N° d'ordre : 1233

THESE

PRESENTEE A

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE I

POUR OBTENIR

LE GRADE DE DOCTEUR DE 3EME CYCLE

SPECIALITE : MATHEMATIQUES APPLIQUEES

PAR

PLACE CHRISTIANE

ALGORITHMES DE CLASSIFICATION SOUS CONTRAINTES ET METHODES MULTICRITERES

THESE SOUTENUE LE 12 DECEMBRE 1984 DEVANT LA COMMISSION D'EXAMEN :

Président : P. POUZET (Université de Lille I)

Directeur de Recherche et Rapporteur : C. LANGRAND (Université de Lille I)

Examinateurs { D. BOSQ (Université de Lille I)
G. BARBAISE (Directeur d'Etudes, COREF-CEGOS (Paris))

A mes parents,

Monsieur le Professeur Pierre POUZET m'a fait l'honneur d'accepter la présidence du jury. Qu'il trouve ici le témoignage de ma reconnaissance.

Je remercie très vivement Monsieur le Professeur Denis BOSQ, qui a accepté de juger ce travail et qui, en maîtrise et en D.E.A., m'a donné le goût de la Statistique.

Mes très sincères remerciements vont également à Monsieur Gérard BARBAISE, Directeur d'Etudes à COREF-CEGOS. Il a bien voulu faire partie du jury et m'a fait découvrir la pratique de l'Analyse des Données dans le cadre d'une grande entreprise, lors d'un stage de six mois effectué en 1982 dans son service à La Redoute.

Monsieur le Professeur Claude LANGRAND, m'a proposé le sujet de cette thèse. Les conseils précieux qu'il n'a cessé de me prodiguer et les nombreuses discussions qu'il m'a si largement accordées ont été pour moi un constant encouragement. Sa rigueur et son sens du concret m'ont été d'une aide efficace tout au long de mes recherches. Qu'il trouve ici l'expression de ma profonde gratitude.

Arlette Lengaigne a dactylographié le texte avec patience et savoir faire, qu'il me soit permis de la remercier ainsi que Albert Gournay, Michel Provost et Monique Lloret qui ont réalisé avec goût le tirage offset et l'assemblage de la thèse.

I N T R O D U C T I O N.

Les méthodes de classification jouent depuis longtemps un rôle important dans de nombreuses disciplines (Biologie, Géographie, Sciences Economiques, Sociologie,...). Leur développement s'est accru considérablement ces dernières années : il est lié à la puissance sans cesse croissante des ordinateurs actuels qui permettent des calculs de plus en plus lourds et qui autorisent la mise en mémoire de grands fichiers, de base de données ... L'utilisation de méthodes de classification automatique doit cependant être faite avec prudence et il est souhaitable que l'ensemble des variables servant à décrire les individus à classifier soit suffisamment homogène pour que la classification puisse avoir un sens. Depuis quelques années, l'intérêt des statisticiens s'est porté sur la recherche de méthodes prenant en compte simultanément plusieurs critères. Parallèlement, et pour satisfaire à la demande de praticiens (géographes,...), des méthodes sous contraintes ont été étudiées afin que la classification fournie à l'aide d'un critère donne des classes assez homogènes suivant un autre critère.

Dans le premier chapitre, nous présentons une synthèse des méthodes récentes. La notion de contiguïté a été introduite dans les méthodes de classification, soit en n'autorisant l'agrégation de deux éléments que s'ils sont voisins - c'est le cas des méthodes fournissant des hiérarchies (Lebart, Perruchet,...) -, soit en donnant à chaque individu un poids fonction du classement de ses voisins (méthode de type Nuées Dynamiques de Barbaise et Langrand). Des méthodes prenant en compte plusieurs critères (Régnier ...) sont également étudiées dans ce chapitre.

Dans le second chapitre, nous proposons une approche de la classification sous contraintes. Cette méthode consiste à traduire la contiguïté à l'aide de variables. Ces variables sont ajoutées à celles retenues pour la description des individus : une classification "classique" sur cet ensemble de variables permet de tenir compte non seulement de la ressemblance du point de vue des variables descriptives, mais également de la contiguïté. Différentes méthodes permettant d'obtenir ces variables de contiguïté sont proposées. Comme celles-ci conduisent généralement à la recherche d'éléments propres d'une matrice symétrique, nous avons rappelé quelques algorithmes permettant leur calcul. Un exemple illustre les résultats de ce chapitre.

Le chapitre III décrit un algorithme de classification du type Nuées Dynamiques qui permet de prendre en compte des groupes de variables descriptives auxquels sont associées des distances de types différents entre les individus. En effet, nous n'utilisons pas directement la valeur de ces distances, mais les rangs obtenus par chacun des noyaux pour un individu fixé. Un programme FORTRAN et sa notice d'utilisation permettent la mise en oeuvre de cet algorithme. Ils font l'objet du chapitre IV. Deux exemples d'utilisation sont proposés.

CHAPITRE I

CLASSIFICATIONS SOUS CONTRAINTES ET MULTICRITERES.

Dans ce premier chapitre, nous présentons quelques méthodes de classification qui tiennent compte de contraintes lors de la formation des classes ou qui utilisent plusieurs critères simultanément.

1 - METHODES DE TYPE HIERARCHIQUE.-

1.1. - Rappels sur la classification ascendante hiérarchique et la classification hiérarchique selon l'algorithme de la vraisemblance du lien.

1.1.1. - La classification ascendante hiérarchique (CAH).

Cette méthode de classification procède par regroupements successifs de deux classes "les plus proches" et permet d'obtenir de nombreux résultats sur l'ensemble des données sans nécessiter d'hypothèses a priori sur la structure de cet ensemble.

Soient E l'ensemble à classifier, $P(E)$ l'ensemble de ses parties et d un indice de distance sur E, c'est-à-dire une application de $E \times E$ dans $\mathbb{R}_+$ vérifiant :

- $\forall (x,y) \in E^2 \qquad d(x,y) = 0 \iff x = y$
- $\forall (x,y) \in E^2 \qquad d(x,y) = d(y,x)$.

On construit la classification ascendante hiérarchique de la manière suivante :

1) On cherche deux éléments i et j de E les plus proches au sens de d ; on agrège ces deux éléments pour former une classe $\{i,j\}$.

2) On définit une application δ sur $P(E) \times P(E)$, dont la restriction aux singletons coïncide avec d, permettant de mesurer la proximité entre deux classes quelconques. On calcule alors $\delta(\{i,j\}, \{k\})$ pour $k \neq i$, $k \neq j$ et on agrège les deux classes les plus proches de $\{\{i,j\}, \{k\}$ pour $k \in E - \{i,j\}\}$.

3) On poursuit le processus (agrégation des deux classes les plus proches, calculs des "distances" entre classes) jusqu'au sommet de la hiérarchie où l'on rassemble dans une seule classe tous les éléments de E.

Cette méthode permet donc d'obtenir une suite de partitions emboîtées, l'utilisateur pourra choisir la plus adaptée à l'ensemble des données en tenant compte des niveaux d'agrégation.

Le principal inconvénient de cet algorithme est qu'il nécessite un espace mémoire et un temps de calcul importants pour des grands ensembles de données.

1.1.2. - La classification hiérarchique selon l'algorithme de la vraisemblance du lien (AVL).

Cette méthode proposée par I.C. Lerman [11] utilise une notion originale de proximité entre variables, pour effectuer une classification d'un ensemble de variables.

Pour introduire cette proximité, plaçons nous dans le cas où les variables sont des attributs descriptifs. A un attribut a correspond E_a l'ensemble des éléments de E qui le possèdent. Il est clair que la valeur de $s = \text{card}(E_a \cap E_b)$ joue un rôle important dans la proximité entre les deux attributs a et b. Cependant l'utilisation "brute" de la valeur de s présente un grand inconvénient puisqu'elle ne tient pas compte des cardinalités respectives des ensembles E_a et E_b. Elle peut donc être "grande" sans pour cela être significative de la proximité entre les attributs a et b. C'est pourquoi, il est nécessaire d'introduire une hypothèse N d'absence de lien ou d'ignorance a priori de la position relative de E_a et E_b. Sous cette hypothèse N, deux variables aléatoires S_a et S_b sont définies respectivement sur E_b ou E_a, ensemble des parties de E de cardinal $n_b = \text{card } E_b$ ou $n_a = \text{card } E_a$:

$$S_a : (E_b, P(E_b)) \longrightarrow (U, P(U))$$

$$S_b : (E_a, P(E_a)) \longrightarrow (U, P(U))$$

où $U = \{0,1,\ldots,\min(n_a,n_b)\}$

avec $\forall Y \in E_b \qquad S_a(Y) = \text{card}(E_a \cap Y)$

$\forall X \in E_a \qquad S_b(X) = \text{card}(X \cap E_b)$

$P(E_b)$ et $P(E_a)$ étant munis d'une loi de probabilité uniforme sous l'hypothèse N, les lois des variables aléatoires S_a et S_b sont identiques (hypergéométriques). Notons qu'en général, la loi de S_a (ou de S_b) peut être approchée par une loi normale. On jugera de la proximité entre les attributs a et b à l'aide de :

$$p(a,b) = \Pr^N(\{S_a < s\}) = \Pr^N(\{S_b < s\})$$

deux attributs a et b étant jugés d'autant plus voisins que la valeur de s est invraisemblablement grande sous l'hypothèse N d'absence de lien.

A la première étape, on agrégera les deux attributs pour lesquels p(a,b) est maximum.

Pour poursuivre l'algorithme, il est nécessaire d'étendre la mesure de proximité à des sous-ensembles disjoints de l'ensemble des variables.

Soient C (de cardinal ℓ) et D (de cardinal m) deux tels ensembles ; Lerman propose comme mesure de proximité de ces classes :

$$p(C,D) = \left[\max_{(c,d)\in C\times D} p(c,d) \right]^{\ell m} ,$$

les classes agrégées à chaque étape étant celles pour lesquelles p(C,D) est maximum.

Remarque : L'indice de proximité a été présenté entre deux attributs, il peut être calculé entre deux structures de même type sous la même hypothèse d'absence de lien ou d'ignorance a priori de la position relative des deux structures.

1.2. - *Classifications hiérarchiques sous contraintes.*

1.2.1. - La contrainte de contiguïté a été introduite en 1977 par P. Monestiez [13] dans la CAH. Les individus qu'il cherche à classer sont des points ou des parcelles d'un champ spatial, décrits par plusieurs variables. La classification des points obtenue par l'algorithme classique tient compte des ressemblances du point de

vue des variables et il est nécessaire de cartographier les partitions retenues dans l'arbre de la CAH pour visualiser les positions géographiques. Dans l'algorithme proposé, deux points ou deux classes ne seront agrégés que s'ils sont contigus. L'introduction d'une contrainte de contiguïté a donc pour but de favoriser la formation de partitions dont les classes sont géographiquement connexes.

- L. Lebart [9] propose également un algorithme de CAH avec contraintes. La méthode qu'il présente, permet en outre un gain de temps et des économies de mémoire appréciables. En effet, les distances ne sont plus calculées pour chaque couple d'individus, mais uniquement pour ceux qui sont contigus. De plus, la matrice booléenne associée au graphe de contiguïté ne figure pas en mémoire centrale, seules les adresses de "1" (autrement dit, les numéros des voisins) sont stockées. L'espace mémoire nécessaire à la mise en oeuvre du programme est donc considérablement diminué, le temps d'exécution également puisque le nombre de tests pour la recherche des couples d'individus les plus proches se fait dans l'ensemble restreint aux couples de voisins.

- C. Roche [17] présente une application de cette méthode. Il cherche à obtenir un découpage en quartiers homogènes de la ville d'Aix-en-Provence en vue d'une prévision de demandes téléphoniques.

1.2.2. - Plus récemment, A. Prod'homme [15], utilisant la classification hiérarchique selon l'AVL, a introduit une contrainte de contiguïté dans la formation des classes. Cet algorithme a été présenté d'une part pour répondre aux géographes, d'autre part pour résoudre le problème de classification dans le cas d'un tableau d'incidence "creux".

La préoccupation des géographes est généralement d'obtenir des classes d'individus non seulement semblables à travers les variables sélectionnées pour les repérer mais également proches sur la carte : les classes agrégées à un niveau de l'arbre doivent donc répondre aux critères suivants :

- elles sont contiguës ,

- l'indice de proximité entre elles est le plus grand possible, si on ne peut le prendre maximal.

L'utilisation de la notion de contiguïté pour la classification des lignes ou des colonnes d'un tableau d'incidence "creux" permet de pallier aux difficultés dûes à l'approximation normale faite dans l'AVL.

Nous avons vu au paragraphe 1.1.2. que la distribution de la variable aléatoire S_a associée à l'indice s pouvait être approchée par une loi normale. Or cette approximation peut s'avérer médiocre si l'indice s est "trop petit". Deux attributs (ou classes d'attributs) étant contigus si l'indice s qui leur est associé est supérieur ou égal à un seuil n_o préalablement fixé, effectuer l'AVL sous contrainte de contiguïté permettra d'éviter le problème précédent : deux classes C et D seront agrégées si :

. $p(C,D)$ est le plus grand possible ,

. $\exists (c,d) \in C \times D$ tel que $card(E_c \cap E_d) \geq n_o$.

1.2.3. - C. Perruchet [14] présente une utilisation de la contiguïté dans la CAH, différente de celle rencontrée dans les méthodes précédemment citées. La classification proposée s'applique à tout ensemble de données repérées dans deux espaces distincts : l'un d'eux a une interprétation "géographique" , l'autre est l'ensemble usuel des variables mesurées sur les éléments à classer. Contrairement aux méthodes précédentes, la contiguïté intervient à deux niveaux, d'une part dans le choix des paires à agréger, d'autre part dans la pondération de l'indice δ utilisé dans le programme de classification pour mesurer les "distances" entre classes. Il est ici nécessaire de connaître les proximités spatiales entre les individus. Ceci n'est pas très gênant puisque si les individus sont des points géographiques (cas le plus fréquent d'utilisation de la contiguïté), ils sont repérables dans $\mathbb{R}^2$ ou $\mathbb{R}^3$. Si d désigne la distance dans l'espace géographique et δ l'indice associé à l'espace des variables, l'indice γ minimisé dans l'algorithme est défini par :

$$\forall (s,s') \in S^2 \qquad \gamma(s,s') = d(s,s')\delta(s,s')$$

où S désigne l'ensemble des individus ou des classes à agréger. On appellera sommet un élément de S.

La valeur de cet indice est calculée pour les paires de sommets contigus. Il est donc nécessaire de définir la notion de contiguïté non seulement entre les individus de E (ensemble à classifier) mais également entre les sommets :

- deux individus i et j de E sont dits contigus si $d(i,j) < \varepsilon$ où ε est un seuil donné par l'utilisateur.

- le nouveau sommet formé par l'agrégation de deux sommets est contigu à tous les sommets contigus à l'un quelconque des deux sommets qui le forment.

Si le seuil ε choisi est trop petit, il peut arriver qu'il n'y ait plus de paires de sommets contigus, donc plus d'agrégation possible. Dans ce cas, un nouveau seuil est déterminé et l'agrégation se poursuit sur les sommets du dernier arbre construit.

2 - METHODE DE TYPE NUEES DYNAMIQUES.-

2.1. - Rappels sur la méthode des nuées dynamiques.

Cette méthode de classification non hiérarchique proposée par E. Diday [6] permet d'améliorer une partition en k classes d'un ensemble donné E, dans le sens où la partition construite est obtenue de manière itérative en minimisant, à chaque étape, un certain critère W. La construction de cette partition se fait à partir de k noyaux, un noyau étant un élément ou un ensemble d'éléments qui représente au mieux la classe à laquelle il est associé. Elle nécessite également la donnée de deux applications f et g, f construisant une partition de E à partir de k noyaux, g associant à une partition de E un ensemble de k noyaux.

Pour décrire l'algorithme, nous utiliserons les notations suivantes :

- E désigne l'ensemble fini à classifier ,
- $]k] = \{1,\ldots,k\}$ l'ensemble des k premiers entiers positifs ,
- $\mathbb{P}_k$ l'ensemble des partitions de E en k classes :

$$P \in \mathbb{P}_k \qquad P = (P_1,\ldots,P_k)$$

- $\mathbb{L}$ l'espace des noyaux ($\mathbb{L}$ peut être en particulier l'ensemble E lui-même).

- D sert à mesurer la "distance" entre un élément de E et un élément de $\mathbb{L}$:

$$D : E \times \mathbb{L} \longrightarrow \mathbb{R}_+$$

$$(x,\lambda) \longmapsto D(x,\lambda)$$

- R mesure l'ajustement d'un noyau λ à un élément P_i de la partition P

$$R : \mathbb{L} \times]k] \times \mathbb{P}_k \longrightarrow \mathbb{R}_+$$

$$(\lambda,i,P) \longmapsto R(\lambda,i,P) \quad .$$

Les fonctions f et g sont alors définies par :

- $$f : \mathbb{L}^k \longrightarrow \mathbb{P}_k$$

$$L = (\lambda_1,\ldots,\lambda_k) \longmapsto P = (P_1,\ldots,P_k)$$

où $\forall i \in]k] \quad P_i = \{x \in E : D(x,\lambda_i) \leqslant D(x,\lambda_j) \quad \forall j \in]k]\}$

si x peut appartenir à deux classes, il est mis dans celle de plus petit indice.

- $$g : \mathbb{P}_k \longrightarrow \mathbb{L}^k$$

$$P = (P_1,\ldots,P_k) \longmapsto L = (\lambda_1,\ldots,\lambda_k)$$

où $\forall i \in]k] \quad \lambda_i$ vérifie $R(\lambda_i,i,P) = \min_{\lambda \in \mathbb{L}} R(\lambda,i,P)$.

Ceci suppose donc qu'il n'y a pas de problèmes d'existence et d'unicité des différents λ_i .

Le critère W s'écrit : $W(L,P) = \sum_{i \in]k]} R(\lambda_i, i, P)$.

L'algorithme se déroule alors de la façon suivante : à partir d'un k-uple de noyaux L^o et de f on construit P^o, puis L^1 en utilisant g et P^o ,...

Sous certaines conditions sur R , le critère W décroît à chaque itération et la suite (L^n,P^n) converge vers une limite (L^*,P^*) qui dépend, en général du L^o de départ. Pour remédier à cela, la méthode est souvent appliquée plusieurs fois avec des L^o différents de façon à mettre en évidence des groupes stables qui sont les sous-ensembles d'éléments toujours classés ensemble.

Cette méthode qui donne des résultats généralement moins précis qu'une CAH présente l'avantage de pouvoir traiter un grand ensemble de données en un temps très court.

2.2. - Dans [1] G. Barbaise et C. Langrand présentent une approche de la classification sous contraintes en introduisant la notion de contiguïté dans la méthode des nuées dynamiques. Cependant cette notion n'est pas utilisée de la même façon que dans les méthodes précédentes ; elle a ici un rôle de test statistique. La partition des individus obtenue au vu des valeurs qu'ils prennent sur les variables utilisées dans l'analyse, n'est remise en question par la considération des voisins que si cela est réellement nécessaire. En effet, même si les voisins d'un individu sont mis dans un autre groupe que lui, il ne sera "reclassé" avec eux que s'il est très proche d'eux au sens des variables.

Pour décrire l'algorithme proposé, nous reprenons une grande partie des notations utilisées au paragraphe précédent. Il est ici nécessaire d'introduire la notion de voisinage : nous supposons que E est muni d'une métrique d. Pour $x \in E$, $\varepsilon \in \mathbb{R}_+^*$, $K \in \mathbb{N}^*$ nous définissons $V(x,\varepsilon,K)$ par :

$$V(x,\varepsilon,K) = \{y \in E : d(x,y) < \varepsilon\} \cap \{K \text{ plus proches voisins de } x\}$$

La première étape se déroule comme dans l'algorithme classique : L^o donné, P^o est défini par :

$$\forall i \in]k] \qquad P_i^o = \{x \in E : D(x,\lambda_i^o) \leqslant D(x,\lambda_j^o) \ , \ \forall j \in]k]\}$$

Une fois P^o déterminée, on définit pour tout $i \in]k]$ et tout $x \in E$:

$$N_i^o(x) = \text{card}(V(x,\varepsilon,K) \cap P_i^o) + 1 \ .$$

La fonction R mesurant l'ajustement d'un noyau à une partition devient :

$$R(\lambda,i,P^o) = \sum_{x \in P_i^o} D(x,\lambda)\ N_i^o(x) \ .$$

Le noyau L^1 est tel que : $\forall i \in]k]$ $R(\lambda_i^1,i,P^o) = \min_{\lambda \in \mathbb{L}} R(\lambda,i,P^o)$.

Le nombre de voisins intervient également dans la construction de P^1 à partir de L^1 :

$$\forall i \in]k] \quad P_i^1 = \{x \in E : D(x,\lambda_i^1)\ N_i^o(x) \leqslant D(x,\lambda_j^1)\ N_j^o(x) \quad \forall j \in]k]\}.$$

On définit ensuite les nombres $N_i^1(x)$ par

$$\forall i \in]k] \quad \forall x \in E \qquad N_i^1(x) = \min(\text{card}(V(x,\varepsilon,K) \cap P_i^1) + 1 \ , \ N_i^o(x))$$

L^2 est alors tel que : $\forall i \in]k]$ $R(\lambda_i^2, i, P^1) = \min_{\lambda \in \mathbb{L}} R(\lambda, i, P^1)$

avec $R(\lambda, i, P^1) = \sum_{x \in P_i^1} D(x, \lambda)\ N_i^1(x)$.

Pour $n \geqslant 2$, les itérations se déroulent de la façon suivante :

- à partir de $P^n = (P_1^n, \ldots, P_k^n)$ on calcule :

$\forall x \in E \quad \forall i \in]k] \quad N_i^n(x) = \min(\text{card}(V(x,\varepsilon,K) \cap P_i^n) + 1,\ N_i^{n-1}(x))$

$L^{n+1} = (\lambda_1^{n+1}, \ldots, \lambda_k^{n+1})$ est alors donné par :

$\forall i \in]k] \quad R(\lambda_i^{n+1}, i, P^n) = \min_{\lambda \in \mathbb{L}} R(\lambda, i, P^n)$

avec $R(\lambda, i, P^n) = \sum_{x \in P_i^n} D(x, \lambda)\ N_i^n(x)$

- à L^{n+1} correspond $P^{n+1} = (P_1^{n+1}, \ldots, P_k^{n+1})$ telle que :

$\forall i \in]k] \quad P_i^{n+1} = \{x \in E : D(x, \lambda_i^{n+1})\ N_i^n(x) \leqslant D(x, \lambda_j^{n+1})\ N_j^n(x)\ \forall j \in]k]\}$

Le critère W , qui s'écrit $W(L,P) = \sum_{i \in]k]} R(\lambda_i, i, P)$ décroît à chaque étape, et la suite (L^n, P^n) converge vers (L^*, P^*).
Comme pour la méthode classique (L^*, P^*) dépend de L^o ; l'algorithme est donc appliqué plusieurs fois avec des noyaux différents. Les groupes stables obtenus tiennent compte à la fois de la ressemblance du point de vue des variables et des voisins au sens de d. Cependant l'importance de la contiguïté dans le résultat est plus faible que dans les méthodes précédemment citées, où deux individus proches au sens des variables ne sont réunis que s'ils sont voisins.

3 - RECHERCHE D'UN ENSEMBLE DE PARTITIONS "MULTICRITERES".-

3.1. - Méthode des partitions centrales.

Cette méthode présentée par S. Régnier [16] a pour but d'obtenir une classification sur un ensemble E de cardinal n dont les éléments sont décrits par un nombre fini de caractères $\{a_1,\ldots,a_p\}$. L'ensemble des différentes modalités prises par un caractère a_h est noté A_h. Régnier suppose que ces ensembles A_h sont finis et sans structure naturelle ; chaque caractère a_h définit une partition P^h de E. Les classes de P^h sont formées d'éléments identiques du point de vue du caractère a_h :

$$x \text{ et } y \text{ dans la même classe } \Longleftrightarrow a_h(x) = a_h(y)$$

Les p caractères définissent donc p partitions, en général différentes, de E. Le problème est alors de construire une (ou plusieurs) partition de E qui tienne compte "au mieux" de ces p partitions.

Soit $\mathbb{P}$ l'ensemble des partitions de E, la distance d utilisée sur $\mathbb{P}$ est :

$d(P,P')$ = cardinal de la différence symétrique des graphes dans $E \times E$ associés respectivement aux relations d'équivalence définies par P et P' .

Cette distance d induit une distance δ entre p-uples de partitions :

$$\delta((P^1,\ldots,P^p),(Q^1,\ldots,Q^p)) = \frac{1}{p}\sum_{h=1}^{p} d(P^h,Q^h)$$

Régnier cherche alors les partitions P qui minimisent $\delta((P^1,\ldots,P^p),(P,\ldots,P))$, où $(P^1,\ldots,P^p)$ sont les p partitions

construites à partir des caractères descriptifs $\{a_1 \ldots a_p\}$, et il les appelle partitions centrales.

Pour la recherche pratique de P, il est nécessaire d'introduire une expression simplifiée de :

$$\delta((P^1,\ldots,P^p),\ (P,\ldots,P)) = \frac{1}{p} \sum_{h=1}^{p} d(P^h,P) \qquad (1)$$

A toute partition P correspondent n^2 nombres $\bar{\omega}_{xy}$ définis par :

$$\forall (x,y) \in E^2 \quad \bar{\omega}_{xy} = \begin{cases} 1 & \text{si } x \text{ et } y \text{ sont dans la même classe de } P \\ 0 & \text{sinon .} \end{cases}$$

La fonction $\bar{\omega}$ ainsi définie à partir de P : $(x,y) \longmapsto \bar{\omega}_{xy}$, est un élément de $\{0,1\}^{E\times E}$; réciproquement à tout élément $\bar{\omega}$ de $\{0,1\}^{E\times E}$ vérifiant :

$$\forall (x,y,z) \in E^3 \quad \begin{cases} \bar{\omega}_{xy} = \bar{\omega}_{yx} \\ \bar{\omega}_{xx} = 1 \\ \bar{\omega}_{xy} = 1 \text{ et } \bar{\omega}_{yz} = 1 \Longrightarrow \bar{\omega}_{xz} = 1 \end{cases} \qquad (2)$$

est associé une partition de E. L'ensemble $\mathbb{P}$ peut donc être considéré comme la partie de $\{0,1\}^{E\times E}$ dont les éléments vérifient (2).

Plongeant $\{0,1\}^{E\times E}$ dans $\mathbb{R}^{E\times E}$ qui est isomorphe à $\mathbb{R}^{n\times n}$, on définit alors une distance D sur l'ensemble des relations binaires sur E :

$$D^2(\rho,\rho') = \sum_{(x,y)\in E\times E} (\rho_{xy} - \rho'_{xy})^2 .$$

Pour $\bar{\omega}$ et $\bar{\omega}'$ associées aux deux partitions P et P', on a alors

$$D^2(\bar{\omega},\bar{\omega}') = d(P,P') .$$

(1) s'écrit : $\frac{1}{p} \sum_{h=1}^{p} D^2(P^h,P) = \frac{1}{p} \sum_{h=1}^{p} \sum_{(x,y)\in E\times E} (s^h_{xy} - \bar{\omega}_{xy})^2$

où s^h et $\bar{\omega}$ sont les éléments de $\{0,1\}^{E\times E}$ associés respectivement à P^h et P.

(1) peut aussi s'écrire $\frac{1}{p} \sum_{h=1}^{p} ||s^h - \bar{\omega}||^2$ où $||.||$ désigne la norme euclidienne dans $\mathbb{R}^{n\times n}$.

Travaillant dans $\mathbb{R}^{n\times n}$, et définissant l'isobarycentre s des points s^h par : $s = \frac{1}{p} \sum_{h=1}^{p} s^h$, on a

$$\frac{1}{p} \sum_{h=1}^{p} ||s^h - \bar{\omega}||^2 = ||s - \bar{\omega}||^2 + \frac{1}{p} \sum_{h=1}^{p} ||s^h - s||^2 .$$

Le second terme du membre de droite est fixe, par conséquent minimiser (1) revient à minimiser $||s - \bar{\omega}||^2$. De plus, comme $\bar{\omega}_{xy}$ ne prend que les valeurs 0 et 1, toutes les applications $\bar{\omega}$ sont à la même distance $(n^2/4)$ du point $\gamma \in \mathbb{R}^{n\times n}$ dont les coordonnées valent $\frac{1}{2}$.

$||s - \bar{\omega}||^2 = ||\bar{\omega} - \gamma||^2 + ||s - \gamma||^2 - 2 \langle \bar{\omega} - \gamma, s - \gamma \rangle$

$||\bar{\omega} - \gamma||^2$ et $||s - \gamma||^2$ sont fixes, donc minimiser $||s - \bar{\omega}||^2$ revient à maximiser $\langle \bar{\omega} - \gamma, s - \gamma \rangle$, soit

$$\sum_{(x,y)\in E\times E} (\bar{\omega}_{xy} - \tfrac{1}{2}) (s_{xy} - \tfrac{1}{2}).$$

Pour obtenir une partition centrale, il suffit donc de maximiser $\sum_{(x,y)\in E\times E} (s_{xy} - \frac{1}{2}) \bar{\omega}_{xy}$ sous les contraintes :

$$\forall (x,y,z) \in E^3 \quad \begin{cases} \bar{\omega}_{xy} = 0 \quad \text{ou} \quad 1 \\ \bar{\omega}_{xx} = 1 \\ \bar{\omega}_{xy} = \bar{\omega}_{yx} \\ \bar{\omega}_{xy} = 1 \quad \text{et} \quad \bar{\omega}_{yz} = 1 \Longrightarrow \bar{\omega}_{xz} = 1 \end{cases}$$

Remarque : De Falguerolles propose dans [3] un critère de classification qui permet, dans le cas où les données sont des partitions d'un ensemble, de retrouver les partitions centrales de Régnier.

3.2. - *Classification optimale bicritère.*

M. Delattre [4], [5] présente une méthode de classification fondée sur l'optimisation conjointe d'un critère d'homogénéité et d'un critère de séparation des classes ; son algorithme lui permet de minimiser les inconvénients des deux méthodes basées sur l'optimisation d'un seul de ces critères. En effet, l'utilisation d'une méthode de classification maximisant uniquement l'homogénéité peut conduire à mettre dans des classes différentes, deux éléments pour lesquels la mesure de dissimilarité est la plus faible (effet de dissection), alors qu'une méthode qui détermine les partitions de séparation maximum conduit parfois à regrouper des objets différents dans une même classe (effet de chaînage).

Pour donner l'expression des critères, nous utiliserons les notations du paragraphe 2.1. et nous appellerons d un indice de dissimilarité entre éléments de E :

$$d : E \times E \longrightarrow \mathbb{R}^+ .$$

La séparation d'une partition $P \in \mathbb{P}_k$ est définie par Delattre comme étant la plus petite dissimilarité entre deux individus appartenant à des classes différentes de cette partition :

$$s(P) = \min_{j \in]k]} \min_{\substack{x \in P_j \\ y \notin P_j}} d(x,y) \quad \text{où} \quad P = (P_1, \ldots, P_k) .$$

Une partition P^s de $\mathbb{P}_k$ de séparation maximum vérifie :

$$s(P^s) = \max_{P \in \mathbb{P}_k} s(P) .$$

L'homogénéité d'une partition $P \in \mathbb{P}_k$ est mesurée à l'aide de :

$$\text{diamètre de } P = d(P) = \max_{j \in]k]} \max_{\substack{x \in P_j \\ y \in P_j}} d(x,y) \quad \text{où} \quad P = (P_1, \ldots, P_k) ,$$

qui est la plus grande dissimilarité entre deux individus appartenant à une même classe.

Une partition P^h de $\mathbb{P}_k$ d'homogénéité maximum vérifie :

$$d(P^h) = \min_{P \in \mathbb{P}_k} d(P) .$$

Ces deux critères sont souvent en conflit : l'optimisation de l'un d'eux conduit à une mauvaise valeur de l'autre.

La méthode proposée par Delattre vise à déterminer l'ensemble des solutions "efficaces" du problème de classification bicritère, une partition étant dite efficace s'il n'est pas possible d'améliorer la valeur de l'un des deux critères sans perdre sur la valeur de l'autre.

Son algorithme consiste en la fusion de l'algorithme de classification "single - link" qui fournit les partitions de séparation maximum et d'un algorithme de classification maximisant

l'homogénéité et basé sur une technique de coloration de graphe.

Pour déterminer toutes les partitions efficaces en k classes :

- on commence par construire la partition d'homogénéité maximum P^h en k classes : le diamètre d'une partition efficace est supérieur ou égal au diamètre de P^h .

- on recherche les partitions P^s de séparation maximum en k , $k+1$,..., $n-1$ classes.

- on examine successivement ces différentes partitions en partant de celle en k classes. A la partition P^s en M classes ($k \leqslant M \leqslant n-1$) considérée, on associe un graphe réduit dont les M sommets correspondent aux M classes ; les arêtes que l'on désire introduire sont les (x_i,x_j) tels que x_i et x_j sont dans des classes différentes et $d(x_i,x_j) \geqslant d(P^s)$. Chaque fois que le graphe réduit est k-colorable, la partition déduite de ce graphe est efficace.

Remarque : L'avantage des deux méthodes proposées dans ce paragraphe est qu'elles utilisent les différents critères au même niveau, ce qui n'est pas le cas dans les méthodes sous contraintes. Par contre, elles présentent l'inconvénient d'être plus lourdes au point de vue calcul : Delattre précise que sa méthode détermine les solutions efficaces d'ensembles contenant jusqu'à 300 objets à classifier, le problème de coloration optimale handicapant la vitesse de fonctionnement de ses programmes.

CHAPITRE II

CLASSIFICATIONS SOUS CONTRAINTES : UNE APPROCHE.

Dans ce chapitre, nous proposons une méthode de classification qui fait intervenir la notion de contiguïté.

Dans le chapitre I, la contiguïté intervient dans la construction des hiérarchies en n'autorisant l'agrégation de deux éléments que s'ils sont voisins. Dans [1], elle intervient à chaque étape de l'algorithme par une pondération de chaque individu, fonction du classement de ses voisins géographiques (ou plus généralement de ses voisins au sens de la métrique d) dans les différents éléments de la partition.

Ici la contiguïté est traduite à l'aide de variables qui sont prises en compte en même temps que celles choisies pour décrire les individus.

1 - TRADUCTION D'UNE CONTRAINTE DANS LES METHODES DE CLASSIFICATION.-

1.1. - Position du problème.

Si E désigne l'ensemble des individus à classifier et n son cardinal, on suppose disposer ici de la connaissance d'une relation binaire R, symétrique, définie sur $E \times E$ par :

$\forall (i,j) \in E^2$ $\quad i \; R \; j \iff i$ et j sont "voisins" .

On veut tenir compte de R lors de la classification que l'on effectue sur les éléments de E décrits par ailleurs par un certain

nombre de variables.

Dans le cas où E est un espace métrique ou peut être plongé dans un tel espace, on peut, par exemple, en notant $\mathcal{B}(e,\varepsilon)$ la boule de centre $e \in E$ et de rayon $\varepsilon > 0$, poser :

$$\forall (i,j) \in E^2 \qquad i \mathcal{R} j \Longleftrightarrow \mathcal{B}(i,\varepsilon) \cap \mathcal{B}(j,\varepsilon) \neq \emptyset$$

et définir ainsi une relation de voisinage entre éléments de E. Dans ce cas $\mathcal{R}$ est réflexive et symétrique. Dans bien des applications cependant, la relation $\mathcal{R}$ n'aura pas une telle définition mathématique ; elle pourra traduire la connaissance qu'a le praticien de l'ensemble E et dont il veut tenir compte lors de la classification, ou pourra être tout simplement imposée dans le cahier des charges de l'étude soumise à son analyse.

A une relation $\mathcal{R}$ réflexive et symétrique, on peut associer un indice de dissimilarité sur E c'est-à-dire une application δ définie sur $E \times E$ à valeurs dans $\mathbb{R}_+$, telle que :

$$\forall i \in E \qquad \delta(i,i) = 0$$

$$\forall (i,j) \in E^2 \qquad \delta(i,j) = \delta(j,i)$$

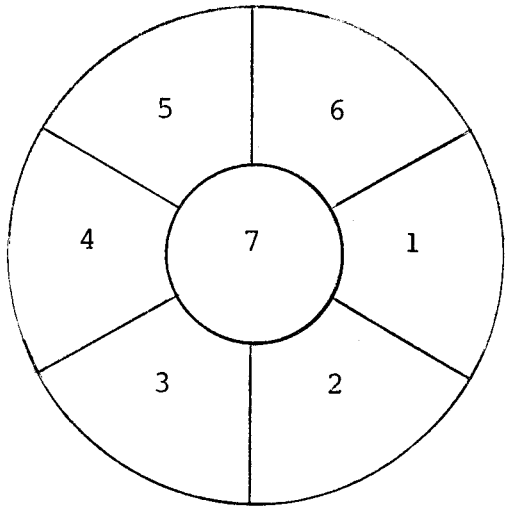
Il suffit en effet de poser par exemple :

$$\delta(i,j) = \begin{cases} 0 & \text{si } i \mathcal{R} j \\ 1 & \text{si } i \not\mathcal{R} j \end{cases} \qquad (1)$$

Si $\mathcal{R}$ n'est pas réflexive mais reste symétrique on peut poser pour tout couple (i,j) de E^2

$$\begin{cases} \delta(i,j) = 1 & \text{si} \quad i \; R \; j & i \neq j \\ \delta(i,j) = 2 & \text{si} \quad i \not R \; j & i \neq j \\ \delta(i,i) = 0 \end{cases} \qquad (2)$$

Si on considère les régions représentées ci-dessous



on définit R par $i \; R \; j$ si et seulement si $i \neq j$ et s'il existe un chemin allant de i à j en ne coupant qu'une seule frontière.

Notons que R n'est pas transitive et que δ défini par (2) est une distance.

Il est clair que s'il existe une image euclidienne de (E,δ), c'est-à-dire, s'il existe un espace euclidien $\mathcal{E}$ muni d'une métrique euclidienne M et une application h de E dans $\mathcal{E}$ tels que :

$$\forall (i,j) \in E^2 \qquad \delta(i,j) = ||h(i) - h(j)||_M ,$$

une analyse factorielle permet alors de simplifier cette image euclidienne : les coordonnées des individus sur les premiers axes principaux donnent une bonne approximation des voisinages (au sens de R) entre individus et peuvent donc être utilisées pour exprimer les proximités entre ces individus.

1.2. - Problème de l'existence d'une image euclidienne.

Remarquons que s'il existe une image euclidienne de (E,δ), les propriétés de la norme $||\ ||_M$ impliquent que nécessairement δ vérifie l'inégalité triangulaire et donc que δ est un écart ; de plus, si h est injective, δ est une distance.

Or l'indice de dissimilarité δ défini par (1) ne vérifie l'inégalité triangulaire que si R est transitive, donc si R est une relation d'équivalence. La contrainte imposée dans ce cas correspond à une classification préexistante des individus de E, E étant partitionné par les classes d'équivalence, éléments de E/R.

Dans le cas où R n'est pas transitive, rencontré dans bien des applications, le résultat précédent prouve qu'il n'existe pas d'image euclidienne "exacte" de (E,δ) quand δ est défini par (1). De manière générale, l'existence d'une représentation euclidienne de (E,δ) peut être assurée par le théorème suivant [12].

Théorème.-

Si Δ est la matrice symétrique (n,n) dont le terme général est défini par :

$$\Delta_i^j = \delta(i,j)$$

où δ est un indice de dissimilarité, et si

$$W = HAH^T$$

avec : A matrice (n,n) de terme général $a_i^j = -\frac{1}{2}\,\delta^2(i,j)$

$H = I_n - \mathbb{1}\mathbb{1}^T D_n$

$\mathbb{1}$ vecteur colonne de $\mathbb{R}^n$ dont toutes les composantes valent 1

D_n matrice diagonale telle que $(D_n)_i^i = p_i$,

$p_i > 0$, $\sum_{i=1}^{n} p_i = 1$,

alors il existe une représentation euclidienne de (E,δ) si et seulement si W est semi-définie positive.

Il est clair que la condition W semi-définie positive doit imposer l'inégalité triangulaire pour δ, puisque comme nous l'avons remarqué précédemment l'existence d'une image euclidienne de (E,δ) implique que δ soit un écart. On a en effet la proposition suivante :

Proposition.-

Soit V un espace vectoriel sur $\mathbb{R}$ de dimension finie n et soit W une forme bilinéaire sur $V \times V$, symétrique, semi-définie positive, dont le terme général est w_i^j ; on a pour tout i et $j \in]n]$,

$$w_i^i + w_j^j - 2\, w_i^j \geqslant 0 \ .$$

De plus, si δ est l'indice de dissimilarité définie sur $]n]^2$ par :

$$\delta^2(i,j) = w_i^i + w_j^j - 2\, w_i^j$$

alors δ vérifie l'inégalité triangulaire.

a) On a

$$\forall x \in V \qquad x^T W x \geqslant 0$$

où W est la matrice associée à la forme bilinéaire W (une base de V a été choisie).

Prenons pour vecteur x le vecteur de V dont toutes les composantes sont nulles, à l'exception de la $i^{ème}$ qui vaut 1 et la $j^{ième}$ qui vaut -1 ; on a :

$$x^T W x = w_i^i + w_j^j - 2\, w_i^j$$

On en déduit que :

$$\forall (i,j) \in]n]^2 \quad w_i^i + w_j^j - 2\, w_i^j \geqslant 0 \quad .$$

b) On peut donc poser :

$$\delta^2(i,j) = w_i^i + w_j^j - 2\, w_i^j \quad ;$$

on remarque que l'on a bien :

$$\delta(i,i) = 0 \qquad \forall i \in]n]$$

$$\delta(i,j) = \delta(j,i) \quad \forall (i,j) \in]n]^2 \text{ puisque } W \text{ est symétrique.}$$

Montrons que δ vérifie l'inégalité triangulaire :

$$\forall (i,j,k) \in]n]^3 \qquad \delta(i,j) \leqslant \delta(i,k) + \delta(k,j)$$

ou encore

$$\delta^2(i,j) \leqslant \delta^2(i,k) + \delta^2(k,j) + 2\, \delta(i,k)\, \delta(k,j)$$

Or

$$\delta^2(i,j) - \delta^2(i,k) - \delta^2(k,j) = w_i^i + w_j^j - 2w_i^j - w_i^i - w_k^k + 2w_i^k - w_k^k - w_j^j + 2w_k^j$$

$$= -\, 2w_k^k - 2w_i^j + 2w_i^k + 2w_k^j$$

L'inégalité de Schwarz, appliquée à la forme bilinéaire symétrique, semi-définie positive W , s'écrit :

$$\forall (x,y) \in V^2 \qquad |W(x,y)| \leqslant \sqrt{W(x,x)}\ \sqrt{W(y,y)}$$

Pour le choix de

$$x = \begin{pmatrix} x^1 \\ \vdots \\ x^\ell \\ \vdots \\ x^n \end{pmatrix} \quad \text{où} \quad x^i = 1,\ x^k = -1 \quad \text{et} \quad x^\ell = 0 \quad \text{si} \quad \ell \neq i \quad \text{et} \quad \ell \neq k$$

$$y = \begin{pmatrix} y^1 \\ \vdots \\ y^\ell \\ \vdots \\ y^n \end{pmatrix} \quad \text{où} \quad y^j = 1,\ y^k = -1 \quad \text{et} \quad y^\ell = 0 \quad \text{si} \quad \ell \neq j \quad \text{et} \quad \ell \neq k$$

On obtient :

$$W(x,y) = y^T W x = w_i^j - w_k^j - w_i^k + w_k^k$$

$$= -\frac{1}{2}\left[\delta^2(i,j) - \delta^2(i,k) - \delta^2(k,j)\right]$$

$$W(x,x) = w_i^i + w_k^k - 2\, w_i^k = \delta^2(i,k)$$

$$W(y,y) = w_j^j + w_k^k - 2\, w_j^k = \delta^2(k,j)$$

et l'inégalité

$$-W(x,y) \leqslant \sqrt{W(x,x)}\ \sqrt{W(y,y)}$$

s'écrit

$$\delta^2(i,j) - \delta^2(i,k) - \delta^2(k,j) \leqslant 2\,\delta(i,k)\,\delta(k,j)$$

ce qu'il fallait démontrer.

Remarque 1.- La matrice W du théorème (1.2) a pour terme général

$$w_i^j = \frac{1}{2} (\delta_{i\cdot}^2 + \delta_{j\cdot}^2 - \delta^2(i,j) - \delta_{\cdot\cdot}^2) \qquad (2)$$

avec $$\delta_{i\cdot}^2 = \sum_{k=1}^{n} p_k \, \delta^2(i,k)$$

et $$\delta_{\cdot\cdot}^2 = \sum_{i=1}^{n} p_i \, \delta_{i\cdot}^2$$

Si W est semi-définie positive, les dissimilarités δ définies dans la proposition et dans le théorème coïncident, on a en effet

$$w_i^i + w_j^j - 2 \, w_i^j = \delta^2(i,j) \; .$$

Notons que si la matrice W est semi-définie positive pour un système de poids D_n (chaque individu a pour poids p_i), alors W l'est aussi pour tout autre système de poids.

Remarque 2.- $\mathbb{1}$ est vecteur propre de WD_n associé à la valeur propre 0.

Remarque 3.- La proposition précédente montre que la forme bilinéaire symétrique définie par le théorème de [12] ne peut être semi-définie positive que si δ est un écart. Cette condition n'est pas suffisante ; l'exemple suivant de [2] montre que la matrice W associée à la distance δ définie sur l'ensemble $]4]$ par :

$$\Delta = \begin{pmatrix} 0 & 2 & 5 & 3 \\ 2 & 0 & 3 & 5 \\ 5 & 3 & 0 & 4 \\ 3 & 5 & 4 & 0 \end{pmatrix}$$

n'est pas semi-définie positive.

1.3. - *Etude du cas où* (E,δ) *n'a pas de représentation euclidienne.*

On essaie dans ce cas d'obtenir une représentation euclidienne "approximative" c'est-à-dire une image de l'ensemble E dans un espace euclidien (de faible dimension) telle que les distorsions entre les dissimilarités $\delta(i,j)$ des éléments i et j de E et les distances (notées $d(i,j)$) de leurs images soient les plus faibles possibles.

1.3.1. - *L'approche de Kruskal.*

Pour mesurer la qualité de cette approximation, Kruskal [7], [8], propose d'utiliser un indicateur qu'il appelle le "stress" : pour une configuration de points $x_1,\ldots,x_n$ dans un espace euclidien de dimension k $(\mathbb{R}^k)$ dont les distances entre points sont notées $d(i,j)$, il appelle stress la quantité S telle que :

$$S^2 = \frac{\sum_{i<j} (d(i,j) - \hat{d}(i,j))^2}{\sum_{i<j} d^2(i,j)}$$

où les distances interpoints $\hat{d}(i,j)$ sont celles qui rendent S^2 minimum en respectant la contrainte :

$$\delta(i,j) < \delta(i',j') \Longrightarrow \hat{d}(i,j) \leqslant \hat{d}(i',j') \ .$$

Considérant que chaque x_i de $\mathbb{R}^k$ a pour coordonnées $(x_i^1,\ldots,x_i^k)$, S est une fonction de $(x_1^1,\ldots,x_1^k,\ldots,x_i^1,\ldots,x_i^k,\ldots,x_n^1,\ldots,x_n^k)$ qu'il s'agit de minimiser. Pour cela il utilise la méthode des gradients ; celle-ci fournit un minimum généralement local et dépendant de la configuration de départ. Il y a donc lieu d'effectuer plusieurs fois l'algorithme en partant de configurations différentes et de comparer les valeurs du stress.

1.3.2. - L'approche de Cailliez et Pages [2].

A un indice de dissimilarité δ défini sur E, on peut associer, à l'aide des formules (2), une forme bilinéaire symétrique ${}^{\delta}W$ de terme général

$$ {}^{\delta}w_i^j = \frac{1}{2} \left[\delta_{i.}^2 + \delta_{j.}^2 - \delta^2(i,j) - \delta_{..}^2 \right] $$

Si δ et δ' sont deux indices de dissimilarité définis sur E, à l'indice λ défini par :

$$ \forall (i,j) \in E^2 \qquad \lambda^2(i,j) = \delta^2(i,j) + \delta'^2(i,j) $$

on peut associer la forme bilinéaire symétrique

$$ {}^{\lambda}W = {}^{\delta}W + {}^{\delta'}W \ . $$

Supposons que ${}^{\delta}W$ ne soit pas semi-définie positive et définissons l'indice δ' par

$$ \forall i \in E \qquad \delta'(i,i) = 0 $$

$$ \forall (i,j) \in E^2 , \quad i \neq j \qquad \delta'(i,j) = c \quad \text{où} \quad c \text{ est une constante} $$

strictement positive.

Faisant d'abord l'hypothèse : $p_i = \frac{1}{n}$ pour tout i ; alors la matrice ${}^{\delta'}W$ de la forme bilinéaire symétrique ${}^{\delta'}W$ s'écrit :

$$ {}^{\delta'}W = -\frac{1}{2n} c^2 \, \mathbb{1}\mathbb{1}^T + \frac{1}{2} c^2 \, I_n \ . $$

Comme d'après la remarque 2, page 26, $\mathbb{1}$ est vecteur propre de ${}^{\delta'}W \frac{I_n}{n}$ (donc de ${}^{\delta'}W$) associé à la valeur propre 0 , le sous-espace $\Delta_{\mathbb{1}}^{\perp}$, I_n - orthogonal à $\mathbb{1}$ est le sous-espace propre de ${}^{\delta'}W$ associé à la valeur propre $\frac{1}{2} c^2$.

Notons $\mu_1 \geqslant \mu_2 \ldots \geqslant \mu_n$ les valeurs propres de ${}^{\delta}W$; on a $\mu_n < 0$ puisque on a supposé que ${}^{\delta}W$ n'était pas semi-définie positive.

$\mathbb{1}$ vecteur propre de ${}^{\delta}W$ associé à la valeur propre 0 est aussi vecteur propre de ${}^{\lambda}W$ associé à la valeur propre 0.

Tout vecteur propre de ${}^{\delta}W$ de valeur propre 0 et situé dans $\Delta_{\mathbb{1}}^{\perp}$ est vecteur propre de ${}^{\lambda}W$ associé à la valeur propre $\frac{1}{2} c^2$.

Tout vecteur propre de ${}^{\delta}W$ de valeur propre $\mu_i \neq 0$ est vecteur propre de ${}^{\lambda}W$ de valeur propre $\mu_i + \frac{1}{2} c^2$. Il résulte de ceci, que la matrice ${}^{\lambda}W$ sera semi-définie positive si on choisit c de manière à ce que

$$\mu_n + \frac{1}{2} c^2 \geqslant 0$$

soit

$$c \geqslant \sqrt{2|\mu_n|} \quad .$$

D'après la remarque 1 du 1.2. on peut donc affirmer que ${}^{\lambda}W$ sera semi-définie positive, quelque soit le système de poids D_n affectés aux individus si $c \geqslant \sqrt{2|\mu_n|}$. On choisira $c = \sqrt{2|\mu_n|}$ afin de transformer au minimum la dissimilarité δ :

$$\forall (i,j) \in E^2 \qquad \lambda(i,j) = \delta(i,j) + \sqrt{2|\mu_n|} \quad \text{si} \quad i \neq j$$

$$\forall i \in E \qquad \lambda(i,i) = \delta(i,i) = 0 \ .$$

La matrice ${}^{\lambda}W$ s'écrit alors

$$ {}^{\lambda}W = {}^{\delta}W + |\mu_n| \ (I_n - \mathbb{1}\mathbb{1}^T D_n) \ (I_n - \mathbb{1}\mathbb{1}^T D_n)^T \ . $$

On utilisera alors une image euclidienne de (E,λ) comme image approximative de (E,δ).

Notons que dans le cas où $D_n = \frac{I_n}{n}$, si a est vecteur propre de ${}^{\delta}WD_n$ associé à la valeur propre $\lambda \neq 0$

$$ {}^{\delta}WD_n a = \lambda a \iff {}^{\lambda}WD_n a = (\lambda + \frac{|\mu_n|}{n}) \ a $$

La qualité globale de l'image dans le plan formé par les deux premiers axes factoriels (dans le cas où les deux plus grandes valeurs propres λ_1 et λ_2 sont positives) sera mesurée par :

$$ \frac{n(\lambda_1 + \lambda_2) + 2\ |\mu_n|}{n\left[\text{trace}({}^{\delta}WD_n) + (n-1)\ |\mu_n|\right]} $$

1.3.3. - Approche par l'analyse factorielle.

Soit R une relation binaire réflexive et symétrique sur $E \times E$ et soit K la matrice symétrique (n,n) dont le terme général est

$$ k_i^j = \begin{cases} 1 & \text{si} \quad i \ R \ j \\ 0 & \text{si} \quad i \not R \ j \end{cases} $$

Si k_i est la somme des éléments de la $i^{\text{ème}}$ colonne de K (k_i est le nombre de voisins de i) :

$$k_i = \text{card}\,\{j \in E : i\ R\ j\}$$

et k la somme des éléments de K, on note D_n la matrice diagonale de terme général $f_i = \frac{k_i}{k}$, F la matrice $\frac{K}{k}$, et on pose $X = F\,D_n^{-1}$.

A chaque individu i de E, associons la $i^{\text{ème}}$ colonne de la matrice X, c'est-à-dire le vecteur

$$x_i = \begin{pmatrix} x_i^1 \\ \vdots \\ x_i^j \\ \vdots \\ x_i^n \end{pmatrix} \in \mathbb{R}^n \qquad \text{où} \qquad x_i^j = \frac{k_i^j}{k_i} \quad , \quad j = 1,\ldots,n \ .$$

On considère alors le nuage pondéré des individus

$$N = \{(x_i, f_i),\ i = 1,\ldots,n\}$$

dont le centre de gravité G a sa $j^{\text{ième}}$ coordonnée proportionnelle au nombre de voisins de l'individu j.

En munissant $\mathbb{R}^n$ de la métrique D_n^{-1} du χ^2, la distance $d(i,i')$ entre deux individus i et i' (représentés par x_i et $x_{i'}$) s'écrit

$$d^2(i,i') = ||x_i - x_{i'}||^2_{D_n^{-1}} = \sum_{j=1}^{n} \frac{k}{k_j} \left(\frac{k_i^j}{k_i} - \frac{k_{i'}^j}{k_{i'}}\right)^2 .$$

On peut remarquer que si i et i' ont les mêmes voisins ($i\ R\ j \Longleftrightarrow i'\ R\ j$), leurs images x_i et $x_{i'}$ sont confondues ; si i et i' ne sont pas voisins mais ont les mêmes voisins "propres" (pour $j \neq i$ et $j \neq i'$, $i\ R\ j \Longleftrightarrow i'\ R\ j$) la distance de leurs images vaut $2k/(k_i)^3$: elle est fonction décroissante de k_i. Plus

généralement, la contribution de j (avec $i \mathcal{R} j$ ou $i' \mathcal{R} j$) à $d^2(i,i')$ sera plus forte si j a peu de voisins et sera maximale, dans le cas où $k_i \leqslant k_{i'}$, si $i \mathcal{R} j$ et $i' \not\mathcal{R} j$. Ces propriétés correspondent bien à la notion intuitive que nous avons des voisinages : on pourra localiser un individu d'autant plus facilement qu'il a beaucoup de voisins et que parmi ses voisins il y en a qui ont eux-mêmes peu de voisins.

Faisons une analyse factorielle du nuage N ; on a le schéma de dualité suivant

$$\begin{array}{ccc} \mathbb{R}^n & \xleftarrow{\quad X \quad} & (\mathbb{R}^n)^* \\ {\scriptstyle D_n^{-1}}\downarrow & & \uparrow{\scriptstyle D_n} \\ (\mathbb{R}^n)^* & \xrightarrow{\quad X^T \quad} & \mathbb{R}^n \end{array}$$

La recherche des axes factoriels se fait en diagonalisant la matrice

$$S = X\, D_n\, X^T\, D_n^{-1} \quad , \qquad \text{soit}$$

$$\begin{aligned} S &= F\, D_n^{-1}\, D_n\, D_n^{-1}\, F^T\, D_n^{-1} \\ &= F\, D_n^{-1}\, F\, D_n^{-1} \\ &= XX \ . \end{aligned}$$

On sait, d'après la théorie générale de l'analyse factorielle des correspondances, que S est diagonalisable et que ses valeurs propres appartiennent à $[0,1]$. Les vecteurs propres de S fournissent une base D_n^{-1} orthogonale de $\mathbb{R}^n$; en éliminant le vecteur propre trivial $\overrightarrow{OG}$ associé à la valeur propre 1, on repérera l'individu i par les coordonnées de x_i sur les autres vecteurs propres, ou ce qui revient à la même chose, sur les axes d'origine G et de direction ces vecteurs propres.

Si u_α est un vecteur propre D_n^{-1}-normé, associé à la valeur propre $\lambda_\alpha \neq 0$ de S, la coordonnée c_i^α de l'individu i sur l'axe Δ_{Gu_α} passant par G et de direction u_α, est la $i^{\text{ème}}$ composante du vecteur

$$c^\alpha = X^T D_n^{-1} u_\alpha .$$

Si on remplace le nuage N par le nuage formé par les projections de N sur les vecteurs propres u_α correspondant aux plus grandes valeurs propres, on obtiendra une image euclidienne approximative de N dont on pourra mesurer la qualité. Cette image pourra servir pour prendre en compte la contrainte associée à R.

On peut remarquer que

$$X^T D_n^{-1} X D_n c^\alpha = X^T D_n^{-1} \underbrace{X D_n X^T D_n^{-1}} u_\alpha$$
$$= \lambda_\alpha c^\alpha$$

et comme $X^T D_n^{-1} X D_n = X^T X^T$,

c^α est vecteur propre de $X^T X^T$ associé à la valeur propre λ_α ; on en déduit, par prémultiplication par X^T

$$X^T (X^T X^T) c^\alpha = (X^T X^T) X^T c^\alpha = \lambda_\alpha X^T c^\alpha$$

que $X^T c^\alpha$ est aussi vecteur propre de $X^T X^T$ associé à la valeur propre λ_α. Les $\frac{c^\alpha}{\sqrt{\lambda_\alpha}}$ forment une base D_n-orthonormée de $X^T[(\mathbb{R}^n)]^*$; on a aussi

$$(X^T c^\alpha)^T D_n (X^T c^\beta) = (c^\alpha)^T X D_n X^T c^\beta$$
$$= (c^\alpha)^T D_n D_n^{-1} F D_n^{-1} D_n D_n^{-1} F c^\beta$$
$$= (c^\alpha)^T D_n \lambda_\beta c^\beta = \lambda_\beta^2 \delta_{\alpha\beta}$$

où $\delta_{\alpha\beta}$ désigne le symbole de Kronecker.

2 - RECHERCHE DES ELEMENTS PROPRES D'UNE MATRICE SYMETRIQUE.-

Les différentes approches du problème proposées conduisent à la recherche des éléments propres d'une matrice symétrique. Nous présentons ici quelques méthodes permettant le calcul de tous ces éléments ou d'une partie d'entre eux : en général seuls les vecteurs propres associés aux plus grandes valeurs propres positives nous intéressent.

2.1. - Méthode de Jacobi.

La matrice A réelle symétrique de dimension (n,n) que l'on cherche à diagonaliser est transformée en une matrice diagonale par une suite de rotations planes. En fait, le nombre de rotations nécessaires peut être infini ; on arrête les itérations lorsque les éléments extradiagonaux sont négligeables. On pose $A_o = A$; on définit la suite de matrices A_k par :

$$A_k = R_k \, A_{k-1} \, R_k^T$$

où R_k est telle que, si l'élément extradiagonal de A_{k-1} de plus grand module occupe la position (p,q), alors, l'angle θ de la rotation est choisi de façon à mettre cet élément à 0. On a donc, si $p < q$:

$$\begin{cases} R_k(p,p) = \cos\theta = R_k(q,q) \\ R_k(p,q) = \sin\theta = -R_k(q,p) \\ R_k(i,i) = 1 \qquad i \neq p \quad \text{et} \quad i \neq q \\ R_k(i,j) = 0 \qquad \text{sinon .} \end{cases}$$

Les matrices A_k et A_{k-1} ne différent que par les lignes et les colonnes p et q.

La programmation est relativement lourde, car il faut à chaque itération rechercher l'élément extradiagonal de plus grand module. De plus, un élément mis à 0 par R_k peut être changé par R_{k+1}. Si on arrête les itérations à l'étape S, les valeurs propres de A_o sont sur la diagonale de la matrice $R_S R_{S-1} \dots R_1 A_o R_1^T \dots R_S^T$, et les vecteurs propres sont les colonnes de la matrice $P_S = R_1^T R_2^T \dots R_S^T$. Pour obtenir tous les vecteurs propres, il faut stocker à chaque itération la matrice $P_k = R_1^T \dots R_k^T$, ce qui nécessite un emplacement mémoire relativement important. On peut économiser l'espace mémoire et le temps en ne calculant que quelques vecteurs propres : à chaque itération, on conserve $\cos\theta$, $\sin\theta$ et la position (p,q). Pour obtenir le vecteur propre v_j associé à la $j^{ième}$ valeur propre, il suffit de calculer :

$$v_j = R_1^T \dots R_S^T e_j ,$$

en multipliant e_j successivement par R_S^T, $R_{S-1}^T \dots R_1^T$.

La méthode de Jacobi qui donne des résultats très fiables, n'est pas la plus rapide pour la recherche des éléments propres, mais elle présente l'avantage de fournir des vecteurs propres orthogonaux, même dans le cas où les valeurs propres sont proches l'une de l'autre.

2.2. - *Méthode de Householder.*

Cette méthode consiste à transformer la matrice A symétrique de dimension (n,n) en une matrice tridiagonale. Le calcul des éléments propres d'une telle matrice étant simple, il sera facile d'en déduire ceux de A.

On pose $A_o = A$, on définit la suite de matrices A_k par :

$$A_k = P_k \, A_{k-1} \, P_k$$

où P_k est une matrice orthogonale élémentaire, c'est-à-dire : $P_k = I - 2 \, w_k \, w_k^T$ avec $w_k \in \mathbb{R}^n$ et $||w_k|| = 1$. w_k est choisi de façon à introduire des 0 sur la $k^{ième}$ ligne et la $k^{ième}$ colonne sans détruire ceux introduits aux étapes précédentes.

Si $a_{i,j}$ désigne l'élément de A_{k-1} en position (i,j) on a :

$$P_k = I - 2 \, w_k \, w_k^T = I - u_k \, u_k^T / 2 \, K_k^2$$

avec

$$\begin{cases} u_k^i = 0 & i = 1,\ldots,k \\ u_k^{k+1} = a_{k+1,k} + (\text{signe de } a_{k+1,k}) \, S_k & \\ u_k^i = a_{k,i} & i = k+2 \, ,\ldots,n \\ 2 \, K_k^2 = S_k^2 + (\text{signe de } a_{k+1,k}) \, a_{k+1,k} \, S_k & \\ S_k^2 = \sum_{i=k+1}^{n} a_{k,i}^2 & \end{cases}$$

Si on tient compte de la symétrie, on peut diminuer le nombre d'opérations nécessaires pour le calcul de A_k :

$$A_k = (I - \frac{u_k\, u_k^T}{2\, K_k^2})\, A_{k-1}\, (I - \frac{u_k\, u_k^T}{2\, K_k^2})$$

$$= A_{k-1} - \frac{u_k\, u_k^T}{2\, K_k^2} A_{k-1} - A_{k-1} \frac{u_k\, u_k^T}{2\, K_k^2} + \frac{u_k\, u_k^T\, A_{k-1}\, u_k\, u_k^T}{4\, K_k^2}$$

En posant

$$p_k = \frac{A_{k-1}\, u_k}{2\, K_k^2}$$

et

$$q_k = p_k - \frac{1}{2}\, u_k\, (\frac{u_k^T\, p_k}{2\, K_k^2})$$

on a :

$$A_k = A_{k-1} - u_k\, q_k^T - q_k\, u_k^T \ .$$

Comme $A = A_o$ est symétrique, les matrices A_k le sont également et A_{n-2} est tridiagonale symétrique.

$$A_{n-2} = P_{n-2}\, P_{n-1} \cdots P_1\, A_o\, P_1 \cdots P_{n-2}$$

Les valeurs propres de A_{n-2} sont également celles de A_o. Si u est vecteur propre de A_{n-2} alors $P_1 \cdots P_{n-2}\, u$ est vecteur propre de A_o.

La recherche des éléments propres de A se ramène donc à celle des éléments propres d'une matrice tridiagonale symétrique.

Si T désigne une telle matrice, alors T peut se mettre sous la forme :

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \beta_n & \\ & & & \beta_n & \alpha_n \end{pmatrix}$$

Supposons que k des β_i soients nuls, la matrice T s'écrit alors sous la forme :

$$\begin{pmatrix} T_1 & & & \\ & T_2 & & \\ & & \ddots & \\ & & & T_{k+1} \end{pmatrix}$$

où, quelque soit ℓ, T_ℓ est une matrice tridiagonale dont tous les β sont nuls.

Si λ_ℓ est une valeur propre de T_ℓ, v_ℓ le vecteur propre associé, alors λ_ℓ est une valeur propre de T associé au vecteur propre $\begin{pmatrix} 0 \\ \vdots \\ 0 \\ v_\ell \\ 0 \\ \vdots \\ 0 \end{pmatrix}$. Pour avoir les éléments propres de T, il suffit donc d'étudier chaque sous matrice T_ℓ. On peut,

par conséquent, supposer que tous les β_i sont non nuls.

Le calcul des valeurs propres de T est basé sur la détermination des zéros du polynôme caractéristique $P_n(\lambda) = \det(T - \lambda I)$.

Celui-ci peut être calculé par récurrence :

$$P_o(\lambda) = 1$$

$$P_1(\lambda) = \alpha_1 - \lambda$$

- - - - - -

$$P_n(\lambda) = (\alpha_n - \lambda)\, P_{n-1}(\lambda) - \beta_n^2\, P_{n-2}(\lambda) \ .$$

L'utilisation des propriétés de cette suite de polynômes permet le calcul des valeurs propres.

Pour obtenir le vecteur propre x associé à la valeur propre λ, il suffit de résoudre le système $Tx = \lambda x$.

2.3. - *Méthode de Hotelling*.

Dans le cas où l'on ne désire extraire que les valeurs propres les plus grandes, et les vecteurs propres correspondants, on peut utiliser la méthode de Hotelling.

Si $\lambda_1 , \lambda_2 , \ldots, \lambda_n$ ($|\lambda_1| \geq |\lambda_2| \ldots \geq |\lambda_n|$) désignent les valeurs propres de A , $x_1 , x_2, \ldots, x_n$ les vecteur propres correspondants, et u_o un vecteur quelconque, la méthode consiste à construire les deux suites (v_k) et (u_k) définies par :

$$v_1 = Au_o \qquad u_1 = \frac{v_1}{\max(v_1)}$$

- - - - - - - - - - - - - - - -

$$v_{k+1} = Au_k \qquad u_{k+1} = \frac{v_{k+1}}{\max(v_{k+1})}$$

où, si $x = \begin{pmatrix} x^1 \\ \vdots \\ x^n \end{pmatrix}$ $\max(x) = \max_{i=1,\ldots,n} |x^i|$.

On a

$$u_k = \frac{A^k u_o}{\max(A^k u_o)}$$

On peut écrire : $u_o = \sum_{i=1}^{n} \alpha_i x_i$, et en supposant $|\lambda_1| > |\lambda_2|$ (le cas $|\lambda_1| = |\lambda_2|$ ne nécessite que peu de transformations dans la démonstration)

$$A^k u_o = \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i = \lambda_1^k \left[\alpha_1 x_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i x_i\right]$$

or quand k tend vers $+\infty$, $\left(\frac{\lambda_i}{\lambda_1}\right)^k$ tend vers 0 , $\forall i \neq 1$,

car $|\lambda_i| < |\lambda_1|$; par conséquent quand k tend vers $+\infty$,

$\frac{A^k u_o}{\max(A^k u_o)}$ tend vers $\frac{x_1}{\max(x_1)}$.

D'autre part, on a :

$$v_k = A\, u_k = \frac{A^{k+1} u_o}{\max(A^k u_o)}$$

d'où, quand k tend vers $+\infty$, $\max(v_k)$ tend vers λ_1 .

A partir d'un u_o quelconque, on peut donc obtenir u_1 et λ_1 ; pour le calcul des autres valeurs propres, il suffit de poser

$$A_1 = A - \lambda_1 x_1 x_1^T$$

et de remarquer que A et A_1 ont même éléments propres, sauf x_1 qui est associé à λ_1 pour A, à 0 pour A_1.

En utilisant la méthode ci-dessus pour A_1, on obtient donc λ_2 et x_2. On peut calculer les r $(r = 1,\ldots,n)$ valeurs propres $\lambda_1,\ldots,\lambda_r$ de A, $(|\lambda_1| > \ldots > |\lambda_r|)$ en recherchant successivement la plus grande valeur propre de A, $A_1,\ldots,A_{r-1}$. On en déduit donc le calcul des plus grandes valeurs propres de A.

3 - APPLICATIONS.-

3.1. - *Utilisation du théorème de Mardia* (page 22).

On reprend l'exemple des 7 régions (page 21) et des distances δ associées. Les valeurs propres de la matrice W sont : $\frac{7}{2}$ (double), $\frac{1}{2}$ (double), 0, $-\frac{1}{7}$, -1. La recherche des deux premiers vecteurs propres de la matrice W permet de visualiser les régions dans un plan ; on sait en effet que la $i^{ème}$ ligne de la matrice (7,2) construite à l'aide de ces deux vecteurs propres donne les coordonnées de la région i dans un plan. Le résultat de cette visualisation (figure page 44) montre que dans ce cas l'ensemble des 7 régions est bien représenté.

3.2. - Exemple de classifications sous contraintes.

L'exemple présenté est tiré de [1]. L'ensemble d'individus E est de cardinal 100. Chaque individu est décrit par deux variables x et y : y est la variable choisie pour classer les individus, x est la variable "de contiguïté".

3.2.1. - Construction des couples (x_i, y_i), $1 \leqslant i \leqslant 100$ *et des voisins.*

On tire 100 nombres suivant une loi uniforme sur $[0,300]$. Ces 100 nombres sont notés x_i.

Pour $1 \leqslant i \leqslant 100$:

$$z_i = x_i \, 1_{[0,150]}(x_i) + (300-x_i) \, 1_{]150,300]}(x_i)$$

$$r_i = 50.E\left[\frac{z_i}{50}\right] \quad \text{où} \quad E[\,] \quad \text{désigne "la partie entière de"} .$$

Soit $\varepsilon_1, \ldots, \varepsilon_{100}$ un échantillon d'une population de loi normale $N(0,1)$ alors

$$\forall i \, , \quad 1 \leqslant i \leqslant 100 \qquad y_i = r_i + 12 \, \varepsilon_i \, .$$

Chaque individu i est donc représenté par un couple (x_i, y_i).

Pour ε fixé (ici $\varepsilon = 15.3$), on détermine les voisins d'un individu i donné, au sens de la variable x, c'est-à-dire les individus j pour lesquels $|x_i - x_j| \leqslant \varepsilon$. Pour $\varepsilon = 15.3$, les individus ont au plus 13 voisins.

3.2.2. - Résultats.

Nous présentons tout d'abord le résultat d'une classification hiérarchique, chaque individu étant repéré uniquement par son ordonnée y. Comme on s'y attend, les classes réunissent des

individus dont l'ordonnée est voisine, quelle que soit l'abscisse de ceux-ci (résultats page 45).

Nous introduisons ensuite la contiguïté : pour chaque individu i, on connaît son ordonnée y_i et le nom de ses voisins.

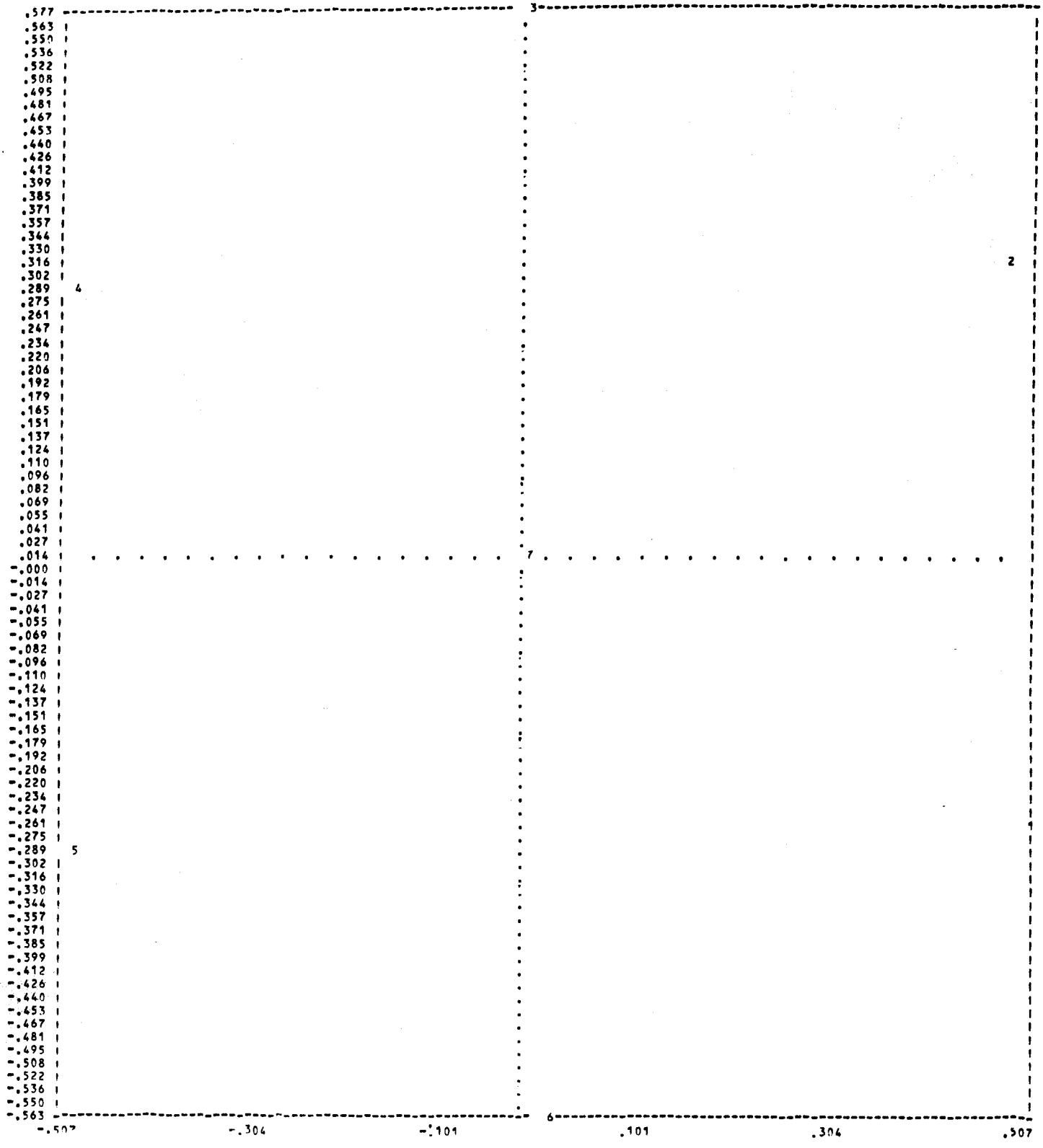
Soit Δ la matrice (n,n) définie par

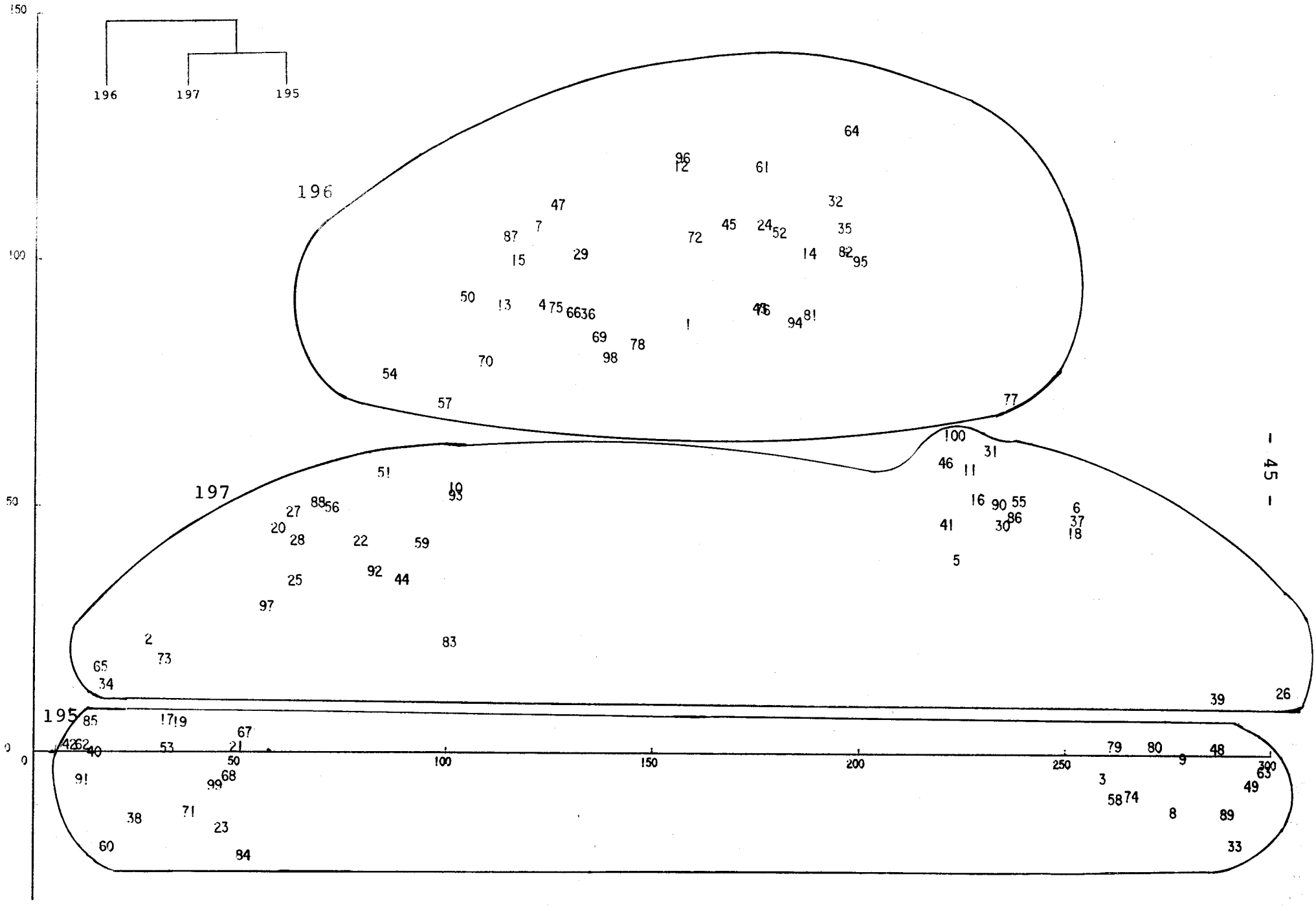
$$\Delta_i^j = \begin{cases} 1 & \text{si } i \text{ et } j \text{ sont voisins (au sens de } x) \\ 0 & \text{sinon .} \end{cases}$$

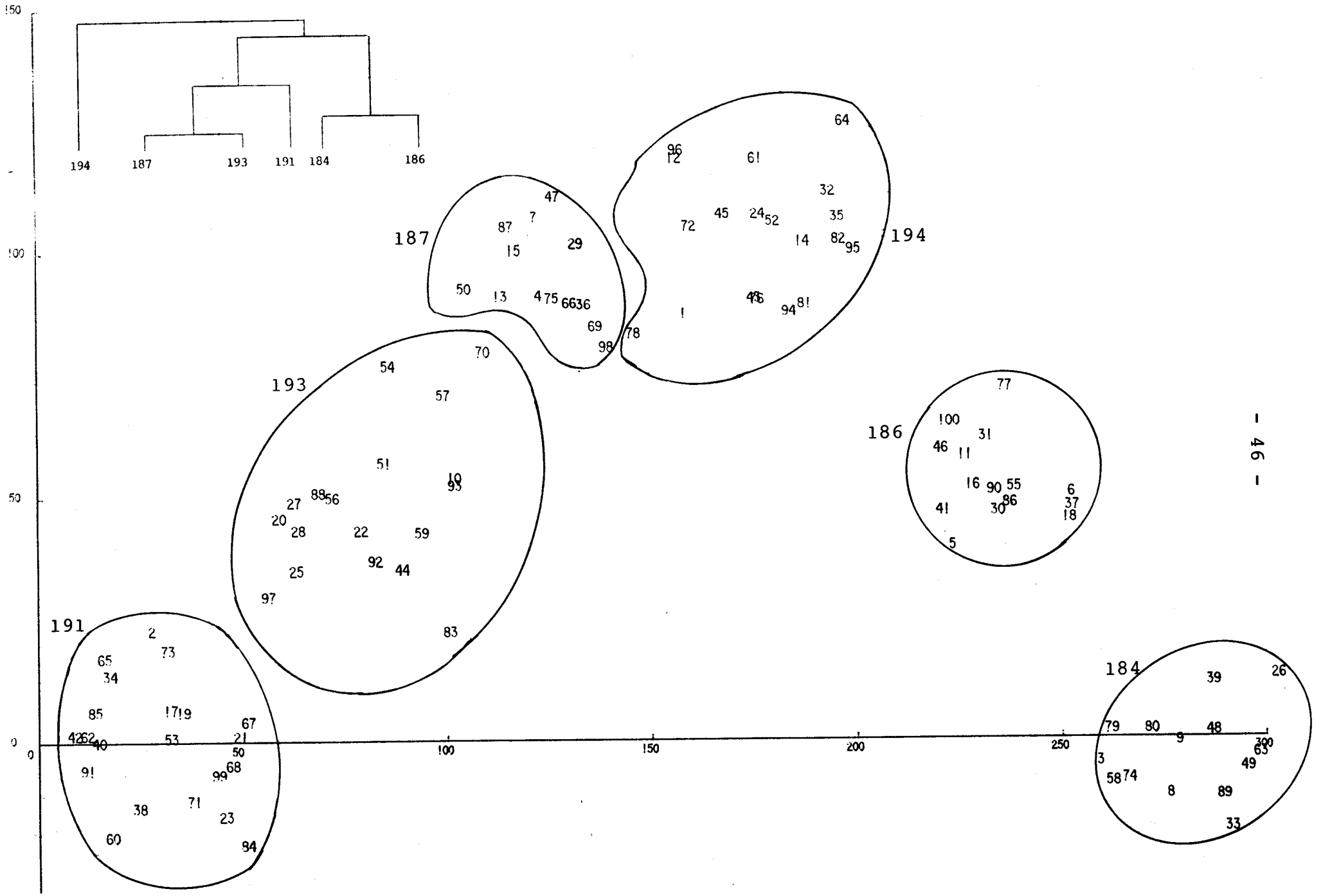
On effectue une A.F.C. sur Δ ; chaque individu i est repéré dans le plan principal par (c_i^1 , c_i^2). On traduira la contiguïté à l'aide des variables c^1 et c^2 .

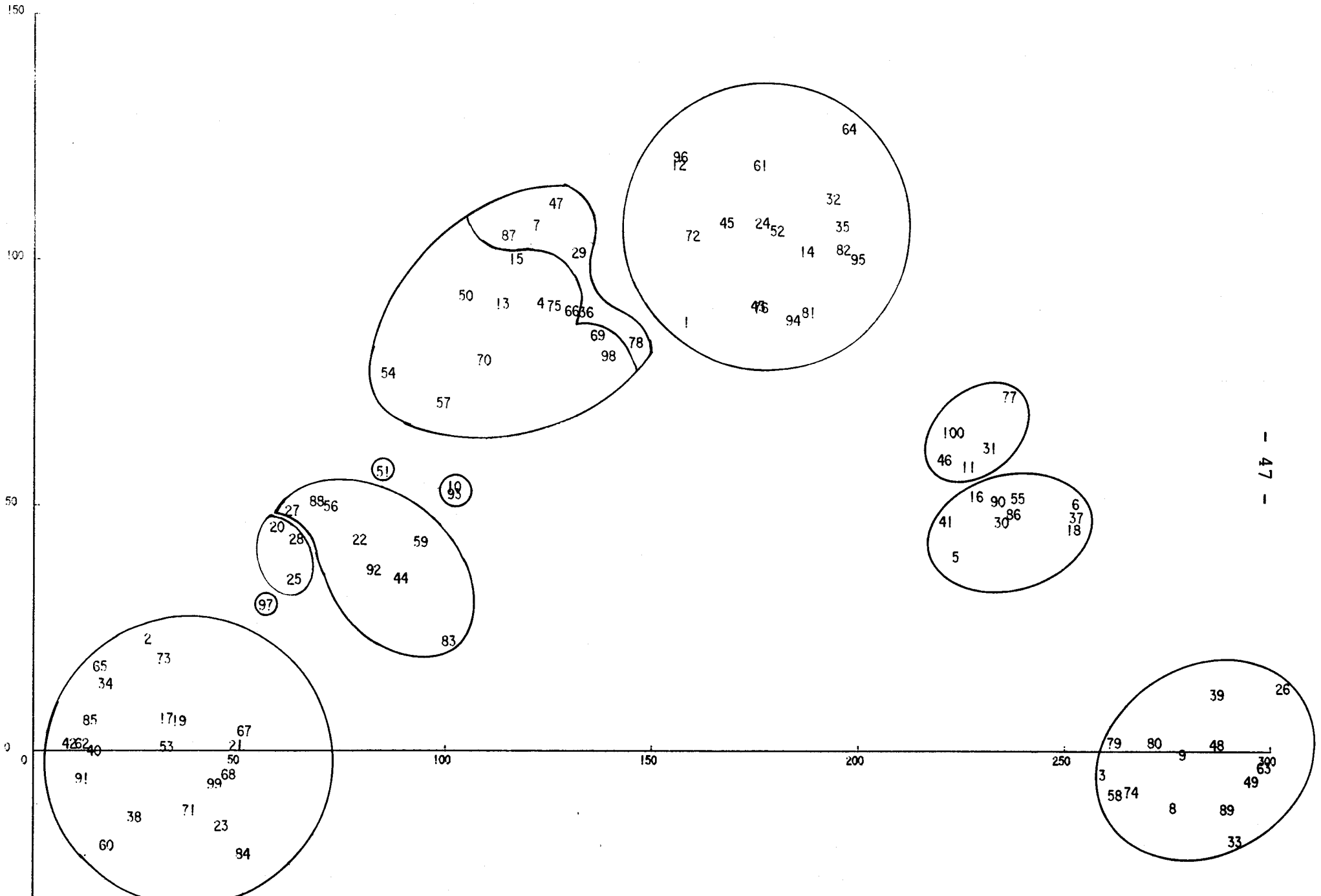
Une classification hiérarchique sur les individus repérés à l'aide des 3 variables préalablement réduites y , c^1 et c^2 fournit les résultats de la page 46 qui montrent la prise en compte effective de la contrainte.

Page 47, nous présentons les formes fortes obtenues sur les individus repérés par y , c^1 et c^2 par la méthode des Nuées Dynamiques ; on a appliqué cinq fois l'algorithme, les noyaux sont tirés au hasard, les partitions cherchées sont à 3 classes.


.577
.563
.550
.536
.522
.508
.495
.481
.467
.453
.440
.426
.412
.399
.385
.371
.357
.344
.330
.316
.302
.289
.275
.261
.247
.234
.220
.206
.192
.179
.165
.151
.137
.124
.110
.096
.082
.069
.055
.041
.027
.014
-.000
-.014
-.027
-.041
-.055
-.069
-.082
-.096
-.110
-.124
-.137
-.151
-.165
-.179
-.192
-.206
-.220
-.234
-.247
-.261
-.275
-.289
-.302
-.316
-.330
-.344
-.357
-.371
-.385
-.399
-.412
-.426
-.440
-.453
-.467
-.481
-.495
-.508
-.522
-.536
-.550
-.563
3
2
4
7
5
6
-.507
-.304
-.101
.101
.304
.507


196
197
195
150
100
50
0
0
50
100
150
200
250
300


194
187
193
191
184
186
150
100
50
0
0
100
150
200
250
300
187
194
193
186
191
184


150
100
50
0
0
50
100
150
200
250
300
64
96
12
61
32
45
24
52
35
72
14
82
95
47
7
87
29
15
50
13
4
75
66
36
69
78
98
70
54
57
1
94
81
77
100
31
46
11
16
90
55
6
41
30
86
37
18
5
51
10
93
27
88
56
20
28
22
59
92
44
25
97
83
2
73
65
34
85
17
19
67
42
62
40
53
21
91
99
68
38
71
23
60
94
39
26
79
80
9
48
3
58
74
8
89
49
63
33

CHAPITRE III

UNE METHODE MULTICRITERE DE CLASSIFICATION DU TYPE "NUEES DYNAMIQUES".

1 - INTRODUCTION.-

Lorsque les données relatives à chaque individu sont de types très divers, un problème de choix s'impose à l'analyste dès qu'il désire utiliser les méthodes de classification. En effet les algorithmes classiques ne peuvent être appliqués valablement que sur un ensemble de variables suffisamment homogènes.

Prenons l'exemple simple qui sera étudié par la suite dans lequel les individus sont des départements et où les variables descriptives sont :

- le profil de vote du département aux dernières élections présidentielles ;

- les coordonnées géographiques des préfectures ;

il est clair qu'une classification des départements effectuée sur ces données doit prendre en compte la proximité des départements au point de vue des élections - où une métrique du type χ^2 s'impose - et la proximité géographique des départements faisant intervenir la métrique euclidienne habituelle. Dans ces conditions, les algorithmes classiques ne permettent pas d'effectuer une telle classification. Dans le cas où différents groupes de variables apparaissent dans les données, l'analyste travaillera donc

généralement successivement avec chacun d'eux, obtenant ainsi autant de partitions ou de hiérarchies que de groupes. Le problème réside alors en la comparaison des différentes hiérarchies ou partitions de façon à présenter un "résumé" le plus fidèle possible des différents résultats.

La méthode que nous proposons, permet d'introduire différents groupes de variables dans un algorithme basé sur celui des Nuées Dynamiques. Ces groupes de variables ne doivent pas forcément définir la même mesure de proximité entre individus (ou plus généralement entre individuset noyaux), car nous n'utilisons pas directement la valeur des mesures mais pour chaque individu les rangs obtenus par les différents noyaux dans un classement ordonné croissant de ces mesures.

Prenons par exemple 5 individus a, b, c, d, e et quatre noyaux λ_1 , λ_2 , λ_3 , λ_4 . Supposons que les mesures de proximité soient données par la matrice D :

D	λ_1	λ_2	λ_3	λ_4
a	2,2	1,4	0,8	2,5
b	1,3	1,5	2	1,5
c	0,2	1,9	1	1,2
d	2,1	0,3	1,8	0,3
e	0,9	1,5	0,5	1,2

Dans la méthode proposée, la matrice D sera transformée en la matrice R , matrice de rangs. Pour construire une ligne de cette matrice, on range par ordre croissant les valeurs qu'elle contient. Une valeur de D est alors remplacée par le rang qu'elle possède dans ce classement.

R	λ_1	λ_2	λ_3	λ_4
a	3	2	1	4
b	1	2	4	2
c	1	4	2	3
d	4	1	3	1
e	2	4	1	3

L'utilisation des rangs se justifie en général pour plusieurs raisons :

1) Les données de base peuvent être elles-mêmes des classements et dans ce cas ce type d'analyse s'impose naturellement.

2) Les échelles de mesure peuvent être si différentes que même la réduction des variables reste insuffisante, cette opération ne remédiant pas par exemple à la dissymétrie des lois de ces variables. Il semble d'ailleurs plus justifiable de synthétiser une famille de classements qu'un ensemble très hétérogène de mesures.

3) Les résultats fournis sont robustes, très peu sensibles à l'existence de valeurs aberrantes.

Mais la raison de notre choix est que, travaillant sur les rangs, nous allons maintenant pouvoir prendre en compte de manière simultanée l'ensemble des groupes homogènes de variables mesurées sur les individus sans préoccupation du type de mesure de proximité associé à chacun de ces groupes. De plus, les variables pourront être quantitatives ou qualitatives à modalités ordonnées.

2 - L'ALGORITHME DE CLASSIFICATION.-

2.1. - Notations.

Nous reprenons les notations du I.2.1., nous bornant à en préciser certaines.

L'ensemble E à classifier (de cardinal n) et l'ensemble $\mathbb{L}$ des noyaux (de cardinal K) sont supposés finis.

Si N est le nombre de groupes de variables décrivant les éléments de E, D_1, D_2 ... et D_N sont les mesures de proximité entre un individu et un noyau associées à chacun de ces groupes :

$$\forall j \in]N] \qquad D_j : E \times \mathbb{L} \longrightarrow \mathbb{R}_+$$

L'application D de $E \times \mathbb{L}$ dans $\mathbb{R}_+$ utilisée dans l'algorithme classique des Nuées Dynamiques pour mesurer la proximité entre un élément de E et un noyau de $\mathbb{L}$ est alors définie par :

$$D(x,\lambda) = \sum_{j \in]N]} r_j(x,\lambda) + u(\lambda)$$

où $r_j(x,\lambda)$ est le rang de $D_j(x,\lambda)$ dans le classement par ordre croissant des $D_j(x,\mu)$ pour $\mu \in \mathbb{L}$ et u une application définie sur $\mathbb{L}$ à valeurs dans $\mathbb{R}_+$, qui sera précisée par la suite.

L'application R de $\mathbb{L} \times]k] \times \mathbb{P}_k$ mesurant l'ajustement d'un noyau λ à un élément P_i de la partition P est donnée par :

$$R(\lambda,i,P) = \sum_{x \in P_i} D(x,\lambda) \qquad \text{où} \quad P = (P_1,\ldots,P_k)$$

Le critère W à minimiser est alors défini par :

$$W(L,P) = \sum_{i \in]k]} R(\lambda_i,i,P) \quad \text{où} \quad L = (\lambda_1,\ldots,\lambda_k)$$

2.2. - *L'algorithme.*

2.2.1. - *Choix de* u.

Choisissant l'application u de manière que D(x,.) soit injective pour tout x de E, on peut associer au k-uple de noyaux $(\lambda_1,\ldots,\lambda_k) \in \mathbb{L}^k$ la partition $P = (P_1,\ldots,P_k)$ définie par :

$$\forall i \in]k] \qquad P_i = \{x \in E : D(x,\lambda_i) < D(x,\lambda_j),\ \forall j \in]k],\ j \neq i\} ;$$

on notera f cette application de $\mathbb{L}^k$ dans $\mathbb{P}_k$.

Choisissant u de manière que, quelle que soit la partition $\in \mathbb{P}_k$, il existe un k-uple de noyaux $(\lambda_1,\ldots,\lambda_k)$, unique, vérifiant :

$$\forall i \in]k] \qquad R(\lambda_i,i,P) = \min_{\lambda \in \mathbb{L}} R(\lambda,i,P) ;$$

nous notons g l'application de $\mathbb{P}_k$ dans $\mathbb{L}^k$ ainsi définie.

On réalisera les deux conditions, imposées ci-dessus à l'application u, de la manière suivante : on indice les éléments de $\mathbb{L}$ de 1 à K ; pour le noyau λ_ℓ, la valeur $u(\lambda_\ell)$ est prise égale à $\ell\ 10^{-m}$, où $m \in \mathbb{N}^*$ est choisi tel que $n\ K\ 10^{-m} < 1$. On ne modifie pas de cette façon la partie entière de $\sum_{j \in]N]} r_j(x,\lambda)$; de plus si pour $x \in E$, on avait pour λ_ℓ et pour $\lambda_{\ell'}$ éléments différents de $\mathbb{L}$, $\ell < \ell'$,

$$\sum_{j \in]N]} r_j(x,\lambda_\ell) = \sum_{j \in]N]} r_j(x,\lambda_{\ell'})$$

on a cependant

$$D(x,\lambda_\ell) < D(x,\lambda_{\ell'}) .$$

Notons enfin, qu'avec un tel choix de u, on a pour tout $A \in P(E)$

$$\sum_{x \in A} D(x,\lambda) = \sum_{x \in A} \sum_{j \in]N]} r_j(x,\lambda) + u(\lambda) \ .\ \text{card } A$$

et donc qu'il existe un $\lambda \in \mathbb{L}$ unique réalisant le minimum de $\sum_{x \in A} D(x,\lambda)$; remarquons que, s'il existe un $\lambda \in \mathbb{L}$ unique réalisant le minimum de $\sum_{x \in A} \sum_{j \in]N]} r_j(x,\ )$, c'est ce noyau qui est choisi ; s'il en existe plusieurs c'est celui d'indice minimum qui est sélectionné. L'application $g : \mathbb{P}_k \longrightarrow \mathbb{L}^k$ est donc bien définie.

2.2.2. - Déroulement de l'algorithme.

Soit $L^o = (\lambda_1^o, \ldots, \lambda_k^o)$ un k-uple de noyaux. Ceux-ci peuvent être tirés au hasard dans $\mathbb{L}$ ou être choisis par le praticien suivant sa connaissance de l'ensemble à classifier.

On définit la partition $P^o = f(L^o)$ par

$$\forall i \in]k] \quad P_i^o = \{x \in E : D(x,\lambda_i^o) < D(x,\lambda_j^o) \ , \ \forall j \in]k] \ \ j \neq i\}.$$

Soit $L^1 = (\lambda_1^1,\ldots,\lambda_k^1) = g(P^o)$ le k-uple de noyaux vérifiant :

$$\forall i \in]k] \qquad R(\lambda_i^1,i,P^o) = \min_{\lambda\in\mathbb{L}} R(\lambda,i,P^o) \ .$$

Soit $P^1 = f(L^1)$,...

Pour tout $n \in \mathbb{N}$, la partition $P^n = (P_1^n,\ldots,P_k^n)$ est construite à partir de $L^n = (\lambda_1^n,\ldots,\lambda_k^n)$ en utilisant f ; P^n est donc telle que :

$$\forall i \in]k] \quad P_i^n = \{x \in E : D(x,\lambda_i^n) < D(x,\lambda_j^n) \ \ \forall j \in]k] \ \ j \neq i\} \ .$$

A la partition P^n est associé $L^{n+1} = (\lambda_1^{n+1},\ldots,\lambda_k^{n+1})$ à l'aide de g, L^{n+1} étant défini par :

$$\forall i \in]k] \qquad R(\lambda_i^{n+1},i,P^n) = \min_{\lambda\in\mathbb{L}} R(\lambda,i,P^n)$$

$$\begin{array}{ccc} L^n & \longrightarrow & P^n \\ & & \downarrow \\ & & L^{n+1} \end{array}$$

2.3. - *Convergence de l'algorithme.*

Suivant la démarche de [6], nous allons montrer que la suite $(v_n)_{n\in\mathbb{N}} = ((L^n,P^n))_{n\in\mathbb{N}}$ converge vers $v^* = (L^*,P^*) \in \mathbb{L}^k \times \mathbb{P}_k$ en montrant d'abord que la suite

$(u_n)_{n\in\mathbb{N}} = (W(v_n))_{n\in\mathbb{N}}$ est une suite décroissante (donc convergente puisque à valeurs dans $\mathbb{R}_+$) et que sa limite est atteinte.

2.3.1.-

Pour $n \in \mathbb{N}$ intéressons nous à trois étapes consécutives de l'algorithme :

$$L^n = (\lambda_1^n,\ldots,\lambda_k^n) \xrightarrow{f} P^n = (P_1^n,\ldots,P_k^n)$$

$$\Big\downarrow g$$

$$L^{n+1} = (\lambda_1^{n+1},\ldots,\lambda_k^{n+1}) \xrightarrow{f} P^{n+1} = (P_1^{n+1},\ldots,P_k^{n+1})$$

Montrons que le critère W décroît à chaque étape, c'est-à-dire prouvons que :

$$W(L^n,P^n) \geqslant W(L^{n+1},P^n)$$

et $$W(L^{n+1},P^n) \geqslant W(L^{n+1},P^{n+1}) .$$

On a

$$W(L^n,P^n) = \sum_{i\in]k]} R(\lambda_i^n,i,P^n)$$

et $$W(L^{n+1},P^n) = \sum_{i\in]k]} R(\lambda_i^{n+1},i,P^n) .$$

Or par définition de L^{n+1} ,

$$\forall i \in]k] \qquad R(\lambda_i^{n+1},i,P^n) = \min_{\lambda\in\mathbb{L}} R(\lambda,i,P^n)$$

d'où, en particulier,

$$\forall i \in]k] \qquad R(\lambda_i^{n+1}, i, P^n) \leqslant R(\lambda_i^n, i, P^n)$$

par conséquent $\qquad W(L^{n+1}, P^n) \leqslant W(L^n, P^n)$

l'inégalité étant en fait une égalité si et seulement si $L^{n+1} = L^n$. Dans ce cas $P^{n+1} = P^n$ et la suite $(u_m)_{m \in \mathbb{N}}$ est constante à partir du rang n.

Si $L^{n+1} \neq L^n$, montrons que $W(L^{n+1}, P^{n+1}) < W(L^{n+1}, P^n)$:

$$W(L^{n+1}, P^n) = \sum_{i \in]k]} R(\lambda_i^{n+1}, i, P^n) = \sum_{i \in]k]} \sum_{x \in P_i^n} D(x, \lambda_i^{n+1})$$

$$= \sum_{x \in E} \sum_{i \in]k]} D(x, \lambda_i^{n+1}) \; 1_{P_i^n}(x)$$

$$W(L^{n+1}, P^{n+1}) = \sum_{i \in]k]} R(\lambda_i^{n+1}, i, P^{n+1}) = \sum_{x \in E} \sum_{i \in]k]} D(x, \lambda_i^{n+1}) \; 1_{P_i^{n+1}}(x)$$

P^n et P^{n+1} sont deux partitions ; il existe donc un unique indice h pour lequel $x \in P_h^{n+1}$, et un unique indice ℓ tel que $x \in P_\ell^n$.

La contribution de x à $W(L^{n+1}, P^n)$ est égale à $D(x, \lambda_\ell^{n+1})$, et sa contribution à $W(L^{n+1}, P^{n+1})$ vaut $D(x, \lambda_h^{n+1})$.

Or d'après la définition de P^{n+1} :

$$P_h^{n+1} = \{x \in E : D(x, \lambda_h^{n+1}) < D(x, \lambda_j^{n+1}) \quad \forall j \in]k] \;, \; j \neq h\}$$

$x \in P_h^{n+1}$ implique donc $D(x, \lambda_h^{n+1}) < D(x, \lambda_\ell^{n+1})$.

Il résulte que : $\qquad W(L^{n+1}, P^{n+1}) < W(L^{n+1}, P^n)$.

On a donc toujours $W(L^n,P^n) \geqslant W(L^{n+1},P^n) \geqslant W(L^{n+1},P^{n+1})$ et par conséquent, la suite $(W(L^n,P^n))_{n\in\mathbb{N}}$ est décroissante donc convergente puisqu'elle est minorée par O.

2.3.2. -

Pour montrer que l'algorithme converge, il faut encore prouver que la suite $((L^n,P^n))_{n\in\mathbb{N}}$ converge. E étant fini, $\mathbb{P}_k$ l'est également. Par conséquent, $g(\mathbb{P}_k)$ est fini.

Le nombre de couples (L^n,P^n) intervenant dans l'algorithme est donc fini. Comme, d'après 2.3.1., la suite $(W(L^n,P^n))$ converge, elle vérifie :

$$\exists M \in \mathbb{N} \quad \forall n \in \mathbb{N} \quad (n \geqslant M) \Longrightarrow (W(L^n,P^n) = u^*) \ .$$

En particulier :

$$\forall n \in \mathbb{N} \quad (n \geqslant M) \Longrightarrow (W(L^n,P^n) = W(L^{n+1},P^n) = W(L^{n+1},P^{n+1}))$$

$W(L^n,P^n) = W(L^{n+1},P^n)$ s'écrit encore :

$$\sum_{i\in]k]} R(\lambda_i^n,i,P^n) = \sum_{i\in]k]} R(\lambda_i^{n+1},i,P^n) \ .$$

Or, comme par définition de λ_i^{n+1} : $R(\lambda_i^{n+1},i,P^n) \leqslant R(\lambda_i^n,i,P^n)$ l'égalité précédente se traduit :

$$\forall i \in]k] \qquad R(\lambda_i^{n+1},i,P^n) = R(\lambda_i^n,i,P^n) \ .$$

L'hypothèse sur l'unicité du λ minimisant $R(.,i,P^n)$ permet d'affirmer que :

$$R(\lambda_i^{n+1},i,P^n) = R(\lambda_i^n,i,P^n) \Longleftrightarrow \lambda_i^{n+1} = \lambda_i^n$$

et ceci $\forall i \in]k]$.

Par conséquent $W(L^{n+1},P^n) = W(L^n,P^n) \Longleftrightarrow \forall i \in]k] \ \lambda_i^{n+1} = \lambda_i^n$

et $\forall n \in \mathbb{N} \quad (n \geqslant M) \Longrightarrow (L^{n+1} = L^n)$

d'où $P^{n+1} = P^n$ et $L^{n+2} = L^{n+1}$ ... .

En conclusion

$$\forall n \in \mathbb{N} \quad (n \geqslant M) \Longrightarrow ((L^n,P^n) = (L^*,P^*)) \ .$$

La convergence de la suite $((L^n,P^n))$ est donc assurée, ainsi que celle de l'algorithme.

2.4. - *Formes fortes.*

Le couple (L^*,P^*) obtenu par l'algorithme dépend généralement du L^o de départ. L'algorithme est donc appliqué plusieurs fois avec des L^o différents. La comparaison des partitions obtenues permet de mettre en évidence des groupes stables appelés formes fortes, éléments de l'ensemble quotient E/R pour la relation d'équivalence sur E définie par :

$x \ R \ y \Longleftrightarrow$ x et y sont classés ensemble dans toutes les partitions P^* obtenues avec différents L^o .

2.5. - *Pondération des critères.*

Une généralisation possible de la méthode consiste en l'introduction de pondération des différents critères. Ceci permet de donner plus ou moins d'importance à un critère, et donc de prendre en compte plus ou moins fortement dans la classification les $r_j(x,\lambda)$ pour certains j.

Si $q(j) \in \mathbb{N}$ est proportionnel au poids accordé au $j^{ième}$ critère, l'application D mesurant la proximité d'un individu x à un noyau λ s'écrira :

$$D(x,\lambda) = \sum_{j=1}^{N} q(j)\ r_j(x,\lambda) + u(\lambda) \ .$$

La démonstration proposée au 2.3. prouve encore la convergence de l'algorithme.

Remarques :

On pourra mesurer a posteriori l'importance d'un des critères en fixant les poids des autres critères et en étudiant l'évolution de la partition P^* obtenue ; une stabilité de P^* traduira le peu d'influence de ce critère dans la classification.

On peut aussi se poser la question de savoir quelles valeurs donner aux $q(j)$ pour que la partition P^* soit proche au sens d'une mesure de proximité entre partitions d'une partition P' donnée. La résolution de ce problème - encore ouvert - permettrait une approche nouvelle de l'analyse discriminante.

3 - *APPLICATIONS*.-

3.1. -

Pour les 96 départements de la France métropolitaine, on dispose des résultats au premier tour des élections présidentielles de 1981 pour les 10 candidats, ainsi que le nombre d'abstentions et le nombre de votes blancs ou nuls.

Chaque département est repéré géographiquement par les coordonnées (latitude, longitude) de sa préfecture.

Le but de la classification effectuée est de fournir des zones géographiques homogènes dans lesquelles sont agrégées des départements qui ont des profils de vote semblables.

Dans cet exemple, les ensembles E et $\mathbb{L}$ sont confondus et constitués des 96 départements. On calcule la distance du χ^2 pour tous les couples de départements ; un département x étant fixé, on affecte au département y le rang $r_1(x,y)$ de $d^2_{\chi^2}(x,y)$ dans la suite ordonnée croissante des valeurs de $d^2_{\chi^2}(x,z)$, pour $z \in E$.

De même on calcule $r_2(x,y)$, rang de $d^2(x,y)$ dans la suite ordonnée croissante des valeurs des distances euclidiennes classiques $d^2(x,z)$, pour $z \in E$.

On choisit deux entiers q_1 et q_2, proportionnels aux poids attribués au critère géographique et au critère électoral.

On définit alors l'application D sur $E \times E$ par

$$D(x,y) = q_1\, r_1(x,y) + q_2\, r_2(x,y) + u(y)$$

Si y apparaît en $\ell^{\text{ième}}$ place dans le fichier de lecture des départements, on a choisi ici :

$$u(y) = \ell\ 10^{-7} .$$

Le nombre de classes demandées pour chaque partition est 4. Pour chaque choix de (q_1,q_2), l'algorithme a été utilisé 5 fois avec des L^o tirés au sort. Pour des raisons de clareté, seules certaines formes fortes ont été reportées sur la carte de France.

- $q_1 = 0$, $q_2 = 1$.

La classification obtenue ne tient compte que du critère géographique : les formes fortes mettent clairement en évidence des régions géographiquement connexes.

- $q_1 = 1$, $q_2 = 0$.

La classification n'est faite que sur le critère électoral. Les formes fortes sont, en général, de cardinal moins important que celles obtenues dans le cas précédent et sont formées de départements généralement dispersés dans la carte de France.

- $q_1 = 1$, $q_2 = 1$.

L'introduction des deux critères simultanément, permet d'obtenir des groupes formés de départements voisins, qui ont des profils de vote proches. Notons que ces groupes ne sont pas, en général, l'intersection de deux formes fortes des cas $(q_1 = 1$, $q_2 = 0)$ et $(q_1 = 0$, $q_2 = 1)$.

- $q_1 = 3$, $q_2 = 4$.

Le poids accordé au critère géographique est un peu plus important que celui du critère électoral. Les formes fortes obtenues ont tendance à se rapprocher de celles obtenues pour $q_1 = 0$ et $q_2 = 1$. Notons cependant que le poids q_2 est encore trop faible par rapport au poids q_1 , pour permettre de regrouper les Côtes-du-Nord avec ses départements voisins, mais suffit pour regrouper le Haut-Rhin et le Bas-Rhin avec leurs voisins.

Remarque : Comme on le fait généralement, il est intéressant d'effectuer une AFC, sur le tableau des résultats électoraux dans les départements et de projeter en éléments supplémentaires les diverses formes fortes mises en évidence par la méthode de classification proposée

3.2. -

Nous reprenons l'exemple proposé page 42. Les ensembles E et $\mathbb{L}$ sont confondus et constitués de 100 points. Les critères utilisés correspondent aux deux variables x et y. L'algorithme a été utilisé 4 fois avec des $L^o \in E^3$ tirés au sort. On notera q_1 le poids accordé à x, q_2 celui de y. Les formes fortes obtenues pour des pondérations différentes sont reportées sur les schémas suivants :

- $q_1 = q_2 = 1$ (page 70)

La méthode met en évidence trois groupes très homogènes. Notons que les individus du groupe {10,51,93} ont été classés trois fois sur quatre avec ceux du groupe {56,88,59 ...}

- $q_1 = 1$, $q_2 = 2$ (page 71)

Les 3 formes fortes dont l'ordonnée moyenne est supérieure à 50 montrent clairement l'influence prépondérante de y. Cependant le poids q_2 accordé à y est encore trop faible devant celui de x pour permettre de regrouper pour chaque tirage des individus proches au sens de y, mais très éloignés au sens de x (voir les formes fortes dont l'ordonnée moyenne est petite). Si on s'intéresse en détail aux différentes partitions obtenues, on peut remarquer que l'individu 5 n'a, en fait, été classé qu'une seule fois sans les individus {30,41,86...}. De même le groupe {9,48,63,79,80} n'est séparé du groupe {3,8,26,33,39,49,58,74,89} que pour un seul tirage.

LATITUDE ET LONGITUDE DES PREFECTURES.

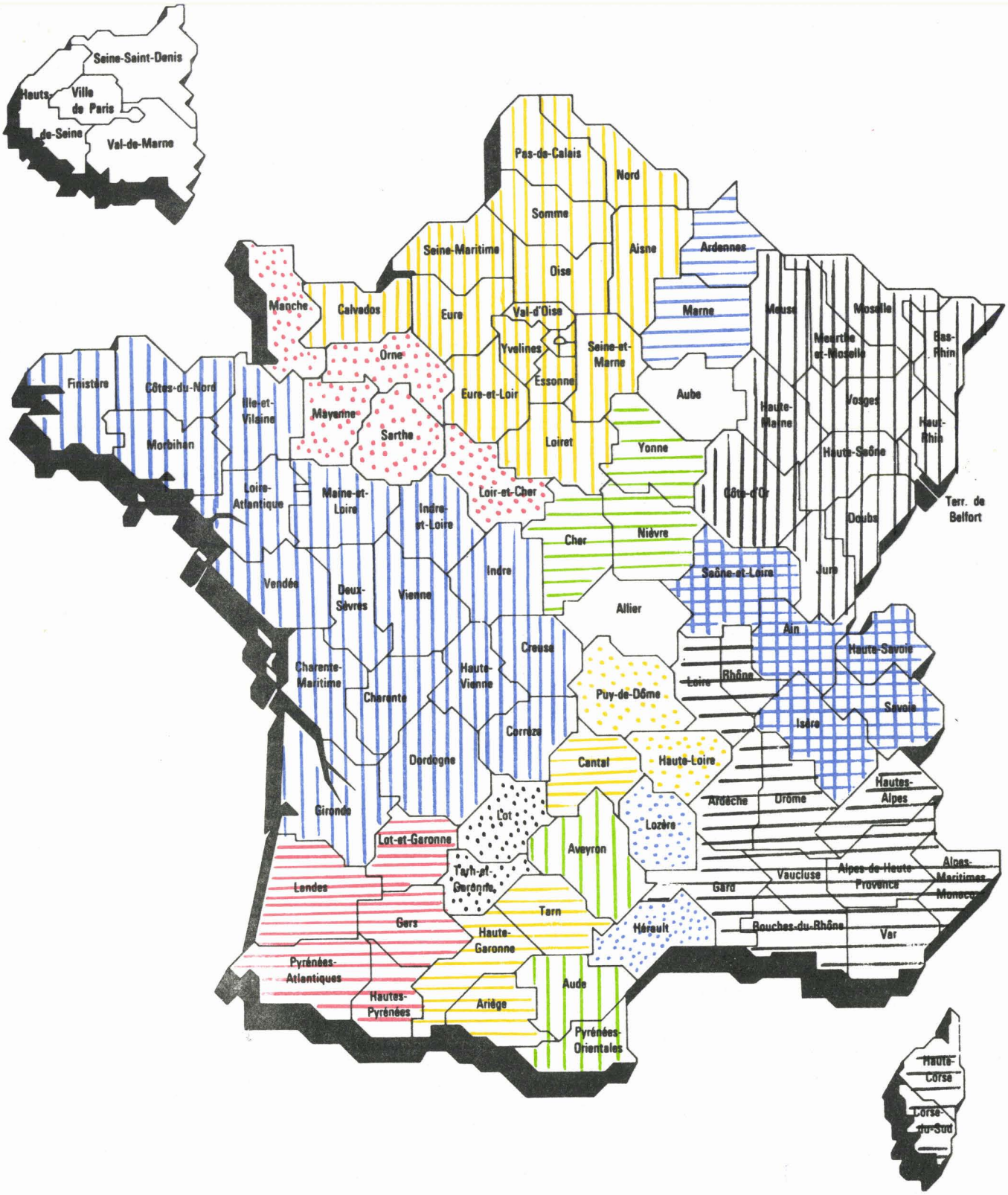
AIN	46.12	5.13
AISN	49.34	3.4
ALLI	46.34	3.2
ALHP	44.06	6.14
HTAL	44.34	6.05
ALMA	43.42	7.15
ARCH	44.44	4.36
ARDE	49.46	4.43
ARIE	42.58	1.36
AUBE	48.18	4.05
AUDE	43.13	2.21
AVEY	44.21	2.35
BORH	43.18	5.24
CALV	49.11	-00.21
CANT	44.56	2.26
CHAR	45.39	0.09
CHMA	46.10	-01.10
CHER	47.05	2.24
CORR	45.16	1.46
COSU	41.55	8.44
HTCO	42.42	9.27
CTOR	47.19	5.01
CTNO	48.31	-02.47
CREU	46.10	1.52
DORD	45.11	0.43
DOUB	47.15	6.02
DROM	44.56	4.54
EURE	49.01	1.09
EULO	48.27	1.3
FINI	48.	-04.06
GARD	43.5	4.21
HGAR	43.36	1.26
GERS	43.39	0.35
GIRO	44.5	- 0.34
HERA	43.36	3.53
ILVI	48.05	- 1.41
INDR	46.49	1.42
IETL	47.23	.41
ISER	45.1	5.43
JURA	46.4	5.33
LAND	43.53	- .30
LETC	47.35	1.2
LOIR	45.26	4.24
HTLE	45.02	3.53
LOAT	47.13	- 1.33
LRET	47.55	1.54
LOTT	44.27	1.26
LETG	44.12	0.37
LOZR	44.3	3.3
METL	47.28	-00.33
MANC	49.07	-01.05
MARN	48.57	4.22
HMAR	48.07	5.08
MAYE	48.04	-00.46
METM	48.41	6.12
MEUS	48.47	5.1
MORB	47.39	-02.46
MOSE	49.08	6.10
NIEV	47.	3.09
NORD	50.38	3.04
OISE	49.26	2.05
ORNE	48.26	0.05
PADC	50.17	2.47
PUY	45.47	3.05
PYRA	43.18	-00.22
HPYR	43.14	0.05
PYOR	42.41	2.53
BARH	48.35	7.45
HARH	48.05	7.22
RHON	45.45	4.51
HASA	47.38	6.1
SALO	46.18	4.5
SART	48.	0.12
SAVO	45.34	5.56
HTSA	45.54	6.07
PARI	48.52	2.20
SEMA	49.26	1.05
SETM	48.32	2.4
YVEL	48.48	2.08
DESE	46.19	-00.27
SOME	49.54	2.18
TARN	43.56	2.09
TAGA	44.01	1.21
VAR	43.07	5.56
VAUC	43.57	4.49
ESSO	48.38	2.27
HSEI	48.53	2.12
SESD	48.54	2.27
VDMA	48.48	2.28
VADO	49.03	2.06
VEND	46.40	-01.26
VIEN	46.35	0.20
HVIE	45.5	1.16
VOSG	48.11	6.27
YONN	47.48	3.34
TBEL	47.38	6.52

RESULTATS AUX ELECTIONS PRESIDENTIELLES DE 1981.

	VGF	CHIP	DERR	HFG	LALO	CREP	MITT	BOUC	MARC	LAGU	ABS	BNUL
ATH	64005.	36170.	3648.	3148.	8555.	5206.	51232.	2906.	22923.	4317.	69793.	3252.
AISN	77024.	49147.	5057.	3296.	9811.	4798.	76651.	2144.	65845.	8673.	56498.	5150.
ALLI	55467.	39294.	2506.	2259.	6804.	3894.	52072.	1332.	49934.	5077.	44351.	3646.
ALHP	18935.	10910.	990.	1070.	3043.	1294.	17495.	739.	13419.	1810.	16725.	1202.
AITAI	17338.	9482.	1050.	803.	2935.	1124.	14132.	786.	8948.	1508.	15817.	1100.
AIMA	149671.	94387.	5768.	6223.	17390.	7761.	98424.	3196.	75558.	6621.	123398.	6549.
ARCH	48577.	24250.	2547.	1629.	6111.	2327.	38828.	1992.	24329.	3383.	36699.	2538.
ARDF	40303.	25829.	3209.	1852.	5283.	2407.	42289.	1662.	30985.	4016.	33298.	2055.
ARIF	17625.	13094.	863.	933.	2723.	1742.	27469.	713.	17343.	2243.	20865.	1305.
AUBF	45665.	26225.	3141.	2378.	5709.	2915.	37441.	1141.	23727.	3539.	34050.	2569.
AUDF	32195.	27256.	1817.	1516.	5336.	3068.	57881.	1359.	34310.	3482.	33853.	2938.
AVEY	51618.	37425.	2534.	2071.	6532.	2985.	44109.	1939.	16427.	4694.	35719.	3446.
BORH	204318.	118662.	10008.	10206.	29206.	12823.	190955.	6477.	204744.	13312.	236702.	11520.
CALV	93118.	61229.	5042.	4329.	13477.	6858.	85255.	3428.	36413.	8435.	67478.	4662.
CANT	25924.	33455.	893.	935.	2142.	1136.	22108.	652.	10596.	2175.	23444.	1056.
CHAP	48486.	37138.	3032.	2478.	5782.	9671.	55734.	1268.	32025.	4573.	44622.	3313.
CHMA	76086.	47265.	4036.	3615.	8464.	34484.	70519.	1893.	32025.	5232.	74216.	4279.
CHEP	49693.	31467.	3305.	1997.	5866.	3736.	40999.	1356.	36392.	4817.	37269.	3192.
CORP	144661.	65311.	1163.	1001.	3087.	2062.	32362.	964.	34459.	2767.	24657.	1780.
COSN	17150.	16251.	451.	401.	1004.	629.	13660.	297.	9172.	347.	29760.	1174.
HTCO	17531.	19825.	460.	446.	1247.	7157.	12867.	404.	12206.	547.	42207.	726.
CTOP	63864.	45145.	4384.	3725.	9693.	5503.	72931.	3032.	24066.	5774.	58336.	3383.
CTUD	96309.	56509.	3847.	2973.	12717.	5047.	92739.	5011.	53724.	8641.	60843.	3442.
CREU	17421.	25317.	719.	854.	1921.	1244.	20826.	607.	18269.	2688.	22892.	1151.
DORD	51049.	52966.	3735.	2684.	7027.	5527.	63830.	2126.	50028.	5690.	44858.	4185.
DOUP	61215.	47025.	4626.	3664.	10765.	4235.	68517.	3845.	26731.	6090.	54243.	3751.
DROM	55028.	31319.	3787.	2798.	9691.	4191.	57146.	2869.	30399.	5043.	50830.	3664.
EURE	72865.	47722.	4004.	3351.	8999.	5799.	67674.	2244.	34374.	6969.	50126.	4116.
FINA	59859.	34104.	3522.	2810.	7473.	5097.	50906.	1827.	23439.	5656.	38962.	3433.
FINI	149050.	95083.	6012.	5024.	20793.	8530.	132338.	10055.	48561.	10936.	104066.	4951.
GARD	73593.	40419.	4390.	3264.	11235.	4991.	70173.	2881.	72916.	6259.	68240.	5651.
HGAR	96658.	65945.	5021.	5885.	17049.	10160.	136912.	5034.	63188.	9659.	100717.	6398.
GERS	24844.	18897.	1271.	1230.	3729.	2420.	36958.	1071.	14929.	2942.	23530.	1768.
GIRO	137166.	97366.	12854.	8388.	19543.	15783.	195169.	4755.	82611.	13142.	139023.	8719.
HERA	91952.	58165.	5547.	4217.	13679.	6767.	95871.	3981.	76020.	7075.	88813.	5788.
ILVT	132068.	80924.	6644.	5154.	17972.	7909.	103117.	6028.	29541.	11163.	85523.	5841.
INDR	39109.	28217.	2621.	1786.	4036.	2930.	34474.	1109.	28294.	4114.	30913.	3041.
IFTL	72901.	40229.	11410.	4172.	9709.	8654.	75217.	2701.	31497.	6653.	61469.	5278.
ISEP	112947.	66019.	7877.	5783.	21353.	9284.	124378.	6675.	72253.	10168.	116110.	7107.
JURA	38213.	22579.	3170.	2191.	6213.	2700.	36078.	2191.	18061.	3910.	31768.	3049.
LAND	47017.	30748.	2625.	1947.	4744.	3095.	62326.	1118.	26095.	3487.	33275.	2902.
LETC	53259.	25841.	4077.	2519.	5805.	4032.	43511.	1754.	24732.	4893.	33927.	3612.
LOIR	108556.	64020.	6688.	4730.	15179.	7094.	91612.	6247.	58742.	7873.	89741.	15112.
HTFI	45022.	22028.	1652.	1455.	4453.	1704.	30687.	1463.	10595.	2931.	28792.	1926.
LOAT	156173.	93688.	9783.	6730.	23077.	14811.	149168.	7999.	48923.	13395.	121243.	7898.
LPET	85294.	50054.	5640.	4554.	11977.	7524.	67198.	2665.	35117.	7560.	56437.	5803.
LOTT	18258.	22732.	1339.	1174.	3561.	3341.	30207.	996.	13337.	2599.	18133.	1463.
LFTG	43425.	30944.	2759.	2150.	6653.	4288.	48332.	1346.	30944.	4745.	35369.	3149.
LOZR	17032.	8907.	775.	513.	1570.	450.	10215.	575.	3900.	1136.	11193.	582.
METL	123998.	70413.	7696.	4551.	15298.	8245.	84082.	4153.	24657.	8584.	71643.	6716.

MANC	96885.	58048.	4120.	3090.	12737.	4516.	59453.	2367.	18833.	6129.	57245.	3779.
MARN	85637.	52035.	5104.	3611.	11388.	5648.	69379.	2785.	44475.	6597.	66613.	4479.
HMAR	33061.	20202.	2188.	1754.	4051.	2008.	32233.	1266.	15390.	3049.	27258.	2310.
MAYE	56966.	37213.	2538.	1936.	5964.	2572.	35883.	1659.	8284.	3996.	27910.	3136.
METM	108646.	50775.	6484.	5037.	13283.	6738.	98127.	4339.	65357.	9475.	90260.	5633.
MEUS	37570.	17834.	2007.	1531.	3974.	1671.	30668.	1086.	13821.	3144.	23415.	2083.
MORB	117064.	65254.	4588.	3552.	13644.	5645.	85749.	4249.	32806.	8148.	69038.	4046.
MOSE	172231.	82204.	7924.	7658.	19117.	6842.	132973.	5257.	59767.	12576.	117980.	9541.
NIEV	32145.	19388.	1749.	1471.	3674.	2277.	55914.	981.	21524.	3045.	32798.	2036.
NORD	366307.	195628.	23763.	13080.	46463.	18482.	346897.	9789.	287069.	31078.	236851.	26133.
OISE	88369.	59094.	5862.	4658.	12887.	7282.	86801.	3071.	61695.	10063.	70733.	5618.
ORNE	51827.	42138.	2710.	2122.	6343.	3158.	39794.	1723.	14598.	4521.	34047.	2662.
PADC	209576.	110499.	11433.	6671.	21241.	9584.	222109.	4789.	185548.	20002.	128656.	15514.
PUY	101027.	46148.	3109.	4074.	11940.	6094.	88820.	3372.	43495.	9194.	66083.	4294.
PYRA	90877.	65703.	4968.	3767.	11699.	5618.	90315.	3410.	33380.	6436.	70652.	3926.
HPYR	30511.	20612.	1771.	1589.	4075.	3173.	40716.	1192.	25005.	2816.	34484.	2096.
PYOR	45555.	26397.	2325.	2181.	6394.	3815.	43943.	1532.	35850.	3746.	50128.	3163.
BARH	210105.	69419.	7956.	7294.	21596.	7174.	101024.	4963.	20865.	7893.	112820.	9063.
HARH	125177.	58109.	6102.	5509.	16953.	5665.	74679.	3731.	18840.	7679.	82473.	8429.
RHON	190124.	116155.	11921.	10011.	31242.	15400.	172694.	11770.	87704.	13350.	188526.	9935.
HASA	36404.	23442.	1936.	1651.	4194.	3068.	37625.	1263.	15353.	3625.	28553.	2521.
SALO	88296.	49566.	4580.	3654.	10509.	6734.	85970.	2788.	46143.	6640.	79023.	5022.
SART	87256.	49001.	4401.	3019.	9832.	6129.	72453.	2941.	40094.	7751.	56865.	5361.
SAVO	46522.	31700.	3237.	2126.	8184.	3264.	41896.	2285.	22978.	3778.	46966.	2522.
HTSA	72619.	47500.	5794.	4362.	12238.	5136.	53879.	3245.	21354.	4845.	66878.	3795.
PARI	253390.	263204.	17596.	17327.	39713.	20728.	239974.	17529.	89613.	16896.	285439.	11121.
SEMA	180849.	91403.	9518.	6997.	24046.	12598.	171161.	6317.	123304.	17824.	128384.	10679.
SETM	111606.	82611.	7462.	6975.	20343.	11369.	108995.	4736.	66171.	10446.	99783.	6798.
YVEL	154481.	118604.	11251.	11559.	28840.	17241.	139543.	8136.	71739.	11870.	127962.	8608.
DESE	65809.	33635.	3396.	3676.	6588.	8429.	53503.	1834.	16142.	4963.	38308.	3991.
SOME	84006.	52271.	4909.	3253.	10264.	4774.	74883.	2442.	70993.	9334.	49501.	5511.
TARN	51875.	38212.	2867.	2718.	7903.	4840.	60954.	1991.	29719.	5364.	36400.	4837.
TAGA	25963.	22387.	1484.	1387.	4417.	6656.	31344.	1132.	15403.	2903.	21777.	2432.
VAR	117514.	64991.	4876.	5465.	13799.	6585.	85749.	2528.	67294.	5632.	90660.	5666.
VAUC	59495.	36102.	3248.	3270.	9306.	4202.	57430.	1996.	42264.	4719.	48691.	4878.
ESSO	110046.	87086.	8293.	8076.	25735.	13712.	128182.	7544.	80779.	11257.	106554.	7378.
HSET	168480.	139222.	12620.	12595.	32557.	17521.	158852.	10660.	109047.	13720.	168715.	9135.
SESO	112841.	89723.	7865.	7364.	25852.	13985.	141812.	7209.	158132.	14634.	154796.	11360.
VDMA	124784.	103773.	9128.	8961.	27028.	14445.	140970.	8290.	122326.	12306.	136234.	8197.
VADO	99926.	74186.	7067.	6909.	21802.	11920.	110608.	5690.	80651.	10172.	94914.	9497.
VEND	104500.	58981.	5258.	3827.	10026.	11330.	61117.	2831.	18985.	5897.	47813.	5135.
VIEN	58253.	39792.	3118.	3490.	7272.	7864.	56615.	2183.	28171.	4533.	43841.	4218.
HVIE	37370.	50251.	2103.	2062.	5844.	4145.	56052.	1616.	52547.	4522.	38605.	3946.
VOSG	65370.	39976.	4364.	3172.	8182.	3777.	59223.	2188.	25136.	7031.	47720.	5152.
YONN	52485.	30758.	3173.	2580.	6702.	3853.	43671.	1729.	24282.	4341.	39510.	3236.
TBEL	16126.	10616.	1088.	870.	2623.	1304.	21711.	894.	7718.	2035.	15072.	1268.

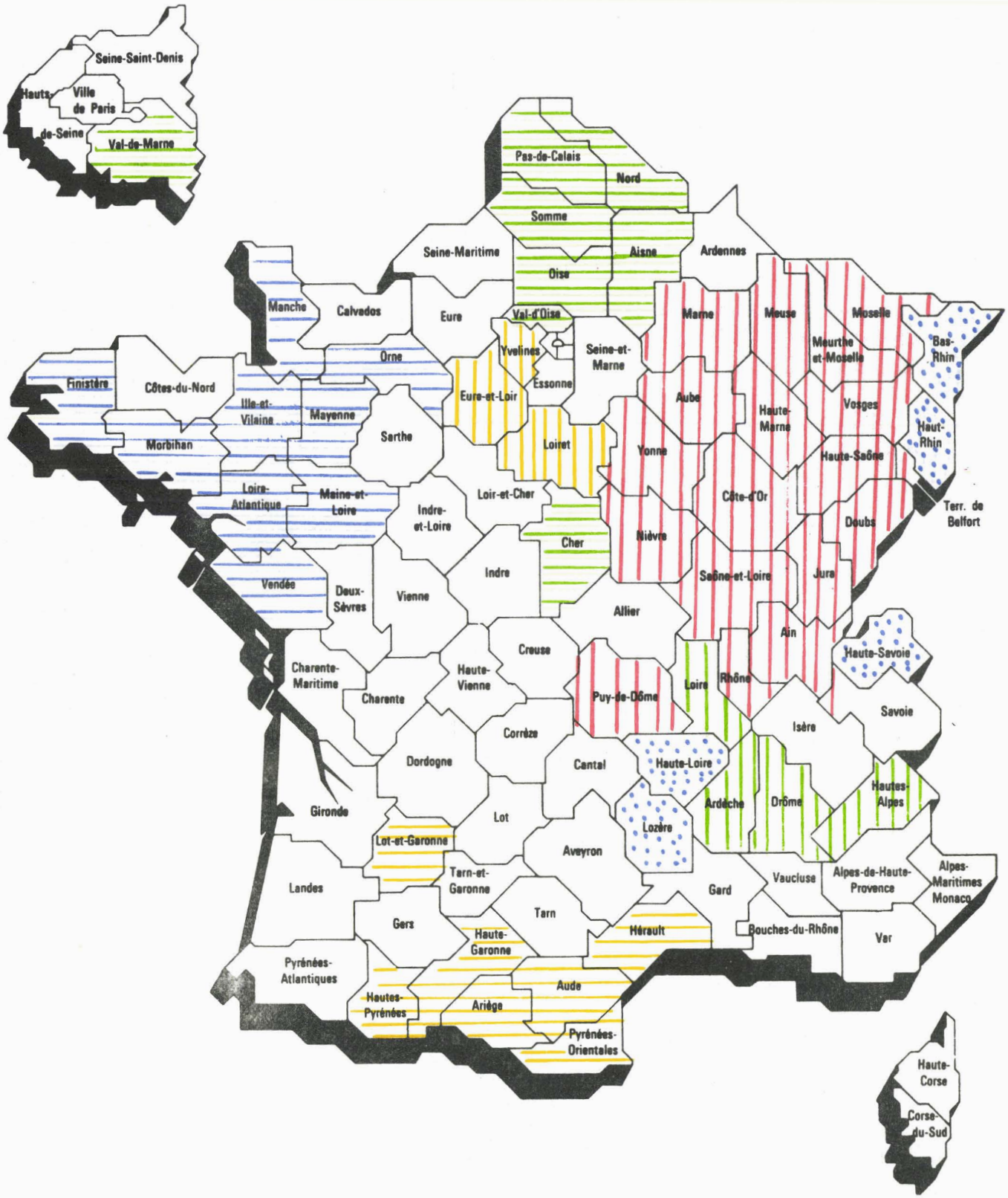
FORMES FORTES OBTENUES SUR LE CRITERE GEOGRAPHIQUE SEUL



REPRESENTATION DE QUELQUES FORMES FORTES AVEC LE CRITERE ELECTORAL SEUL

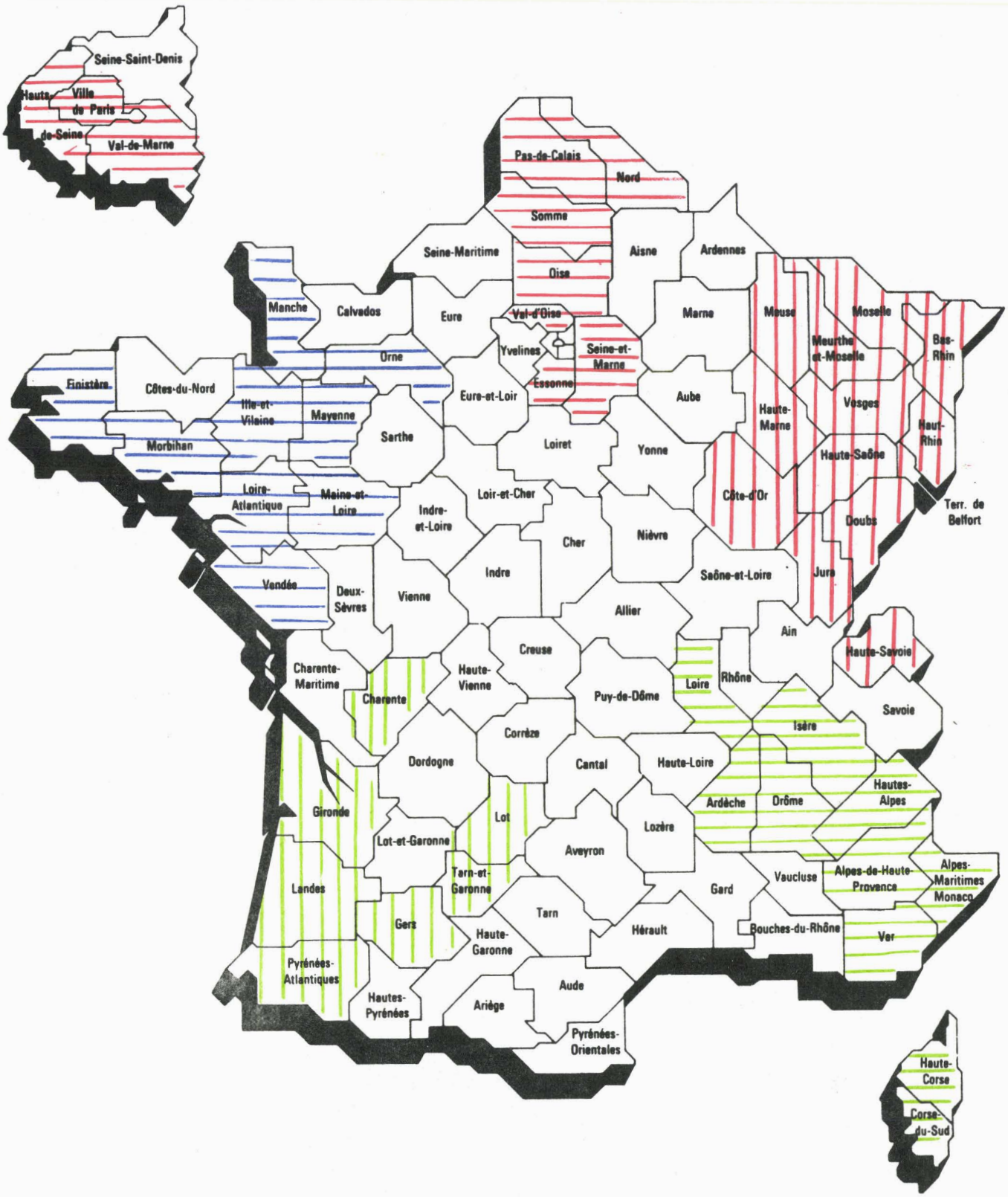
REPRESENTATION DE QUELQUES FORMES FORTES DANS LE CAS

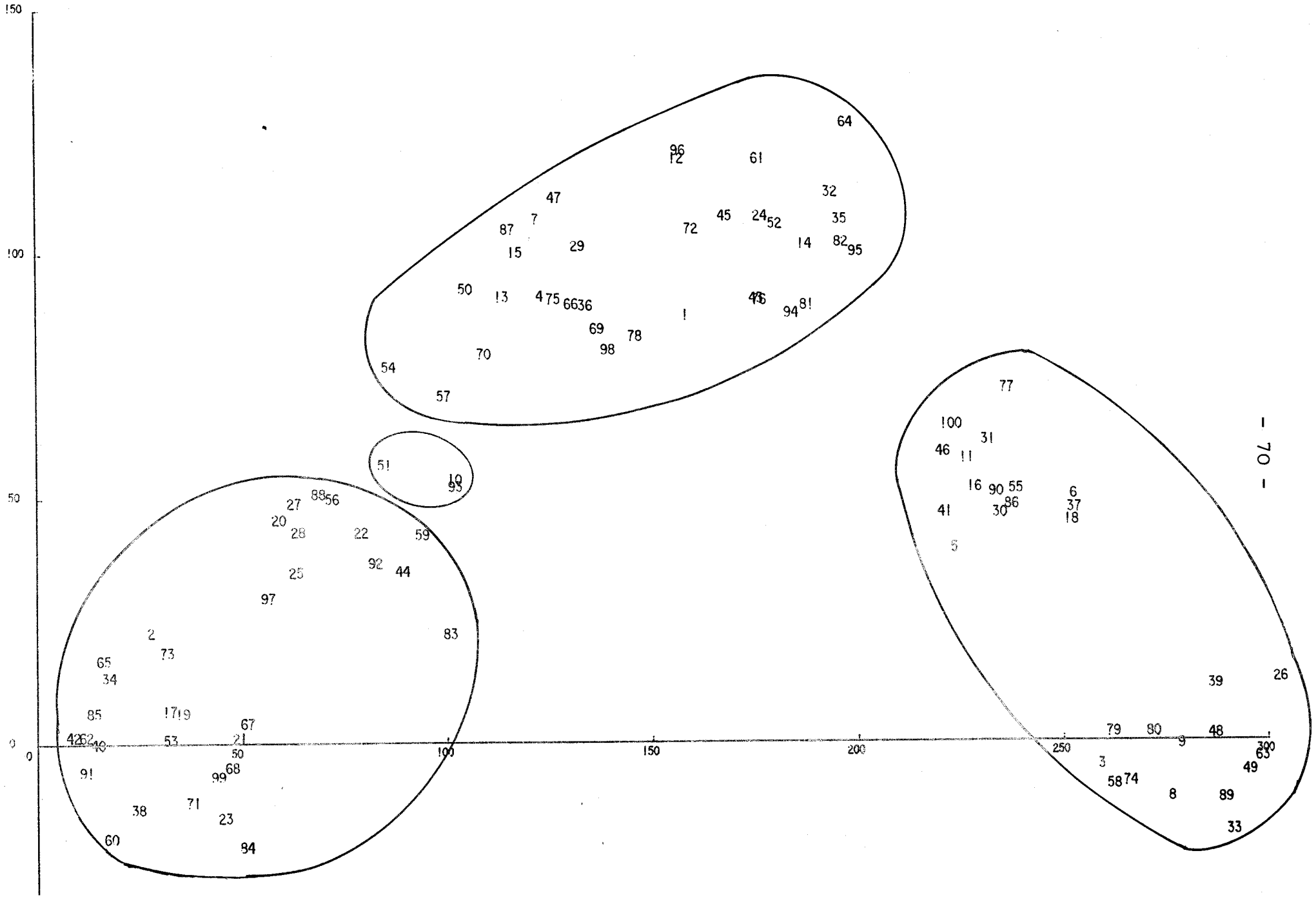
$q_1 = 1$, $q_2 = 1$.

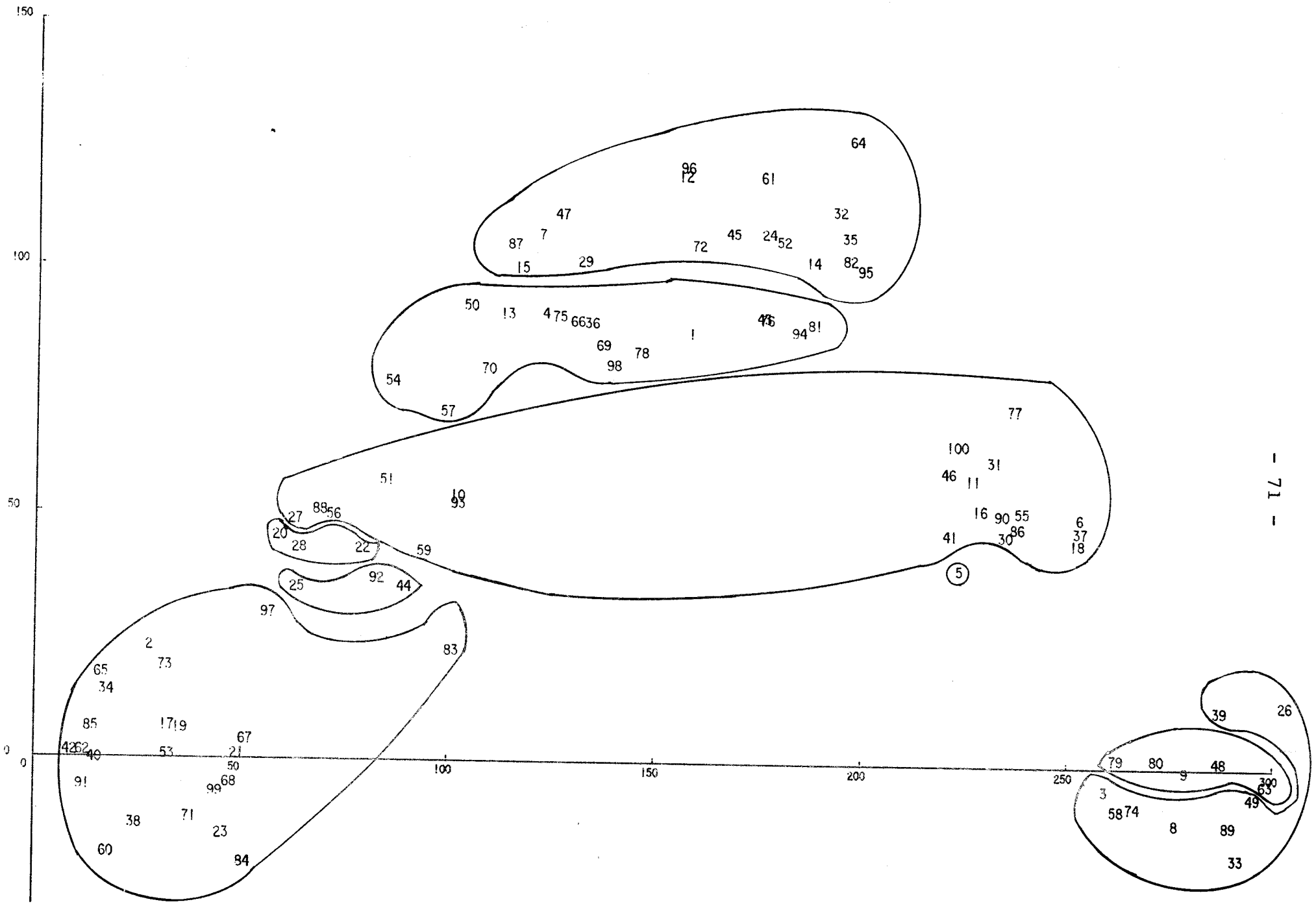


REPRESENTATION DE QUELQUES FORMES FORTES DANS LE CAS

$q_1 = 3$, $q_2 = 4$.




150
100
50
0
0
50
100
150
200
250
300
64
96
12
61
32
47
45
24
52
35
7
87
72
14
82
95
29
15
50
13
4
75
66
36
1
94
81
69
78
98
70
54
57
51
10
93
27
88
56
20
28
22
59
92
44
25
97
83
2
73
65
34
85
17
19
67
42
62
40
53
21
91
99
68
38
71
23
60
84
77
100
31
46
11
16
90
55
6
41
30
86
37
18
5
39
26
79
80
48
9
3
58
74
8
89
33
49
63


150
100
50
0
0
50
100
150
200
250
300
64
96
12
61
47
32
87
7
45
24
52
35
72
15
29
14
82
95
50
13
4
75
66
36
94
81
69
78
98
70
54
57
77
100
31
46
11
51
10
93
16
90
55
6
88
56
27
20
28
22
59
86
37
41
30
18
5
25
92
44
97
2
83
73
65
34
85
17
19
67
42
62
40
53
21
91
99
68
38
71
23
60
84
39
26
79
80
9
48
3
58
74
8
89
49
33

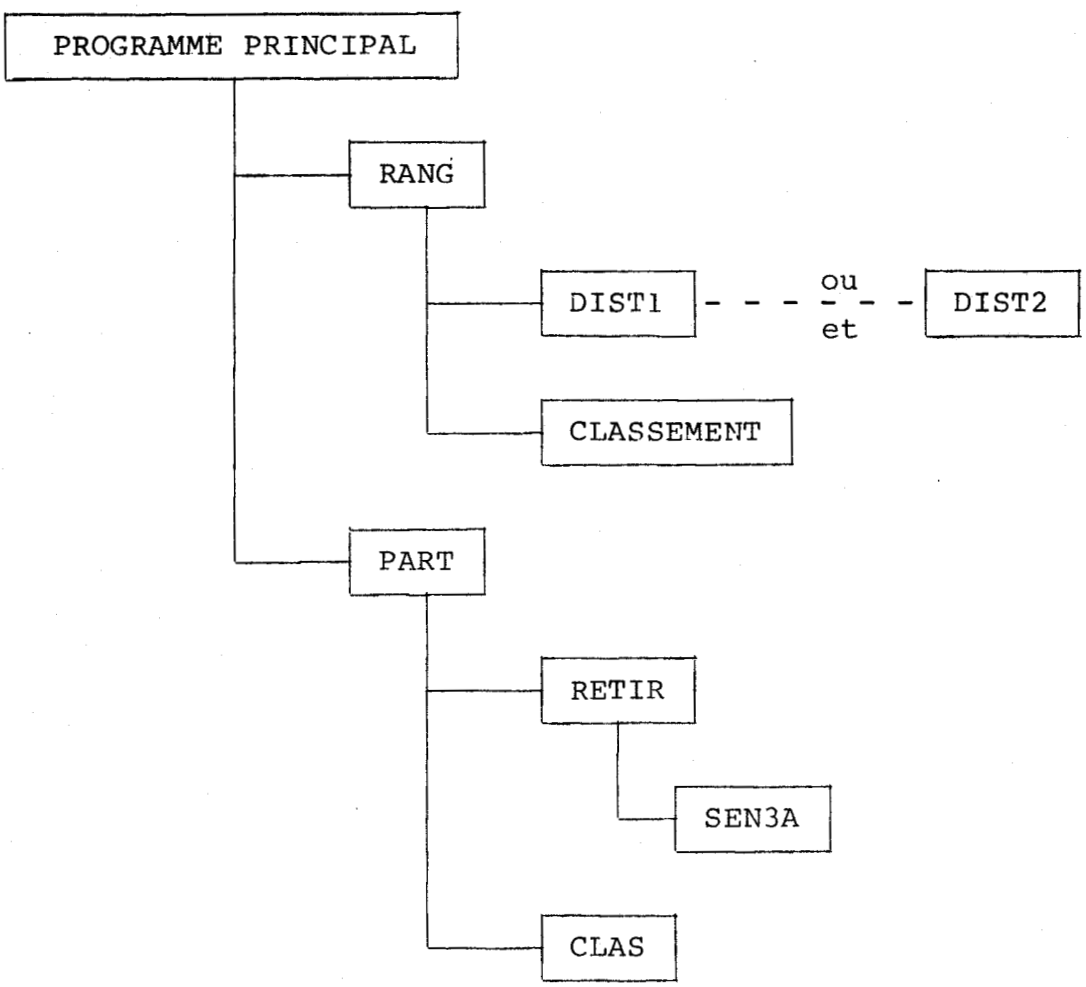
CHAPITRE IV

ORGANISATION DU PROGRAMME PERMETTANT LA MISE EN OEUVRE DE LA METHODE DECRITE AU CHAPITRE III.

Le programme FORTRAN présenté ici, permet l'utilisation de la méthode proposée au chapitre III dans le cas où l'espace $\mathbb{L}$ des noyaux est confondu avec l'espace E des individus à classer.

1 - DESCRIPTION DU PROGRAMME.-

1.1. - Organigramme.



1.2. - Programme principal.

La lecture des principaux paramètres de l'analyse se fait dans le programme principal. A partir de ces paramètres, on effectue le partage de la mémoire pour les différents tableaux nécessaires. Ce partage est fait de façon dynamique de manière à réduire au maximum l'encombrement mémoire : très peu de tableaux sont communs aux deux sous-programmes principaux et l'emplacement affecté à un tableau dans un sous-programme peut être réutilisé pour un tableau du second sous-programme. Dans cette version, l'emplacement mémoire est limité à 40 000 mots ; s'il est insuffisant pour l'analyse demandée, le programme s'arrête en imprimant la valeur de la mémoire nécessaire.

1.3. - Description des sous-programmes.

RANG — permet la construction de la matrice D. Pour chaque groupe de variables, cette matrice est construite ligne par ligne de façon à éviter le stockage de la matrice des distances associée à ce groupe.

PART — après la lecture ou le tirage au sort des noyaux, on construit la partition P^* associée à l'aide du sous-programme CLAS . Les résultats concernant l'affectation de chaque individu à une classe de P^* sont conservés pour la construction des formes fortes.

DIST1 — permet le calcul de la distance euclidienne classique entre un individu donné et tous les individus placés après lui dans le fichier des données. Ces distances sont stockées dans un fichier temporaire.

DIST2 même utilisation que DIST1 dans le cas d'une distance du χ^2.

CLASSEMENT pour un individu i fixé, ce sous-programme range par ordre croissant les distances d(i,j) pour $j \in E$. Il permet donc la construction d'une ligne de la matrice D.

RETIR effectue, parmi les n individus à classer, le tirage du nombre de noyaux nécessaires à l'aide de la fonction SEN3A (programme de génération de variable pseudo-aléatoire uniforme sur (0,1)).

Remarque : Ces deux sous-programmes sont ceux décrits par Lebart [10].

CLAS construit la partition P^n à partir de L^n, puis L^{n+1} à partir de P^n. Dans le cas où un élément P_i^n de la partition P^n est vide, un noyau λ_i^{n+1} est tiré au sort.

2 - NOTICE D'UTILISATION DU PROGRAMME.-

2.1. - Cartes paramètres.

2.1.1. - Description de la carte P1.

· Une seule carte lue en 20A4 : cette carte porte le titre du programme.

2.1.2. - Description de la carte P2.

Une seule carte lue en 10I4. Elle précise les paramètres principaux de l'analyse.

COL 1- 4 NI = nombre d'individus à classer.

COL 5- 8 NCR = nombre de groupes de variables.

COL 9-12 NCLAS = nombre de classes désirées par partition.

COL 13-16 NITER = nombre maximum d'itérations tolérées pour obtenir la convergence.

COL 17-20 NBT = nombre de tirages ou de lectures d'étalons.

COL 21-24 LETI = 0 si les étalons sont lus.
= 1 si les étalons sont tirés au sort.

COL 25-28 NVMAX = nombre maximum de variables pouvant être rencontrées pour les NCR groupes.

2.1.3. - Description de la carte P3.

Une seule carte lue en 20F4.0 : cette carte porte les poids respectifs de chacun des groupes de variables.

2.1.4. - Description du groupe de cartes P4.

Il y a NCR groupes de cartes P4 à lire. Chacun de ces groupes comporte 2 cartes :

- Carte P4.1 lue en 10I4 :

COL 1- 4 ILEC = étiquette logique du support des données du groupe de variables considéré.

COL 5- 8 IDIST = 1 si la distance utilisée est euclidienne classique.
= 2 si la distance est du type χ^2 .

COL 9-12 NV = nombre de variables pour le groupe considéré.

- Carte P4.2 lue en 20A4 : cette carte fournit le format de lecture sur ILEC.

2.1.5. - Description de la carte P5 *(optionnelle).*

Il y a NBT cartes P5. Ces cartes ne sont lues que dans le cas où LETI = 0 . Chacune d'elles porte les numéros des étalons (un étalon par classe). Chaque carte est lue avec le format 20I4.

Remarques : - l'identifiant des individus est toujours lu avec le premier groupe de variables.

- les différents groupes de variables peuvent être sur le même fichier ILEC.

- si les données sont sur carte, mettre ILEC = 5.

2.2. - Ensemble des cartes à constituer pour une analyse.

```
!JOB,A␣NURANG, n° de compte, chaîne
!LIMIT␣(CORE,120),(TIME,4),(SPDISC,600)
!ASSIGN␣5,DEV,IN
!ASSIGN␣6,DEV,OUT,DCB,VFC,(LIN,255)
!ASSIGN␣16,FIL,(SIZ,4),DCB,(ORG,D)
!ASSIGN␣7,FIL,(STS,OLD),(NAM,DONNEES1)
!ASSIGN␣8,FIL,(STS,OLD),(NAM,DONNEES2)
!RUN
Carte P1
Carte P2
Carte P3
Premier groupe de cartes P4 (correspond au premier groupe
de variables).
Mettre ici les cartes de données si ILEC = 5 ; supprimer
la carte !ASSIGN␣7 ..... .
```

Deuxième groupe de cartes P4 .

Mettre ici les cartes de données si ILEC = 5 ; supprimer la carte !ASSIGN␣8 ...

Cartes P5 éventuellement

!EOD .

Remarque : Cet ensemble de cartes a été présenté pour le cas où l'on dispose de deux groupes de variables

- le premier est sur le fichier 7 (ou 5) ,
- le second est sur le fichier 8 (ou 5) ,

et pour une utilisation sur IRIS 80.

3 - *LISTING*.-

Un listing du programme FORTRAN se trouve dans les pages suivantes.

```
C.....PROGRAMME PRINCIPAL APPEL DE DEUX SOUS PROGRAMMES :
C.... RANG = CALCUL DE MATRICE ID
C.... PART = PARTITION EN NITER ITERATIONS
C.....
      DIMENSION Q(40000)
      DOUBLE PRECISION DQ(100)
      EQUIVALENCE (DQ(1),Q(1))
      DIMENSION IT(20)
      M=40000
      COMMON/ENSOR/LEC,IMP
      LEC=5
      IMP=6
      READ(5,500)(IT(J),J=1,20)
      WRITE(6,510)(IT(J),J=1,20)
      READ(LEC,100)NI,NCR,NCLAS,NITER,NBT,LETI,NVMAX
      WRITE(IMP,110)NI,NCR,NCLAS,NITER,NBT
      WRITE(IMP,150) NVMAX
      IF(LETI.EQ.0) WRITE(IMP,130)
      IF(LETI.EQ.1) WRITE(IMP,120)
C.....CALCUL DE LA MEMOIRE NECESSAIRE
      NOT=NBT*NI
      QM1=2*NI*NI+5*NI+NI*NVMAX+NVMAX+NCR
      QM2=2*NI*NI+NCLAS*NI*2+5*NCLAS+2*NI+NOT+1
      QM=QM1
      IF(QM2.GT.QM1) QM=QM2
      WRITE(IMP,140)M,QM
      IF(QM.GT.M) GO TO 200
C.......  CALCULS DES EMPLACEMENTS DANS Q
      I1=1+2*NI*NI
      I2=I1+NI
      I3=I2+NI
      I4=I3+NI
      I5=I4+NI*NVMAX
      I6=I5+NI
      I7=I6+NVMAX
      I8=I7+NI
      CALL RANG(NI,NCR,NVMAX,DQ(1),Q(I1),Q(I2),Q(I3),Q(I4),Q(I5),
     1Q(I6),Q(I7),Q(I8))
C..................
C....CALCUL DES NOUVELLES VALEURS POUR LES EMPLACEMENTS DE Q;SEULS
C... ID ET IND DOIVENT ETRE CONSERVES
      I3=I2+NCLAS
      I4=I3+NCLAS
      I5=I4+NI
      IF(I5/2*2.EQ.I5) I5=I5+1
      I6=I5+2*NCLAS*NI
      I7=I6+2*NCLAS
      I8=I7+NCLAS
      I9=I8+NI
      CALL PART(NI,NCLAS,NITER,DQ(1),Q(I1),Q(I2),Q(I3),Q(I4),DQ(I5/2+1),
     1DQ(I6/2+1),NBT,LETI,Q(I7),Q(I8),Q(I9),NOT)
```

```
  100  FORMAT(10I4)
  110  FORMAT(5X,20HNOMBRE D INDIVIDUS =,I6/,
     15X,20HNOMBRE DE CRITERES =,I6/,
     25X,19HNOMBRE DE CLASSES =,I6/,
     35X,21HNOMBRE D ITERATIONS =,I6/,
     45X,31HNOMBRE DE TIRAGES A EFFECTUER =,I6//)
  120  FORMAT(5X,31HLES ETALONS SONT TIRES AU SORT )
  130  FORMAT(5X,21HLES ETALONS SONT LUS )
  140  FORMAT(/////,5X,23HUTILISATION DE MEMOIRE /,5X,
     118HVOUS AVEZ RESERVE ,I8,5X,19HVOUS AVEZ BESOIN DE,I8)
  150  FORMAT(5X,' NOMBRE MAXIMUM DE VARIABLES =',I6,//)
  500  FORMAT(20A4)
  510  FORMAT(10X,20A4////)
C.... FIN POUR L INSTANT
  200  CONTINUE
       END
```

```
      SUBROUTINE RANG(NI,NCR,NVMAX,ID,IND,IND1,X,T,SL,SC,IR,P)
      DIMENSION IND(NI),ID(NI,NI),IND1(NI),T(NI,NVMAX)
      DIMENSION FMT(20),X(NI),SL(NI),SC(NVMAX)
      DIMENSION IR(NI),P(NCR)
      COMMON/ENSOR/LEC,IMP
      DOUBLE PRECISION ID
C..LECTURE DU POIDS DE CHAQUE CRITERE
      READ(LEC,120) P(K),K=1,NCR
C....LE PREMIER CRITERE EST TRAITE A PART---CAUSE LECTURE DE
C..L IDENTIFIANT
      READ(LEC,100)ILEC,IDIST,NV
      READ(LEC,110)FMT(I),I=1,20
C.....
      IF(ILEC.NE.LEC) REWIND ILEC
      DO 2 I=1,NI
      READ(ILEC,FMT) IND(I),(T(I,J),J=1,NV)
   2  CONTINUE
      IF(IDIST.NE.2) GO TO 40
      DO 42 I=1,NI
  42  SL(I)=0.
      DO 43 J=1,NVMAX
  43  SC(J)=0.
      DO 41 I=1,NI
      DO 41 J=1,NV
      SC(J)=SC(J)+T(I,J)
  41  SL(I)=SL(I)+T(I,J)
  40  CONTINUE
C.........
      MK=1
      DO 3 I=1,NI
      IF(I.EQ.1) GO TO 47
C...RECHERCHE DES DISTANCES POUR J<I DANS FICHIER 16
      NK=1
      DO 48 K=1,I-1
      L=I-K-1
      READ DISC 16,NK+L,X(K)
      NK=NK+NI-K
  48  CONTINUE
  47  CONTINUE
C....CALCUL DES DISTANCES POUR J>=I LES AUTRES DISTANCES SONT
C..CONSERVEES DU CALCUL PRECEDENT PAR SYMETRIE
      GO TO (10,11) IDIST
  10  CALL DIST1(I,T,X,NI,NV)
      GO TO 20
  11  CALL DIST2(I,T,X,SC,SL,NI,NV)
  20  CONTINUE
C...STOCKAGE DES DISTANCES UTILES PAR LA SUITE
      WRITE DISC 16,MK,X(J),J=I+1,NI
      MK=MK+NI-I
C....CLASSEMENT  UNE LIGNE DE ID EST CREEE
      DO 21 IJ=1,NI
```

```
   21   IND1(IJ)=IJ
        DO 22 IJ=1,NI-1
   22   CALL CLASSEMENT(IJ,NI-IJ,X,IND1,NI)
        M=1
        DO 23 IJ=1,NI-1
        IF(X(IJ).EQ.X(IJ+1)) GO TO 60
        IR(IND1(IJ))=M
        M=M+1
        GO TO 23
   60   IRR=1
        DO 61 IK=IJ+2,NI
        IF(X(IK).NE.X(IJ)) GO TO 62
        IRR=IRR+1
   61   CONTINUE
   62   CONTINUE
        DO 63 IK=IJ,IJ+IRR
   63   IR(IND1(IK))=M
        M=M+IRR+1
        IJ=IJ+IRR
   23   CONTINUE
        IF(X(NI-1).NE.X(NI)) IR(IND1(NI))=NI
        DO 64 IJ=1,NI
   64   ID(I,IJ)=IR(IJ)*P(1)
    3   CONTINUE
C.....
        IF(NCR.EQ.1) GO TO 999
C...BOUCLE SUR LES AUTRES CRITERES
        DO 4 IN=2,NCR
        DO 5 I=1,NI
        DO 5 J=1,NVMAX
    5   T(I,J)=0.
        READ(LEC,100) ILEC,IDIST,NV
        READ(LEC,110) FMT(I),I=1,20
        IF(ILEC.NE.LEC) REWIND ILEC
        DO 7 I=1,NI
        READ(ILEC,FMT) (T(I,J),J=1,NV)
    7   CONTINUE
        IF(IDIST.NE.2) GO TO 51
        DO 44 I=1,NI
   44   SL(I)=0.
        DO 45 J=1,NVMAX
   45   SC(J)=0.
        DO 46 I=1,NI
        DO 46 J=1,NVMAX
        SC(J)=SC(J)+T(I,J)
   46   SL(I)=SL(I)+T(I,J)
   51   CONTINUE
C.........
        MK=1
        DO 8 I=1,NI
        IF(I.EQ.1) GO TO 49
C...RECHERCHE DES DISTANCES
        NK=1
        DO 50 K=1,I-1
```

```
      L=I-K-1
      READ DISC 16,NK+L,X(K)
      NK=NK+NI-K
   50 CONTINUE
   49 CONTINUE
C....CALCUL DES DISTANCES POUR J>=I LES AUTRES DISTANCES SONT
C..CONSERVEES DU CALCUL PRECEDENT PAR SYMETRIE
      GO TO (30,31) IDIST
   30 CALL DIST1(I,T,X,NI,NV)
      GO TO 25
   31 CALL DIST2(I,T,X,SC,SL,NI,NV)
   25 CONTINUE
C...STOCKAGE DES DISTANCES UTILES PAR LA SUITE
      WRITE DISC 16,MK,X(J),J=I+1,NI
      MK=MK+NI-I
C....CLASSEMENT
      DO 26 IJ=1,NI
   26 IND1(IJ)=IJ
      DO 27 IJ=1,NI-1
   27 CALL CLASSEMENT(IJ,NI-IJ,X,IND1,NI)
      M=1
      DO 65 IJ=1,NI-1
      IF(X(IJ).EQ.X(IJ+1)) GO TO 66
      IR(IND1(IJ))=M
      M=M+1
      GO TO 65
   66 IRR=1
      DO 67 IK=IJ+2,NI
      IF(X(IK).NE.X(IJ)) GO TO 68
      IRR=IRR+1
   67 CONTINUE
   68 CONTINUE
      DO 69 IK=IJ,IJ+IRR
   69 IR(IND1(IK))=M
      M=M+IRR+1
      IJ=IJ+IRR
   65 CONTINUE
      IF(X(NI-1).NE.X(NI)) IR(IND1(NI))=NI
      DO 28 IJ=1,NI
   28 ID(I,IJ)=ID(I,IJ)+IR(IJ)*P(IN)
    8 CONTINUE
    4 CONTINUE
C..............
      DO 70 K=1,NI
      U=1.0E-07*K
      DO 71 I=1,NI
   71 ID(I,K)=ID(I,K)+U
   70 CONTINUE
C............
  100 FORMAT(10I4)
  110 FORMAT(20A4)
  120 FORMAT(20F4.0)
  999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PART(NI,NCLAS,NITER,ID,IND,IU,PCLAS,KLAS,IR,MIN,
     1NBT,LETI,IU1,IV,ICU,NOT)
      DIMENSION ID(NI,NI),IND(NI),IU(NCLAS),PCLAS(NCLAS),KLAS(NI)
      DIMENSION IR(NCLAS,NI),MIN(NCLAS),IU1(NCLAS),IV(NI)
      DIMENSION ICU(NOT)
      DIMENSION FMT1(15),FMT2(14),FMT3(8)
      DOUBLE PRECISION ID,IR,MIN,W1,W2
      COMMON/ENSOR/LEC,IMP
      DO 50 NT=1,NBT
      WRITE(IMP,170)
      IF(LETI.EQ.1) GO TO 14
      READ(LEC,140)(IU(N),N=1,NCLAS)
      GO TO 15
   14 CONTINUE
C...RECHERCHE DES ETALONS PAR TIRAGE AU SORT
      CALL RETIR(NI,KLAS,NCLAS,IU)
   15 WRITE(IMP,100)(IND(IU(N)),N=1,NCLAS)
C... CONSTRUCTION DE PARTITION A PARTIR DES ETALONS CHOISIS
      DO 1 NIT=1,NITER
      DO 2 IK=1,NCLAS
    2 IU1(IK)=IU(IK)
      CALL CLAS(NI,NCLAS,IU,KLAS,PCLAS,ID,IR,MIN,W1,W2,IV)
C....IMPRESSION DE LA VALEUR DU CRITERE....
      WRITE(IMP,200)NIT,W1,W2
  200 FORMAT(1X,'ITERATION NUMERO ',I4/4X,'VALEUR DU CRITERE AVANT ',
     1F16.8,3X,'VALEUR DU CRITERE APRES ',F16.8)
C.........................
C.........................
      DO 3 IK=1,NCLAS
      IF(IU(IK).NE.IU1(IK)) GO TO 1
    3 CONTINUE
      GO TO 4
    1 CONTINUE
    4 IF(NIT.GE.NITER) GO TO 51
      WRITE(IMP,110)NIT
      DO 10 K=1,NCLAS
      IF(PCLAS(K).EQ.0.) GO TO 10
      I=0
      J=0
      JA=PCLAS(K)
      DO 11 L=1,JA
   12 I=I+1
      IF(KLAS(I).NE.K) GO TO 12
      J=J+1
      IV(J)=IND(I)
   11 CONTINUE
      WRITE(IMP,120)K,PCLAS(K),IND(IU(K)),(IV(J),J=1,JA)
   10 CONTINUE
C....STOCKAGE DES RESULTATS DANS ICU EN VUE DU CALCUL DES FORMES
C....FORTES
      IN=(NT-1)*NI
```

```
      DO 13 I=1,NI
      IN=IN+1
   13 ICU(IN)=KLAS(I)
C...FIN DU STOCKAGE  PASSAGE A L ITERATION SUIVANTE
      GO TO 50
   51 WRITE(IMP,180)NITER
   50 CONTINUE
C....CALCUL DES FORMES FORTES
      WRITE(IMP,150)
      DO 30 I=1,NI
      IV(I)=0
   30 KLAS(I)=I
      M=1
      IDA=NBT
      K=1
      I=1
   31 K=K+1
      IDO=0
      JA=-NI
      DO 16 NH=1,NBT
      JA=JA+NI
   16 IF(ICU(JA+KLAS(I)).NE.ICU(JA+KLAS(K))) IDO=IDO+1
      IF(IDO.GT.0) GO TO 17
      I=I+1
      JA=KLAS(I)
      KLAS(I)=KLAS(K)
      KLAS(K)=JA
      IV(KLAS(I))=IDO
      IF(KA.EQ.I) KA=K
      IF(I.EQ.NI) GO TO 18
      GO TO 19
   17 IF(IDO.GT.IDA) GO TO 19
      IDA=IDO
      KA=K
   19 IF(K.NE.NI) GO TO 31
      IF(IDO.EQ.0) GO TO 20
      I=I+1
      JA=KLAS(I)
      KLAS(I)=KLAS(KA)
      IV(KLAS(I))=IDA
      KLAS(KA)=JA
   20 CONTINUE
      IF(I.GE.NI) GO TO 18
      K=I
      M=1
      IDA=NBT
      GO TO 31
   18 CONTINUE
      IPP=1
  151 IP2=IPP+17
      IF(IP2.GT.NBT) IP2=NBT
```

```
      JC=(IPP-1)*NI
      JB=(IP2-1)*NI
      NET=(IP2-IPP+1)
      NET2=(NET*8)+7
      ENCODE(60,901,FMT1)NET2,NET,NET2
  901 FORMAT(5H(1HO,,I3,21H(1H*),/,1X,'* NOM *',,I3,
     117H(I4,3X,1H*),/,1X,,I3,8H(1H*))    )
      WRITE(IMP,FMT1)(I,I=IPP,IP2)
      ENCODE(32,903,FMT3)NET
  903 FORMAT(17H(1HO,'*',A4,' *',,I3,12H(I5,2X,1H*)))
      DO 74 I=1,NI
      M=KLAS(I)
      JA=JB+M
      JD=JC+M
   74 WRITE(IMP,FMT3)IND(M),(ICU(K),K=JD,JA,NI)
      IF(IP2.EQ.NBT) GO TO 152
      IPP=IPP+18
      GO TO 151
  152 CONTINUE
      WRITE(IMP,170)
      WRITE(IMP,160)
      DO 108 I=1,NI-1
      NET=17
      M=KLAS(I)
      NM=KLAS(I+1)
      IFF=IV(NM)
      IF(IFF.NE.0) NET=NET+8
      IF(IFF.EQ.NBT) NET=NET+8
      ENCODE(56,902,FMT2)NET
      WRITE(IMP,FMT2)IND(M),IV(M)
  108 CONTINUE
      NET=33
      M=KLAS(NI)
      ENCODE(56,902,FMT2)NET
  902 FORMAT(47H(1H ,'*  ',A4,' * ',I4,'  *',2('   !   *'),/1X,,I3,
     16H(1H*)) )
      WRITE(IMP,FMT2)IND(M),IV(M)
C.....FIN DU CALCUL ET IMPRESSION DES FORMES FORTES

  100 FORMAT(10X,40HCONSTRUCTION D UNE PARTITION A PARTIR DE,
     1/10X,20(2X,A4))
  110 FORMAT(1X,29HCOMPOSITION DES CLASSES APRES,I4,11H ITERATIONS)
  120 FORMAT(/////,1X,16H*******CLASSE  :,I6,//,1X,11HEFFECTIF  :,F6.0,//,
     '11X,23HELEMENT REPRESENTATIF :,A4,/,(10(1X,A4)))
  140 FORMAT(20I4)
  150 FORMAT(1H1,30H******CALCUL DES FORMES FORTES)
  160 FORMAT(1HO,/,1X,33(1H*)/1X,'*  NOM  * DELTA *F.FORT *F.FAIBL*',
     1/1X,33(1H*))
  170 FORMAT(1H1)
  180 FORMAT(//,1X,'LA CONVERGENCE N EST PAS ATTEINTE APRES',I4,
     1' ITERATIONS')
      RETURN
      END
```

```
      SUBROUTINE DIST1(I,T,X,NI,NV)
      DIMENSION T(NI,NV),X(NI)
      COMMON/ENSOR/LEC,IMP
      DO 1 K=I,NI
      X(K)=0.
      DO 2 IV=1,NV
      AX=T(I,IV)-T(K,IV)
      AX=AX*AX
    2 X(K)=X(K)+AX
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DIST2(I,T,X,SC,SL,NI,NV)
      DIMENSION T(NI,NV),X(NI),SL(NI),SC(NV)
      COMMON/ENSOR/LEC,IMP
      DO 1 K=I,NI
      X(K)=0.
      DO 2 IV=1,NV
      AX1=T(I,IV)/SL(I)
      AX2=T(K,IV)/SL(K)
      AX=AX1-AX2
      AX=AX*AX
    2 X(K)=X(K)+AX/SC(IV)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CLASSEMENT(J,K,T,NUM,N)
      DIMENSION T(N),NUM(N)
      AMIN=T(J)
      DO 1 L=J+1,J+K
      IF(AMIN.LE.T(L)) GO TO 1
      AMIN=T(L)
      NIND=NUM(L)
      T(L)=T(J)
      NUM(L)=NUM(J)
      T(J)=AMIN
      NUM(J)=NIND
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RETIR(N,IDN,K,IDK)
C.......
C..TIRAGE SANS REMISE DE K INDICES ENTRE 1 ET N,STOCKES DANS IDK(K)
C..IDN(N) EST UN VECTEUR DE TRAVAIL
C.......
      DIMENSION IDN(N),IDK(K)
      DO 10 J=1,N
   10 IDN(J)=J
      KFIN=K
      IF(K.GT.N)  KFIN=N
      DO 30 L=1,KFIN
      X=N-L+1
      I=X*SEN3A(BIDON)+1.0
      IDK(L)=IDN(I)
      IF(I.EQ.N)  GO TO 30
      N1=N-L
      IF(N1.LE.0) GO TO 30
      DO 20 J=1,N1
   20 IDN(J)=IDN(J+1)
   30 CONTINUE
      RETURN
      END
```

```
      FUNCTION SEN3A(BIDON)
      DATAM12/4096/
      DATA F1/2.44140625E-04/,F2/5.96046448E-08/,F3/1.45519152E-11/
      DATA J1/3823/          ,J2/4006/          ,J3/2903/
C.....DONNEES INITIALES PUIS VALEURS CALCULEES
      DATA I1/3823/          ,I2/4006/          ,I3/2903/
C.... CALCUL DE CONGRUENCE AVEC SEGMENTATION DES NOMBRES
      K3=I3*J3
      L3=K3/M12
      K2=I2*J3+I3*J2+L3
      L2=K2/M12
      K1=I1*J3+I2*J2+I3*J1+L2
      L1=K1/M12
      I1=K1-L1*M12
      I2=K2-L2*M12
      I3=K3-L3*M12
      SEN3A=F1*FLOAT(I1)+F2*FLOAT(I2)+F3*FLOAT(I3)
      RETURN
      END
```

```
      SUBROUTINE CLAS(NI,NCLAS,IU,KLAS,PCLAS,ID,IR,MIN,W1,W2,IV)
C.....
C..CONSTRUCTION DES DE LA PARTITION  A CHAQUE ETAPE LES NOYAUX CHANGENT
C.....
      DIMENSION ID(NI,NI),IU(NCLAS),KLAS(NI),IR(NCLAS,NI),MIN(NCLAS)
      DIMENSION PCLAS(1),IV(NI)
      DOUBLE PRECISION ID,IR,MIN,W1,W2,IDIST
      DO 1 N=1,NCLAS
      MIN(N)=100000
    1 PCLAS(N)=0.0
C... CLASSEMENT DES INDIVIDUS AUTOUR DU NOYAU LE PLUS PROCHE
      DO 2 I=1,NI
      IDIST=ID(I,IU(1))
      KAT=1
      DO 3 N=2,NCLAS
      IF(ID(I,IU(N)).GE.IDIST) GO TO 3
      IDIST=ID(I,IU(N))
      KAT=N
    3 CONTINUE
      KLAS(I)=KAT
      PCLAS(KAT)=PCLAS(KAT)+1.0
    2 CONTINUE
C...W1 ANCIENS ETALONS...W2 NOUVEAUX ETALONS....
C..CALCUL DE W1
      W1=0.
      DO 10 N=1,NCLAS
      DO 11 I=1,NI
      IF(KLAS(I).NE.N) GO TO 11
      W1=W1+ID(I,IU(N))
   11 CONTINUE
   10 CONTINUE
C......
C...RECHERCHE DES NOUVEAUX ETALONS
      DO 4 IE=1,NI
      DO 4 N=1,NCLAS
    4 IR(N,IE)=0
      DO 5 IE=1,NI
      DO 6 I=1,NI
      KAT=KLAS(I)
    6 IR(KAT,IE)=IR(KAT,IE)+ID(I,IE)
      DO 7 K=1,NCLAS
      IF(IR(K,IE).GE.MIN(K)) GO TO 7
      MIN(K)=IR(K,IE)
      IU(K)=IE
    7 CONTINUE
    5 CONTINUE
C.....DANS LE CAS OU LA CLASSE EST VIDE NOUVEAU NOYAU EST TIRE AU SORT
      DO 14 K=1,NCLAS
      IF(PCLAS(K).NE.0.) GO TO 14
      CALL RETIR(NI,IV,1,IU2)
      IU(K)=IU2
```

```
  14  CONTINUE
C..CALCUL DE W2
      W2=0
      DO 12 N=1,NCLAS
      DO 13 I=1,NI
      IF(KLAS(I).NE.N) GO TO 13
      W2=W2+ID(I,IU(N))
  13  CONTINUE
  12  CONTINUE
C......
      RETURN
      END
```

B I B L I O G R A P H I E.

[1] BARBAISE G., LANGRAND C. - *Analyse typologique "robuste" et tests statistiques.* Communication : Journées de Statistique Bruxelles mai 1982.

[2] CAILLIEZ F., PAGES J.P. - *Introduction à l'analyse des données.* SMASH 1976.

[3] DE FALGUEROLLES A. - *Classification automatique : un critère et des algorithmes d'échange.* Séminaire IRIA 1977.

[4] DELATTRE M. - *Classification optimale bicritère. Méthodes, algorithmes et applications.* Thèse Faculté Universitaire Catholique de Mons. Décembre 1979.

[5] DELATTRE M., HANSEN P. - *Bicriterion cluster analysis.* IEEE Transactions on pattern analysis and machine intelligence. Vol. PAMI2, n° 4, july 1980.

[6] DIDAY E. - *Optimisation en classification automatique et reconnaissance des formes.* Note scientifique n° 6 : supplément au Bulletin de l'IRIA n° 12 (1972).

[7] KRUSKAL J.B. - *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis.* Psychometrika Vol. 29 n° 1 March 1964.

[8] KRUSKAL J.B. - *Nonmetric multidimensional scaling : a numerical method.* Psychometrika Vol. 29 n° 2 June 1964.

./..

[9] LEBART L. - *Programme d'agrégation avec contraintes.* Cahiers de l'analyse des données, Vol. III, n° 3, 1978.

[10] LEBART L., MORINEAU A., TABARD N. - *Techniques de la description statistique. Méthodes et logiciels pour l'analyse des grands tableaux.* Dunod 1977.

[11] LERMAN I.C. - *Classification et analyse ordinale des données.* Dunod 1981.

[12] MARDIA K.V., KENT J.T., BIBBY J.M. - *Multivariate analysis.* Academic press 1979.

[13] MONESTIEZ P. - *Méthodes de classification automatique sous contraintes spatiales.* Statistique et Analyse des données 3, 1977.

[14] PERRUCHET C. - *Classification sous contrainte de contiguïté continue. Application aux sciences de la terre.* Thèse 3ème cycle Paris VI, novembre 1979.

[15] PROD'HOMME A. - *Indice d'explication des classes obtenues par une méthode de classification hiérarchique respectant la contrainte de contiguïté spatiale. Application à la viticulture girondine et à la construction de logements dans les Bouches-du-Rhône.* Thèse 3ème cycle Rennes I, décembre 1980.

[16] REGNIER S. - *Sur quelques aspects mathématiques des problèmes de classification automatique.* ICC Bulletin Vol. 4, 1965.

./..

[17] ROCHE C. - *Exemple de classification hiérarchique avec contraintes de contiguïté : le partage d'Aix-en-Provence en quartiers homogènes.* Cahiers de l'Analyse des Données, Vol. III, n° 3, 1978.

[18] WILKINSON J.H. - *The algebraic eigenvalue problem.* Clarendon Press, Oxford 1965.

RÉSUMÉ

Dans la pratique du travail de statisticien les méthodes de classification ont toujours joué un rôle fondamental en permettant un découpage simplificateur de la réalité étudiée.

Ces dernières années Lebart, Perruchet, Delattre,... ont proposé des méthodes de classification qui prennent en compte simultanément plusieurs critères ou qui opèrent sous contraintes. Une synthèse sur l'ensemble de ces méthodes fait l'objet du chapitre 1.

Disposant d'"individus" sur lesquels on a mesuré les variables, on se pose le problème de constituer des classes d'individus qui soient homogènes vis à vis de ces variables et qui rassemblent, de plus, des individus "semblables" eu égard à un autre critère (de localisation géographique par exemple). Dans le chapitre 2 on présente une méthode permettant d'effectuer de telles classifications.

Le chapitre 3 quant à lui, propose un algorithme de type "Nuées Dynamiques" dans lequel une utilisation systématique de rangs permet de prendre en compte simultanément plusieurs critères. Des exemples sont traités et montrent l'utilisation des programmes informatiques du chapitre 4.

MOTS CLÉS : CLASSIFICATION
CONTRAINTE
ANALYSE MULTICRITERE .