REVUE DE STATISTIQUE APPLIQUÉE

PH. LEHERT

Classification en composantes connexes, cas particulier de l'ultramétrique inférieure maximale : un algorithme $O(n)$ en temps moyen

Revue de statistique appliquée, tome 40, n° 3 (1992), p. 63-72

<http://www.numdam.org/item?id=RSA_1992__40_3_63_0>

Rev. Statistique Appliquée, 1992, **XXXX** (3), 63-72

CLASSIFICATION EN COMPOSANTES CONNEXES, CAS PARTICULIER DE L'ULTRAMÉTRIQUE INFÉRIEURE MAXIMALE : UN ALGORITHME *O(N)* EN TEMPS MOYEN

Ph. Lehert

Facultés Universitaires Catholiques de Mons (Belgique)
Université libre de Lille (France)

RÉSUMÉ

Un algorithme particulier dérivé de la classification hiérarchique est présenté, destiné à classifier de très grands ensembles de données $(n > 50\,000)$ dans des espaces métriques, avec utilisation des distances de Minkowsky. Un comportement $O(n)$ en temps moyen est démontré. A notre connaissance, cet algorithme est le seul qui possède conjointement des propriétés de maximisation de l'écart de la partition résultante et d'une complexité de calcul minimale $O(n)$. En pratique, l'algorithme est particulièrement adapté lorsque les variables sont en petit nombre.

Mots-clés : *Composantes connexes, Arbre de Longueur Minimum, Complexité, Pavage, Filtration, Adressage pseudo aléatoire.*

ABSTRACT

A very fast Algorithm is proposed for clustering data sets in coordinate p-spaces by single linkage analysis at a predetermined threshold μ. This algorithm is proved to run in $O(n)$ expected time, for any Minkowsky distance. It is probably the first algorithm to combine exact output partition and minimum $O(n)$ complexity. Experiments seem to confirm that the interest of this approach is limited to low dimensions $(p < 5)$.

Key-words : *Connected Components, Minimum Spanning Tree, Single Link, Hashing, Square Lattice, Percolation.*

Introduction

Parmi l'ensemble des techniques de classification automatique, la classification ascendante hiérarchique est une technique couramment utilisée, mais souvent restreinte à de petits échantillons, étant donné la complexité des algorithmes de construction de la hiérarchie. Considérant des échantillons d'effectifs n, l'algorithme élémentaire de construction qui est de complexité $O(n^3)$, peut être ramené à

$O(n^2 \log(n))$ par une structure de données appropriées [7], et à des performances plus marquantes via des structures particulières [11] [12]. L'ultramétrique inférieure maximale est un cas particulier et peut s'obtenir directement par la construction de l'arbre de longueur minimum en $O(n^2)$ [3]. Ce comportement quadratique restant prohibitif pour l'analyse des grands ensembles, on a recherché des structures géométriques permettant de diminuer le calcul des distances : parmi les nombreuses approches proposées, on note un comportement prouvé $O(n \log(n))$ en temps moyen, et d'autres algorithmes expérimentalement estimés mais on démontrés à $O(n \log\log(n))$ temps moyen [8], [13].

Au sein de la méthode ultramétrique inférieure maximale, on peut distinguer le problème particulier de la classification en composantes connexes CC d'un ensemble de points E, associé à une fonction de distance $\delta(X, Y)$, X et $Y \in E$, et un seuil prédéterminé μ, qui peut se définir comme suit : soit $G(E, F)$ le graphe non orienté et F l'ensemble des arêtes reliant tout couple (X, Y) tels que $\delta(X, Y) < \mu$. Les composantes connexes $\{CC1, \ldots, CCk\}$ sont les classes de la partition telle que deux points appartiennent à la même classe si et seulement si il existe une chaîne les joignant. La partition CC peut être obtenue par la construction de l'arbre de longueur minimum ALM, par la suppression de toutes les arêtes de poids supérieur à μ. Sous cet aspect, la partition CC est une des partitions délivrées par l'ultramétrique inférieure maximale, et de ce fait est optimale dans le sens suivant : définissant l'écart associé à une partition comme la distance minimum entre deux points appartenant à deux classes différentes, la partition CC a un écart maximal parmi l'ensemble de toutes les partitions de E en un même nombre de classes.

Dans de nombreuses application, le seuil μ est fixé [1] [2], ce qui limite l'intérêt de la construction du dendrogramme associé à toute la hiérarchie, et on a donc cherché à mettre en évidence un algorithme plus rapide que via la construction de l'ALM. L'algorithme élémentaire est de complexité $O(n^2)$, puisque requérant l'ensemble des distances. Un algorithme a été proposé, permettant par une structure de pavage cubique d'assurer une complexité $O(n)$ dans l'utilisation particulière de la Max-Norm distance ou distance de Tchebychev d_∞ [6]. Nous proposerons désormais un algorithme permettant l'obtention de la partition CC, plus rapide, permettant l'utilisation de toute distance de Minkowsky, et que nous démontrerons opérer en $O(n)$ temps moyen.

Algorithme

Soit E, un ensemble de n points $X_i(x_{i1}, \ldots, x_{ip})$ dans R^p. Sans perte de généralité, considérons tout $x_{ij} > 0$. Un pavage cubique $\Omega(\sigma)$ est défini comme une partition de E en cellules $C_I - I$ désignant un p-uple d'entiers $(i_1, \ldots, i_p)$ – telles que

$$C_I = \{X_s \in E | \ulcorner x(s,j)/\sigma \urcorner = i_j, \quad j = 1, \ldots, p\},$$

La notation $\ulcorner x \urcorner$ désignant le plus petit nombre entier directement supérieur à x. Soit Γ, ensemble des C_I non vides. Définissons le voisinage direct $V(I)$ d'une

cellule C_I, comme l'ensemble des C_J tel que :

$$V(I) = \{C_J \in \Gamma | d_1(I, J) = 1\},$$

d_1 désignant la distance du city block. On désignera de façon générale par d_r la distance de Minkowsky d'ordre r, d_∞ correspondant à la Max-norm distance. On définira de même le voisinage complet $W(I)$ de C_I :

$$W(I) = \{C_J \in \Gamma | d_\infty(I, J) \leqslant 1\}$$

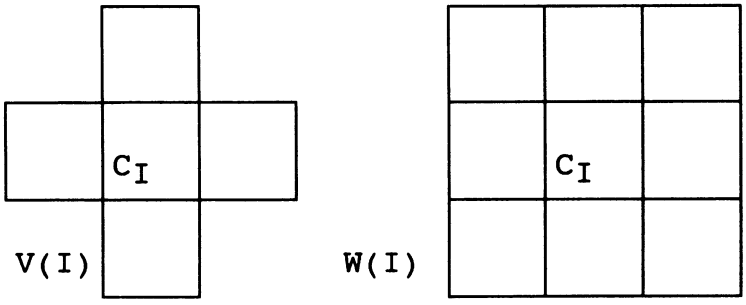

FIGURE 1
Voisinages direct $V(I)$ et complet $W(I)$ dans R^2

La figure 1 visualise les voisinages complet et direct d'une cellule C_I. De manière intuitive, raisonnant sur R^2, le voisinage direct d'une cellule C_I est l'ensemble des cellules non vides C_J qui ont avec C_I une arête commune. Il y en a 4 au maximum dans R^2. Dans R^p, il peut y en avoir $2p$. Le voisinage complet $W(I)$ est l'ensemble des cellules qui ont au moins un point commun avec C_I. Il y en a 9 dans R^2 et 3^p dans R^p.

La construction de l'algorithme est basée sur quelques propriétés géométriques simples dont voici la première :

Propriété 1. – Considérant une distance de Minkowsky d_r dans R^p, un seuil μ associé, la partition CC recherchée, un pavage $\Omega(\sigma = \mu(p - 1 + 2^r)^{-1/r})$ déterminant un ensemble de cellule $\{C\}$, tout point X d'une cellule C_I et tout point_Y d'une cellule C_J appartenant à $V(I)$ appartiennent à la même classe de la partition CC.

Démontrons cette simple propriété, nous référant à la figure 2, visualisant le cas simple du plan ($p = 2$), et la distance euclidienne ($r = 2$). Dans le cas de deux cellules voisines au sens défini ci-dessus, la distance la plus grande entre deux points x de C_I et Y de C_J est celle de la diagonale construite sur le rectangle $\{C_1 \cup C_J\}$. Fixant sa longueur à μ, tout point de C_I et C_J sera forcément dans la même classe CC. Or,

$$\mu^2 = \sigma^2 + (2\sigma)^2$$

et en généralisant pour tout R^p, et pour tout r,

$$\mu^r = (p-1)\sigma^r + (2\sigma)^r$$

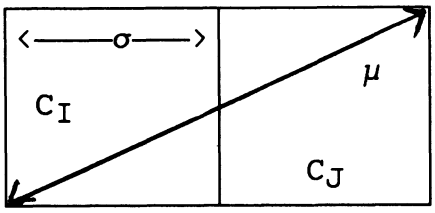

FIGURE 2
Deux cellules C_I, C_J, avec $C_J \in V(I)$ dans R^2, $r = 2$.

Considérons un pavage $\Omega(\sigma)$ et construisons une partition P comme suit : initialisons la classe P_1 par le choix d'une cellule quelconque C_I. En vertu de la propriété précédente, non seulement tous les points de C_I sont dans la même composante connexe, mais aussi tout point Y de l'ensemble des points appartenant au voisinage direct $V(I)$ est tel qu'il existe un point X de C_I tel que $\delta(x, y) < \mu$, et par conséquent Y appartient à la même classe de la partition CC. De cette manière, sans aucun calcul de distance on pourra annexer automatiquement tous les points de $V(I)$, en investiguant les $2p$ possibles cellules de $V(I)$. Toute nouvelle cellule C_J annexée dans CC_1 nécessitera le même examen du voisinage $V(J)$ et conduira en l'annexion de nouvelles cellules, ce processus s'interrompant lorsque toute cellule annexée aura eu son voisinage investigué. A l'aide d'une cellule non vide non encore affectée, on initialisera une nouvelle classe P_2, et ainsi de suite jusqu'à l'annexion de toutes les cellules. La partition $P(P_1, \ldots, P_p)$ est définie sur la graphe $G(\Gamma, \Phi)$ tel que Γ est l'ensemble des cellules non vides du pavage, et

$$\Phi = \{(C_I, C_J) | C_I \in \Gamma, C_J \in \Gamma, \text{ et } C_J \in V(I)\}$$

P est au moins aussi fine que la partition CC recherchée : d'une part si deux points sont connectés dans P, ils le sont à fortiori dans CC, mais il est possible de trouver des points connectés dans CC qui ne le sont pas dans P. Un exemple d'un tel lien est illustré en figure 3 : le point X dans la cellule C_I et le point Y dans C_J sont distants de moins du seuil μ, cependant, vu l'absence de points dans le voisinage $V(I)$ (les 4 carrés hachurés), C_J ne sera pas annexée dans la même classe que C_I. Il est donc nécessaire, une fois la partition Φ obtenue, de s'assurer de l'existence de tels liens, en examinant le voisinage de chaque point X, et recherchant l'existence d'un point Y appartenant à une classe de P différente, tout en réalisant $\delta(x, y) < \mu$.

En vue de limiter au maximum les calculs de distances, on utilisera, à partir du pavage initial de rayon σ, un nouveau pavage regroupé, de côté τ, défini par

$$\tau = \lceil \mu/\sigma \rceil \sigma.$$

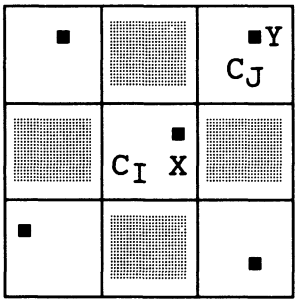

FIGURE 3
$d(X,Y) < \mu$ *et* $C_J \in V(I)$

Propriété 2 : Soit un pavage imbriqué $\Omega(\tau = \lceil \mu/\sigma \rceil \sigma)$ déterminant l'ensemble de cellules non vides $\{E\}$. Pour tout point X inclus dans une cellule E_I, le voisinage complet $W(I)$ d'une cellule C_I contient la boule $B_r(X,\mu)$ définie par :

$$B_r(x,\mu) = \{Y \in E | d_r(X,Y) < \mu\}$$

Ce nouveau pavage $\Omega(\tau)$ basé sur le pavage initial permet de limiter la recherche des points Y affectés à une autre classe de la partition P, et tels que la distance $\delta(X,Y) < \mu$. Par conséquent, le calcul des distances autour de chaque point sera limité aux 3^p cellules du pavage $\Omega(\tau)$ formant le voisinage complet $W(I)$. La figure 4 illustre le cas simple du plan $(p = 2)$ et la distance euclidienne $(r = 2)$: considérant $\mu = 1$, $\sigma = 1/\sqrt{5}$ et donc $\tau = \lceil \sqrt{5} \rceil \sigma = 3\sigma = 3/\sqrt{5}$. Chaque cellule $E(I)$ du nouveau pavage a une aire égale à 9 fois celle des cellules C_I du pavage initial. Il est clair que pour tout point X appartenant à la cellule C_I, le voisinage complet $W(I)$ (entouré en gros traits), contient forcément tous les points Y tels que $d(X,Y) < \mu$. Dans le plan, le nombre de cellules E_J constituant $W(I)$ est au plus égal à $3^2 = 9$.

En conclusion, l'algorithme CC comporte donc deux phases essentielles : la première au niveau des cellules décèle la partition préliminaire P, la seconde au niveau des points contrôle l'existence de liens non détectés dans la première phase :

Phase 1. Recherche de la partition P ***sur graphe*** $G(\Gamma, \Phi)$ ***:***

a) Prétraitement : Construction de $\Omega(\sigma)$ définissant un ensemble de cellules $\{C\}$; $k = 0$;

b) Obtention de la partition P **:**

Tant qu'il reste des cellules non annexées à P,
faire
$k = k + 1$;
$P_k = \{C_v\}$, C_v désignant une cellule non annexée
Tant qu'existe $C_I \in P_k$ à investiguer,

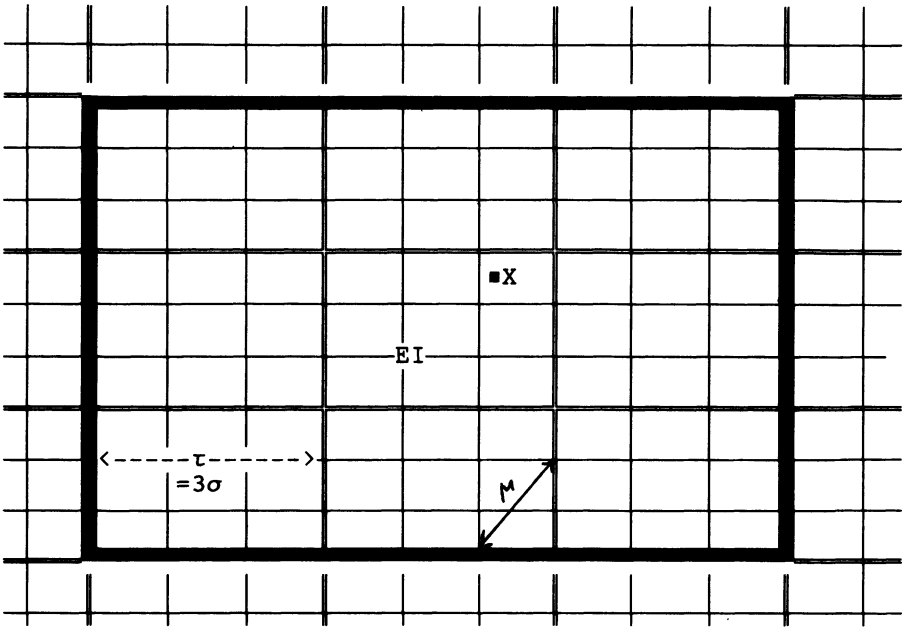

FIGURE 4

Plan et distance euclidienne ($p = r = 2$) *: pavage* $\Omega(\sigma)$ *(ligne simple) et pavage* $\Omega(\tau = 3\sigma)$ *(ligne double). Un point* X *de* E_I *a tout son voisinage* $B_2(X, \mu)$ *de rayon* μ *contenu dans les 9 cellules formant le voisinage complet* $W(I)$.

faire
$P_k = P_k + \{V(I)\}$
Fin
Fin

Phase 2. Recherche de la classe finale CC **:**

a) Regroupement du pavage initial de côté $\Omega(\sigma)$ en pavage imbriqué $\Omega(\tau = \lceil \mu/\sigma \rceil \sigma)$ définissant l'ensemble des cellules $\{E\}$.

b) Pour toute classe P_k de P,

Faire
Pour tout point $X \in E_I$ et contenu dans P_k,
Faire
Pour tout $Y \in W(I)$ appartenant à une autre P_1,
Faire
si $dr(X, Y) < \mu$, P_1 est annexée à P_k.
Fin
Fin
Fin

Complexité

Dans la suite, on démontre la complexité T en temps moyen de cet algorithme dans l'hypothèse de n points situés dans R^p, et une distance d_r de Minkowsky, $(0 < r < \infty)$.

La phase 1 consiste d'abord en la construction du pavage $\Omega(\sigma)$: celui-ci est supposé organisé selon une structure de hachage sur la clé des cellules et requiert $O(n)$ calculs[6]. La recherche du voisinage $V(I)$ pour chaque cellule C_I requiert $2p$ recherches $O(p)$, et notant le nombre de cellules non vides par $N(< n)$, on obtient :

$$T(\text{phase 1}) = O(n) + 2NpO(p) < O(n) + 2pO(np) \approx O(np^2) \qquad (1)$$

La phase 2 consiste à examiner le voisinage complet $W(I)$ de tout point X dans la pavage imbriqué $\Omega(\tau)$. Pour les besoins de la démonstration, introduisons le graphe dit réseau discret à maille cubique (N^p, Θ), N désignant l'ensemble des naturels, les nœuds étant composés de l'ensemble des p-uples $I(i_1, \ldots, i_p)$ de N^p et Θ défini par

$$\Theta = \{(I, J) | I \text{ et } J \in N^p, \quad \text{et} \quad d_1(I, J) = 1\}$$

Considérons sur le réseau un sous-ensemble M de N^p, sélectionné de manière aléatoire et uniforme avec une certaine densité α, en ne considérant que le sous-ensemble $\Theta' \in \Theta$ des arêtes reliant deux nœuds appartenant à M. Le graphe aléatoire (M, Θ') contient un nombre de composantes connexes, qui clairement augmente avec la densité α. On dispose à ce sujet d'une propriété importante :

Propriété 3. – Sur un graphe aléatoire défini sur un réseau discret, l'espérance du nombre de nœuds connectés est infinie, dès que la densité dépasse une valeur π dite densité critique de filtration. Cette valeur, finie, ne dépend que du type de maillage du réseau et de la dimension de l'espace. Cette propriété a été démontrée dans R^2 et R^3 [9][14], et généralisée à R^p [9][10].

Revenant au problème étudié, considérons dans une sous-région de R^p suffisamment petite pour assimiler la densité de points à une densité uniforme, le sous-graphe (M, Θ') tel que M est le sous-ensemble $\{J\}$ de N^p tel que C_J est non vide. La relation entre le pavage $\Omega(\sigma)$ et (M, Θ') est visualisé par un exemple (Fig. 5).

La sous-région étudiée est caractérisée par une densité α. Nous distinguerons les deux cas possibles $\alpha < \pi$ et $\alpha > \pi$.

1. $\alpha < \pi$ *:* Dans une région de basse densité, l'investigation autour de chaque point X nécessite 3^p appels de cellules, chaque appel dans une structure de pavage nécessitant $O(p)$ calculs, et dans chacune de ces cellules, le calcul des distances de tout point Y de la cellule au point X : sachant que la densité $\alpha < \pi$, le nombre de distance à partir du point X sera limité par π (valeur finie), tout calcul de distance nécessitant $O(p)$ opérations. Au total on a donc :

$$T(\text{Phase 2} | \alpha < \pi) < n(3^p(1 + \pi O(p)) \approx O(np3^p) \qquad (2)$$

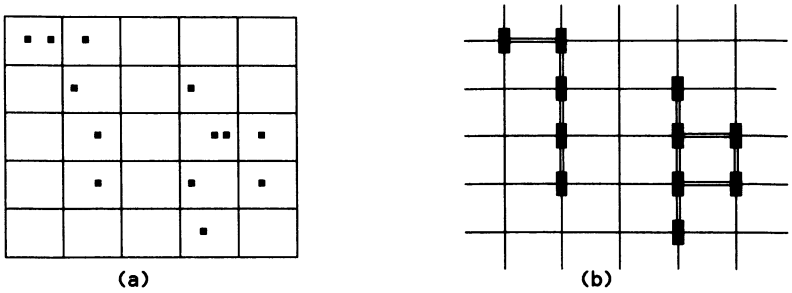

FIGURE 5

(a) visualise un pavage dans R^2. (b) visualise le graphe associé (M, Θ), les nœuds de M étant représentés par (■), et les arêtes de Θ par des doubles traits (‖) sur le réseau cubique représenté en simples traits (|). Dans ce cas, deux composantes connexes existent.

2. $\alpha > \pi$: A partir du pavage $\Omega(\sigma)$, considérons le graphe associé (M, Θ'). En vertu de la propriété 3, (M, Θ') étant un graphe défini sur réseau à maille cubique, et puisque la densité $\alpha > \pi$, le nombre moyen de nœuds I connectés est infini, en d'autres termes toutes les cellules C_I et par conséquent tous les points de la région ont déjà été annexés dans la première phase de l'algorithme. Le calcul des distances ne s'effectuera que dans les cellules du voisinage C_I associé au second pavage, qui contiennent des points n'appartenant pas à la même classe P_i que X. Considérant une structure de liste linéaire reliant les points situés dans une même cellule C_I, il est aisé de supprimer toutes les cellules situées dans la même classe de P (étape 1), et l'appel des cellules se soldera par un nombre moyen nul de points dont il faut calculer la distance à X. La complexité en temps moyen se ramène donc dans ce cas à $n \cdot 3^p$ appels de cellules, et d'un nombre moyen nul de distances, négligeable par rapport à n :

$$T(\text{phase } 2 | \alpha > \pi) = n \cdot 3^p O(p) \tag{3}$$

Sommant la complexité sur l'ensemble des régions de l'espace, et des résultats partiels (1, 2, 3), on conclut que l'algorithme est de complexité $O(np)$ en temps moyen, et donc linéaire en fonction de l'effectif de l'échantillon à classifier, en considérant une dimension p fixée.

Performances

Des temps de calcul ont été relevés sur des données artificiellement engendrées par des réalisations de mélanges de distributions gaussiennes dont les paramètres (vecteur des moyennes et matrice de variance-covariance) étaient elles-mêmes tirées aléatoirement. Les résultats confirment le comportement linéaire en fonction du nombre de points à classifier. La table 1 compare les performances de l'algorithme CC et l'algorithme $BF\text{-}MST$ de Bentley et Friedman[5] considéré comme l'algorithme le plus rapide en vue du calcul de l'ultramétrique inférieure maximale, via la recherche du MST.

TABLEAU 1

Gain en temps par rapport à la méthode élémentaire $O(n^2)$, comparé entre l'algorithme B-F-MST et l'algorithme CC proposé, $(p = 2, 3, 5)$.

	BF-MST	*CC*
$p = 2$		
$n = 10^3$	33	290
10^4	250	1790
10^5	2807	39385
$p = 3$		
$n = 10^3$	16	134
10^4	167	1345
10^5	1309	11605
$p = 5$		
$n = 10^3$	8	89
10^4	29	156
10^5	189	867

En vue de présenter des valeurs indépendantes du matériel informatique utilisé, les résultats sont donnés sous forme de gain par rapport à la méthode de base de recherche de l'arbre de longueur minimum programmée par Gower et Ross[3], considérée comme méthode étalon, et utilisée sur les mêmes données. La gain d'un algorithme est défini par son temps de calcul rapporté à celui de la méthode étalon, sur les mêmes données. Plusieurs essais ont été réalisés pour une même dimension et un même effectif. Les résultats moyens sont groupés pour plusieurs dimensions $(p = 2, 3, 5)$ et plusieurs effectifs $(n = 10^3, 10^4, 10^5)$. On remarque des gains particulièrement élevés pour des grands effectifs, et sur les dimensions faibles. Au delà de $p = 5$, le nombre de points doit dépasser 10^5 pour confirmer un gain appréciable. Il est clair que le nombre exponentiellement croissant des cellules (se matérialisant dans les expressions (3) et (4) par 3^p) tend à anihiler et même inverser l'économie de calcul réalisée sur les distances.

Conclusions

Un algorithme de classification a été proposé, jouissant conjointement d'une partition résultante exacte au sens du critère de l'écart, et d'une complexité asymptotique $O(n)$ en temps moyen. Cette complexité peut être considérée comme minimale : on voit mal comment descendre en dessous de $O(n)$. Cet algorithme est à notre connaissance le seul assurant une propriété d'optimalité, en l'occurence la maximisation de l'écart pour toutes les partitions en un même nombre de classes, tout en convergeant en un temps linéaire en fonction de l'effectif de

l'échantillon. Basé sur un principe géométrique de réduction de calculs de distances, cet algorithme est fortement influencé par la dimension de l'espace considéré et son gain ne sera réel que pour des dimensions inférieures à 5.

Références

[1] LEVINTHAL C. Molecular model building by computer, Scientific american, 214, pp. 42-52, (1966).

[2] ROSENFELD A. Picture Processing by computer, Academic Press, New York, (1969).

[3] GOWER J.C., ROSS J.S. Minimum Spanning tree and Single Linkage Clustering Analysis, Applied Statictics, 18, pp. 54-64, (1969).

[4] BENTLEY J., STANAT D. and WILLIAMS E.H. The complexity of finding fixed radius near neighbours, Inf. Proc. letters, 6.6, pp. 209-213, (1977).

[5] BENTLEY J. FRIEDMANN J.M. Fast Algorithms for constructing minimum spanning trees in coordinate spaces, I.E.E.E. Trans. on computers, Vol. C-27, pp. 97-104, (1978).

[6] LEHERT Ph. Clustering by Connected Components in $O(n)$ expected time, R.A.I.R.O. Computer Science, 28, (1981).

[7] ANDERBERG M.R. Cluster Analysis for Applications, New York, Academic Press, (1973).

[8] LEHERT Ph., Ultramétrique inférieure maximale et Complexité, Data Analysis and Informatics, Diday Ed., North Holland, (1985).

[9] HAMMERSLEY J.M. On rate of convergence to the connective constant of the hypercubical lattice, Quart. J .math. 2-12, p. 250-256 (1961).

[10] SANTALO L.A. Integral Geometry and Geometric probability, Encyclopedia of Mathematics and its applications, v. 1. Addison Wesley, Reading, MA. (1976).

[11] DAY W.H.E. Efficient algorithms for agglomerative hierarchical clustering methods, J. of Classification, 1, 7-24, 1984.

[12] KARCHAF I. Sur la complexité des algorithmes de classification ascendante hiérarchique, Les cahiers de l'analyse des données, XII, 195-197, 1987.

[13] ROHLF F.J. A probabilistic Minimum Spanning Tree Algorithm, Information Processing Letters, 7, 44-48 (1978).

[14] REH W., First Passage Percolation under weak moment conditions, J. App. Prob, 16, 750-763, (1979).