) = \sum_{i=1}^m p(P_i) \times (1 - p(P_i))$$

Avec la connaissance de X est en utilisant la même stratégie nous avons

$$Q(Y/X) = \sum_{j=1}^k \sum_{i=1}^m \frac{p(P_i \cap U_j)}{p(U_j)} \times \left(1 - \frac{p(P_i \cap U_j)}{p(U_j)} \right)$$

Si la variable à prédire est une variable qualitative et la variable prédictive est une variable quantitative à discrétiser nous avons choisi cette stratégie. Dans ce cas d'une ventilation aléatoire des points en fonction des probabilités des classes a priori, une classe a de Ω à une probabilité d'erreur égale à

$$P_{\Omega}(a) (1 - P_{\Omega}(a)).$$

Alors la probabilité d'erreur du classement aléatoire de cette variable a priori est égale à

$$Q = \sum_{a \in \Omega} P_{\Omega}(a) (1 - P_{\Omega}(a))$$

Connaissant l'appartenance des points à un intervalle U de X la probabilité d'erreur d'un classement aléatoire est maintenant égale à

$$\sum_{a \in \Omega} \frac{P_{X \times \Omega}(U, a)}{P_X(U)} \cdot \left(1 - \frac{P_{X \times \Omega}(U, a)}{P_X(U)} \right)$$

Donc la probabilité d'erreur d'un classement aléatoire en connaissance d'un découpage en intervalles de X est égale à

$$Q(P) = \sum_{i=1}^K \sum_{a \in \Omega} \frac{P_{X \times \Omega}(P_i, a)}{P_X(P_i)} \cdot \left(1 - \frac{P_{X \times \Omega}(P_i, a)}{P_X(P_i)} \right)$$

avec $P = (P_1, \dots, P_K)$

Le critère de prédiction W_1 doit mesurer le gain entre la non connaissance du découpage de X et la connaissance du découpage de X, ce critère W s'écrit alors

$$W_1(P) = \frac{Q - Q(P)}{Q}$$

c'est un critère défini par Goodman et Kruskal (1954). Une bonne prédiction, à partir du découpage de X, de la variable des classes a priori est évaluée par une valeur élevée du critère W. Dans ce cas il est équivalent à minimiser le critère $Q(P)$.

Notre algorithme permet de trouver une partition P de $Q_k(E)$ qui minimise la probabilité d'erreur de classement, connaissant le découpage en K intervalles disjoints et non vides de X , des formes a priori de Ω . Pour cela il suffit de vérifier que la valeur $Q(P)$ ne dépend que de P_i et Ω ce qui est évident.

Ainsi notre méthode donne le découpage de l'ensemble E en K intervalles disjoints et non vides dont la connaissance de l'appartenance à l'un de ses intervalles donne la meilleure prédiction de la variable qualitative associée aux classes a priori.

La prédiction réciproque est plus délicate car la probabilité d'erreur d'un classement aléatoire dépend du découpage de la variable. Dans la cas il faut prédire l'appartenance à une classe a priori connaissant l'appartenance à l'intervalle U . Ainsi la probabilité d'erreur d'un classement aléatoire du découpage $P = (P_1, \dots, P_k)$ est égale à

$$T(P) = \sum_{i=1}^k P_X(P_i) (1 - P_X(P_i))$$

Donc la probabilité d'erreur d'un classement aléatoire en connaissance de la variable des classes a priori est égale à

$$T(P/\Omega) = \sum_{a \in \Omega} \sum_{i=1}^k \left(\frac{P_{X \times \Omega}(P_i, a)}{P_{\Omega}(a)} \right) \left(1 - \frac{P_{X \times \Omega}(P_i, a)}{P_{\Omega}(a)} \right)$$

Le critère de prédiction W_2 mesurant le gain entre $T(P)$ et $T(P/\Omega)$, s'écrit ainsi

$$W_2(P) = \frac{T(P) - T(P/\Omega)}{T(P)}$$

A partir de ces deux critères Goodman et Kruskal proposent la définition d'un critère de prédiction symétrique qui est la moyenne entre ces deux critères W_1 et W_2 . Ce critère donne un coefficient d'association entre le découpage P et la variable associé aux classes a priori. Pour obtenir un coefficient proche du ϕ^2 il faut mieux prendre le coefficient suivant

$$C(P) = \left(\frac{Q - Q(P)}{Q} \right)^2 + \left(\frac{T(P) - T(P/\Omega)}{T(P)} \right)^2$$

ce coefficient peut être normalisé comme ϕ^2 .

2.2.9 Généralisation à plusieurs variables

A chaque élément e_i de E on associe maintenant un $(p+1)$ uple tel que

$$e_i \in E(x_i^1, \dots, x_i^p, w_i) \in R^p \times \Omega$$

ou x_i^j est la valeur que prend l'élément e_i pour la variable quantitative j .

Le $(p+1)$ uple associé à chaque élément est décomposé en p couples

(x_i^j, w_i) . A chaque couple (x_i^j, w_i) nous sommes dans le cas précédent, c'est

à dire la recherche pour la variable j de la meilleure liaison stochastique entre le découpage en k_j intervalles disjoints de cette variable et l'ensemble Ω . Nous avons maintenant p découpages en k_1, \dots, k_p intervalles disjoints. Chaque découpage est optimal au sens de la liaison de dépendance entre la variable j et l'ensemble Ω . Sur le tableau de contingence construit à partir du croisement des p variables (U^1, \dots, U^p) et la variable qualitative définie par Ω , l'intensité de dépendance est optimale au sens suivant :

Soit $P_k(E, j)$ l'ensemble des partitions en k classes de E respectant la propriété d'ordre sur la variable j alors si (V^1, \dots, V^p) est un découpage quelconque (avec $V^j \in P_{k_j}(E, j)$) on a

$$\phi^2(V^1, \dots, V^p) < \phi^2(U^1, \dots, U^p) \text{ car le critère du } \chi^2 \text{ est additif.}$$

Aussi dans l'hypothèse d'indépendance nous avons la meilleure liaison stochastique entre les p découpages en k_j intervalles et l'ensemble Ω et, dans

ce cas, l'inertie du nuage suit un χ^2 à $\left(\sum_{i=1}^p (k_i - 1)(q-1) \right)$ degrés de liberté.

Dans ce cas l'inertie du nuage de points est maximale au sens de χ^2 .

2.3 Découpage d'une variable quantitative en fonction d'un ensemble de variables

2.3.1 Introduction

Le problème est de découper une variable quantitative en intervalles homogènes, cette homogénéité est définie par un ensemble de variables. Par exemple il s'agit de rechercher en fonction de la profondeur des zones homogènes, ces zones étant décrites par un ensemble de variables exogènes. Dans ce cas la variable à découper sert à définir sur les éléments à classer un ordre total. La mesure d'homogénéité d'un intervalle peut être l'inertie si des variables quantitatives ou le χ^2 si les variables sont qualitatives.

2.3.2 Théorème

A chaque élément e_i de E est associé une valeur $(x_i^1, \dots, x_i^p, y_i)$ tel que

$$x_i^j \in X_j \text{ et } y_i \in Y$$

où Y est élément totalement ordonné, et X_1, \dots, X_j un ensemble de variables.

On note $Q_k(E)$ l'ensemble des partitions des E sur k classes non vides vérifiant la propriété d'ordre $[P_1]$ sur Y .

On définit un critère W mesurant l'homogénéité du découpage de Y en fonction de (X_1, \dots, X_p) par

$$W(P) = \sum_{i=1}^K w(B_i)$$

où $P = (B_1, \dots, B_k)$ avec $P \in Q_k(E)$ et

w une application de $P(E)$ dans \mathbb{R}^+ alors l'algorithme que nous proposons permet de trouver une partition Q de $Q_k(E)$ optimale pour ce critère W .

Démonstration :

Il faut vérifier que le principe de la programmation dynamique [P₂] est respecté. Le critère est additif, par définition, il suffit que la valuation w d'une classe est indépendante de la définition des autres classes. Ceci est vérifié si w est une application de $P(E)$ sur \mathbb{R}^+ .

Si les variables X_1, \dots, X_p sont quantitatives nous pouvons prendre le critère de l'inertie intra classes. Soit g_i le centre de gravité de la classe B_i alors la variance de la classe s'écrit

$$W(B_i) = \sum_{e \in B_i} \sum_{j=1}^p (y_j^{(e)} - g_j^i)^2$$

cette valeur est bien indépendante des autres classes.

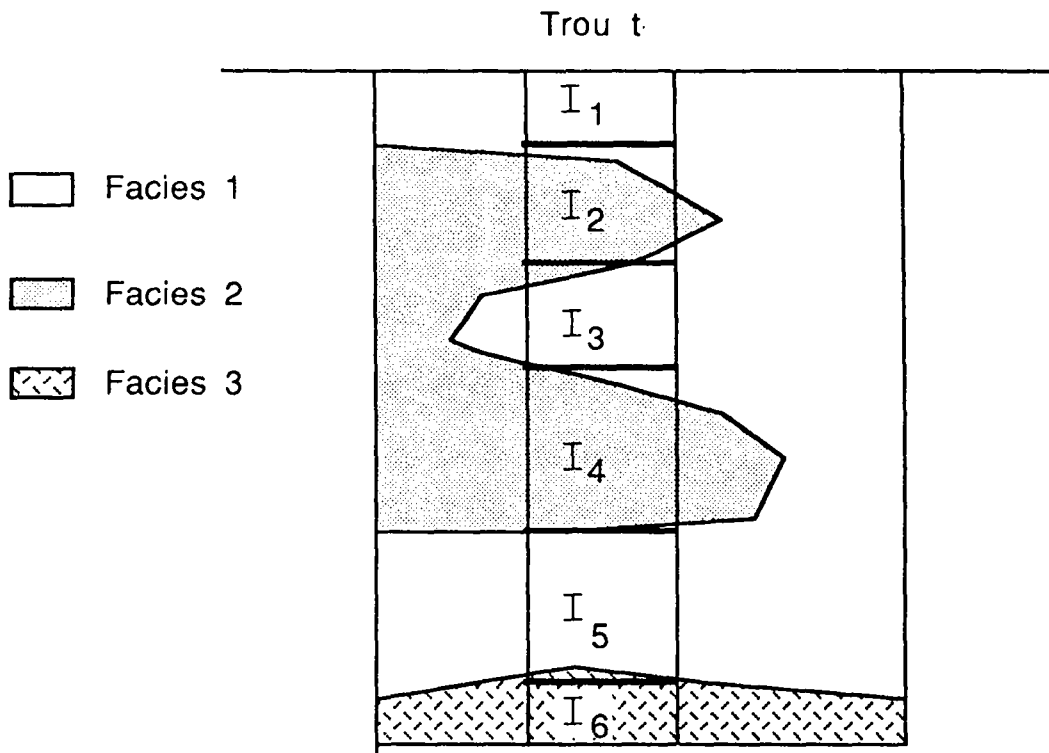
Si les variables sont qualitatives nous pouvons mesurer l'homogénéité d'un intervalle par la variance du nuage des éléments muni de la masse P_i et de la métrique du χ^2 .

2.3.3 Exemple d'application

Les images sismiques sont une sorte d'échographie du sous-sol. Une phase importante de l'interprétation sismique est la recherche de zones homogènes dites sismiques. L'homogénéité étant défini en termes d'attributs de texture locale (Keshes N., Baclanouan A., Lechevallier Y., Zaccagnino P., 1982). Ce problème peut être approché de la manière suivante.

A chaque point de l'image sismique sont associées ses coordonnées géographiques g_1 et g_2 et ses mesures x_1, \dots, x_p caractérisant ses textures locales. L'axe représentant la profondeur est un axe fondamental dans la recherche de ces zones homogènes nous l'avons choisi comme variable d'ordonnement. Afin que les valeurs de texture aient un sens statistique nous avons défini une large bande sur le 2e axe de l'image sismique et calculé pour une profondeur donnée les valeurs de texture mesurées sur cette bande. Ainsi nous avons utilisé cet algorithme pour rechercher les zones sismiques sur l'axe de la profondeur. Considérons un trou t de la section sismique. L'intersection de cette trace avec les différents facies présents dans la région est une suite de K intervalles I_k .

L'intersection d'un facies particulier avec la trace t est représenté soit par l'ensemble vide soit par un ou plusieurs intervalles.



Facies n°1 est représenté par les intervalles I_1, I_3 et I_5

Facies n°2 est représenté par les intervalles I_2 et I_4

Facies n°3 représenté par I_6

Autour de ce trou t , chaque intervalle I_k peut être vu comme l'intersection d'un facies particulier avec le trou t . Sous cette hypothèse les facies sismiques passant par le trou t réalisent en partitionnement de l'ensemble E des points de t en K classes $(B_1, \dots, B)_K$.

L'extension des limites de facies déterminées dans une sous-image aux autres parties de l'image est assurée grâce aux alignements, qui eux sont définis sur la totalité de l'image. Ces alignements introduisent de nouvelles limites dans les sous-images qui n'ont pu être trouvées ou bien retenues grâce à la texture assurant ainsi la continuité spatiale.

2.4 Reconnaissance et mémorisation condensée de contours

2.4.1 Introduction

Pour améliorer de façon condensée l'information fournie par une fonction $g(x)$ inconnue mais échantillonnée en différents points d'un intervalle $[a, b]$ il faut rechercher une fonction dépendant d'un petit

nombre de paramètres qui approche au mieux $g(x)$ au sens des normes L_2 ou L_m . En pratique les fonctions les plus utilisées sont les polynômes. Toutefois si la fonction $g(x)$ est complexe il est nécessaire de choisir soit un polynôme de degré élevé ou bien de découper l'intervalle $[a,b]$ en plusieurs intervalles et sur chaque intervalle choisit un polynôme de faible degré.

La première solution engendre souvent des instabilités numériques. La deuxième solution nécessite le choix des points limites associés à la division de l'intervalle $[a,b]$ en sous-intervalles. Aussi la clé du succès d'une approximation locale réside dans le choix de ces points limites. Dans ce paragraphe nous allons montrer que notre algorithme de programmation dynamique permet le choix de ces points limites.

2.4.2 Différents approches possibles

Généralement l'approximation locale d'une courbe Pavlidies et Horowitz (1973), Dorofeyuk, Ibragimli and Movsuniov (1976) est faite relativement à un seuil maximal d'erreur fixé a priori. Cette stratégie privilégie la qualité de l'approximation mais non la recherche de points limites stables. Dans ce cas on recherche une approximation numérique correcte de la courbe mais pas une interprétation possible des fonctions approximant localement chaque sous-intervalle défini par les points limites. Comme autres approches nous pouvons citer (Bellman (1961)) qui utilise la programmation dynamique et aussi (Dufour (1974)) qui proposent des techniques de régression.

En suivant le principe des Nuées Dynamiques nous cherchons simultanément une partition en k classes contingües et k représentations de ces classes parmi les fonctions d'approximation qui maximise le critère suivant

$$\text{Max}_{j = 1, K} \quad \text{Max}_{t \in [t_{j-1}, t_j]} |g(t) - f(t)|$$

Avec $g(x)$ étant la fonction observée inconnue sur $[a,b]$ et $f_j(x)$ la fonction calculée sur $[t_{j-1}, t_j[$.

Pratiquement la fonction $g(x)$ est connue sur T points de l'intervalle $[a,b]$ et la représentation f_j de l'intervalle $[t_{j-1}, t_j[$ est constituée des coefficients du polynôme. D'où, par échantillonnage et pour une représentation polynomiale d'ordre q , les critères s'écrivent :

$$\sum_{j=1}^K \sum_{t \in P_j} \left(y_t - \sum_{h=0}^q \alpha_{jh} t^h \right)^2 \quad (\text{norme } L_2)$$

ou bien

$$\text{Max}_{j=1, K} \quad \text{Max}_{t \in P_j} \quad \left| y_t - \sum_{h=0}^q \alpha_{jh} t^h \right| \quad (\text{norme } L_\infty)$$

où y_t est la valeur de la fonction g au point t échantillonné.

Aussi on associe une valuation w pour tout intervalle de $[a, b]$ de la manière suivante

$$w([t_j, t_{j+1}[) = \sum_{t \in P_j} \left(y_t - \sum_{h=0}^q \alpha_{jh} t^h \right)^2$$

Ainsi le problème peut être résolu en deux étapes.

Etape 1 : Connaissant un intervalle $[c, d[$ de $[a, b]$ trouver la meilleure approximation polynômiale de cet intervalle au sens de la norme L^2 ou L^∞ .

Etape 2 : Connaissant l'erreur d'approximation pour tout intervalle de $[a, b]$ trouver les meilleures $(K-1)$ points coupures permettant de décomposer cet intervalle en K intervalles disjoints. Ce découpage doit minimiser le critère suivant

$$W(P) = \sum_{j=1}^K w(P_j)$$

où P_j sont K intervalles disjoints de $[a, b[$ et tels que $\cup P_j = [a, b[$.

2.4.3 Approximation polynomiales avec contrainte

Les approximations précédentes ne sont dépendantes que de l'intervalle sur lequel elles sont calculées. Si la représentation globale doit être continue, dérivable...il faut ajouter des contraintes. Nous avons vu que le critère optimisé doit être une sommation de valuations indépendantes entre elles. Dans ce cas, si on veut utiliser notre algorithme pour résoudre

ce problème, il faut ajouter à chaque intervalle la borne de droite c'est à dire que l'ensemble des intervalles se définit maintenant ainsi : $[t_j, t_{j+1}]$. Dans ce cas nous imposons que le polynôme de degré q passe par les points t_{j-1} et t_j ce polynôme est une fonction spline. Dans le cas où la fonction spline est d'ordre $(q-1)$ il ne s'agit plus d'estimation puisque nous avons $(q+1)$ contraintes sur $(q+1)$ coefficients.

On peut généraliser ceci au cas d'une fonction définie par les couples $(X_t, Y_t)_{t=1, T}$ observés. C'est le cas en cartographie des courbes de niveaux. Ici nous emploierons simultanément la regression et la classification sous contrainte. Comme les contraintes sont importantes nous démontrons que notre solution est optimale au sens d'un critère des moindres-carrés. De plus cette approche présente l'avantage de pouvoir introduire des contraintes de continuité, de dérivabilité aux points limites, confiant aussi ces propriétés à la fonction approximante globale. A la fin la condensation de l'information est obtenue par la mémorisation des points limites des intervalles et les paramètres de la fonction approximante. (Charles (1977)), (Charles et Lechevallier (1980)).

La formalisation du problème

Le but est de rechercher les $(K-1)$ points coupures t_1, t_2, \dots, t_{k-1} de $[a, b]$ avec

$$a = t_0 < t_1 \dots < t_{k-1} < t_k = b$$

tels que

$$\sum_{j=1}^K \int_{t_{j-1}}^{t_j} (g(x) - f_j(x))^2 dx$$

2.4.4 Recherche de la solution optimale

Dans tous les cas, les approximations polynomiales avec ou sans contraintes ne dépendent que de l'intervalle sur lequel elles sont calculées, et le critère global est la somme des résidus sur chaque intervalle. Aussi ce critère est bien additif et peut s'écrire sous la forme

$$W(P) = \sum_{h=1}^K W(I_h)$$

où I_k est un intervalle de $[a, b]$. Si on n'impose pas la continuité et la dérivabilité de la fonction sur tout $[a, b]$ les intervalles I_k sont d'intersections vides, sinon l'intersection de ces intervalles représente les

points frontières entre ces intervalles et ces points frontières sont pris dans l'ensemble des points de la courbe échantillonnés.

BIBLIOGRAPHIE

Bellman. "On the approximation of curves by line segments using dynamic programming" Communication of the ACM (1961).

Bellman R.E et Dreyfus S.E. "Applied Dynamic Programming", Princeton, n°5, (1962).

Celeux G., Diday E., Govaert G., Lechevallier Y., Ralambondrainy H. "Classification automatique des données : environnement statistique et informatique", Dunod 1989.

Charles C. "Régression Typologique et Reconnaissance des formes", Thèse 3ème cycle, Université Paris IX-Dauphine,(1977), *Journées Internationales d'Analyse des Données* (1980).

Charles C. et Lechevallier Y. "Compression de courbes par approximation polynomiales a noeuds variables" - *Journées Internationales d'Analyse des Données* (1980).

Delattre M et Hansen P. "Bicriterion cluster Analysis" Research Report, Faculté Universitaire Catholique de Mons, (1978).

Diday E. et collaborateurs (1980), "Optimisation en classification automatique". INRIA.

Duran B.S et Odell P.L. "Cluster Analysis : A Survey", Lectures Notes in Economics and Mathematical Systems, n°100, Springer Verlag (1974).

Dufour "Programmes de mémorisation de profils et contours sous forme condensée". I.G.N (1974)

Dorofeyuk A.A., Ibragimli ShD, Mouserniov V.G "The use of the statistical equivalence criterion in the piecewise approximation problem" *Automatica i telemechanica* n°7 (1976)

Everitt B. "Cluster Analysis", Heinemann (1974).

Fisher W.D. "On grouping for maximum homogeneity", *J.A.S.A* 53, (1958), pp 789-98.

Fisher R.A. (1936), "*The use of multiple measurements in taxonomic problems*". *Eugenics* Vol. 7.

Fisher W.D. (1958), "*On grouping for maximum homogeneity*". *JASA* Vol. 53.

Garfinkel R., Nemhauser G.L. (1969), "*The set-partitioning problem set covering with equality constraints*". *Operat. Research* Vol. 17.

Goodman L.A et Kruskal W.H. "*Measures of association for cross classifications*", *J.A.S.A.*, n°49, (1954) pp 732-64.

Goodman L.A et Kruskal W.H. "*II : Further discussion and references*", *J.A.S.A.*, n°54, (1959) pp 123-63.

Goodman L.A et Kruskal W.H. "*III : Approximate Sampling Theory*", *J.A.S.A.*, n°58, (1963) pp310-64.

Goodman L.A et Kruskal W.H. "*IV : Simplification of asymptotic variances*", *J.A.S.A.*, n°67, (1972) pp 415-21.

Hartigan J. (1975), "*Clustering algorithms*". Wiley.

Horowitz, Pavlidis. "*Picture segmentation by a directed split and merge procedure*" Department of electrical Engineering, Princeton (1974).

Hubert L. (1974), "*Some applications of graph theory to clustering*". *Psychometrika* Vol. 39.

Jensen R.E. (1969), "*A dynamic programming algorithm for cluster analysis*". *Operat. Research* Vol. 12.

Lechevallier Y. (1979), "*Classification optimale sous contrainte d'ordre total*". Rapport de recherche IRIA n° 200.

Prim R.C. (1957), "*Shortest connection network and some generalizations*". *Bell System Tech. Journal* Vol. 36.

Rao M.R. "*Cluster Analysis and Mathematical Programming*" *J.A.S.A.* (1971), pp 622-26.

Saporta G. "*Liaisons entre plusieurs ensembles de variables et codage de données qualitatives*" - Thèse 3^{ème} cycle (1975)

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249 - 6399