La classification des grands échantillons : proposition d'un algorithme rapide du type *single link*

par

Ph. LEHERT

Faculté Universitaire de Mons - BELGIQUE

INTRODUCTION

La classification est une technique bien connue d'analyse statistique consistant à mettre en évidence dans un ensemble E de n objets l'existence éventuelle de classes homogènes et séparées.

Cette technique est utilisée très fréquemment dans les sciences humaines et parmi celles-ci, l'économie (Fisher [6]), la psychologie (Hantaluoma [9]), l'anthropologie (Weiner [16]) ou encore la médecine (Bouckaert [2]).

De nombreux algorithmes ont été proposés : l'homogénéité ou la séparation d'une partition étant mesurée par un critère mathématique, un algorithme tendra à diminuer la valeur d'un critère particulier. Lorsque la partition résultante correspond à l'optimum global du critère, l'algorithme est dit exact.

Or, la recherche d'une partition optimale par de tels algorithmes s'avère exigeante en temps d'exécution : la complexité en temps des algorithmes exacts est proportionnelle à n^n, dans le cas le plus général; pour certains critères, cette complexité peut être ramenée à $0(n^3)$, $0(n^2 \log n)$ et même $0(n^2)$.

Dans ce travail, on s'intéressera au cas le plus fréquent dans lequel les n objets sont mesurés par rapport à p variables quantitatives : E peut donc être assimilé à un sous-ensemble discret de R^p. En pratique, des ensembles à classifier de 1000 objets ou plus ne sont pas rares, et l'application d'algorithmes exacts s'avère dans ces cas prohibitive vu le temps calcul nécessité.

Les seules approches utilisables pour les grands échantillons sont des méthodes approchées : parmi elles, on cite souvent les agrégations rapides (Hartigan [8]), ou encore les K-means (Mc Queen [11]), dont une variante est particulièrement utilisée en France sous le nom de Méthode des nuées dynamiques (Diday [4]).

Parmi les méthodes délivrant une partition optimale, la méthode dite ultramétrique inférieure maximale, mieux connue sous le nom de Single Linkage Clustering Analysis (SLCA) est caractérisée par une complexité la plus faible parmi l'ensemble de toutes les méthodes exactes connues : une très nombreuse littérature est consacrée à ce problème dont on trouvera les références principales dans Gower et Ross [7]. En substance, SLCA délivre une hiérarchie de partitions $p_1, \ldots, p_n$ telle que P_k réalise un écart[1] maximum parmi toutes les partitions de E en k classes (Delattre et Hansen [3]). La construction de la hiérarchie s'apparente à celle de l'arbre de longueur minimum (ALM) du graphe complet G(E, D) dont les noeuds sont les éléments de E et dont les arêtes entre tout couple (X, Y) de E^2 sont valorisées par $a(X, Y)$, a étant un indice de dissimilarité choisi. La complexité en temps minimale associée au problème général est $0(n^2)$, correspondant essentiellement au calcul de $a(Y, Y)$ pour tout couple de E^2. Cette complexité est obtenue par l'algorithme de Prim [12] dont

l'implantation informatique a été étudiée par Dykstra [5].

Or, l'application d'un algorithme $0(n^2)$ à des échantillons de taille élevée ($n > 1000$) s'avère largement trop coûteuse, si bien qu'on a recherché des algorithmes plus rapides, adaptés à des configurations particulières de données. Dans le cas bidimensionnel ($p = 2$) Shamos et Hoey [15] puis Hwang [10] proposent successivement un algorithme de complexité $0(n \log n)$ en utilisant le diagramme de Voronoï (Santalo [14]). Bentley et Friedman [1] définissent un algorithme $0(n \log n)$ en temps moyen pour des données dans RP, basé sur la structure de K-d Free. Enfin, Rohlf [13] propose un algorithme sur le même type de données basé sur les pavages de Rabin. Ces algorithmes voient leurs performances décroître sensiblement avec la dimensionalité p; de plus, ils ne sont pas à l'abri d'une dégénérescence en $0(n^2)$.

Dans la suite de cet article, nous définissons une nouvelle méthode de recherche de l'ALM et de la hiérarchie correspondante.

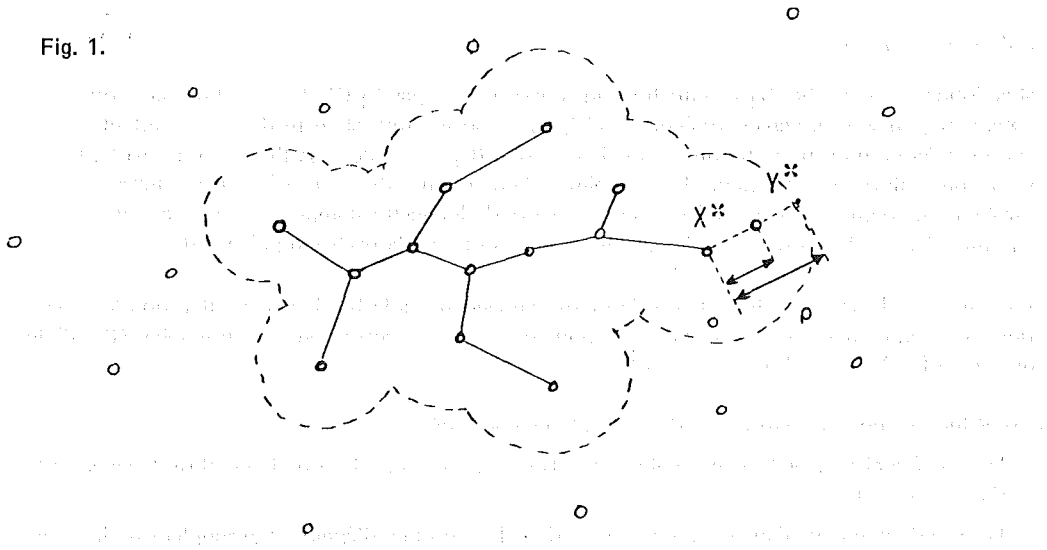
Dans le paragraphe suivant (2), les principes de la méthode sont évoqués. On définira dans un point suivant (3) l'algorithme correspondant ainsi que les structures de données nécessaires (4). Nous terminerons par une conclusion (5) relatant des essais expérimentaux.

1. PRINCIPE DE LA METHODE

Un des principes de base de Prim en vue de construire l'ALM s'énonce : un sous arbre $S(P_1, F)$ étant déjà construit, l'arête (X^*, Y^*) telle que $d(X^*, Y^*) = \min (X, Y)$ pour tout $X \in x$ et $Y \in E - P_1$ appartient à ALM. L'algorithme consiste donc au départ d'un point quelconque $X_1 = P_1$ et $F = \{\phi\}$ initialisant S, de rechercher à chaque itération (X^*, Y^*), d'ajouter (X^*, Y^*) à F et Y^* au fragment P_1.

De manière générale, la recherche de (X^*, Y^*) exige le calcul d(X, Y) en tout couple de points (X, Y) et présente donc une complexité $0(n^2)$. La figure (1) suggère pourtant que le calcul total est partiellement inutile : il suffirait de n'investiguer que les points de $E - P_1$ situés dans un voisinage de rayon ρ supérieur à $d(X^*, Y^*)$.

Fig. 1.



Considérons dès lors un seuil ρ, choisi de manière telle que $\rho > d(X^*, Y^*)$ (ce point sera examiné plus loin) et pour tout $X \in P_1$ calculons $d(X, Y)$ pour tout Y restreint à la boule $B(X, \rho)$, de centre X et de rayon ρ. L'ensemble des liens calculés est considérablement plus réduit que l'ensemble E^2 et est suffisant, puisque contenant le lien minimal (X^*, Y^*).

Précisons que la restriction à $B(X,)$ ne peut être réalisée que moyennant la construction de structures associatives spéciales, mentionnées dans le point 3 suivant. Leur connaissance n'est pas nécessaire pour la compréhension du principe.

Partant d'un point quelconque et se restreignant à $B(X, \rho)$ pour tout nouveau point centré, le sous-arbre croîtra tant qu'il est possible de mettre en évidence des arêtes contiguës de l'ALM telles que $d(X, Y) < \rho$. Considérant un vecteur D tel que D(Y) est la distance minimum $d(X, Y)$ pour tout $X \in P_1$, D est mis à jour à chaque itération en calculant $d(X, Y^*)$ pour tout Y dans $B(Y^*, \rho)$.

Si toute arête contiguë supplémentaire est supérieure à ρ, deux stratégies sont possibles :

1) pour chaque point X de S investiguer successivement les couronnes $\{B(x, k\rho) - B(x, (k-1)\rho\}$ jusqu'au moment où une arête de l'ALM de longueur inférieure à $k\rho$ soit découverte. Le seuil

utilisé sera alors $k\rho$.

2) abandonner le sous-arbre S_1 et initialiser un sous-arbre S_2, par X_l ($\notin P_1$) que l'on fera croître, comme S_1, en se restreignant toujours à $B(X, \rho)$. Lorsque tout élément de E a été affecté de cette manière à un sous-arbre, on a donc la partition $(P_1, \ldots, P_k)$. A l'itération suivante, la valeur du seuil passe à 2ρ: pour chaque point X, la couronne $\{B(X,2\rho) - B(X,\rho)\}$ est investiguée, permettant de découvrir les arêtes de l'ALM de longueurs comprises entre ρ et 2ρ; chacune d'entre elles fusionnant deux fragments P_1 et P_k de la partition précédente.

L'avantage de la seconde stratégie est de mettre en évidence à la fin de chaque itération les partitions dites en composantes connexes de E pour $\rho = 1, 2 \ldots$ correspondant au graphe G(E, V) tel que $V = \{(X, Y) \in E^2 / d(X, Y) < i\rho\}$.

La procédure est donc constituée de deux itérations imbriquées :

- la première élève la valeur du seuil de $(k - 1)\rho$ à $k\rho$ et s'arête lorsque le nombre de classes $|P|$ de P est unitaire.
- la seconde contenue dans la première investigue la couronne d'épaisseur ρ supplémentaire, pour tout point X : soit la partition Φ obtenue à l'itération (k - 1) précédente. Au début, chaque classe Ψ_i de Φ est constituée d'un point C et donc $|\Phi| = n$.

P_1 initialisée par un Ψ_1 quelconque, l'investigation des points de Ψ_i conduit à un ensemble Λ de liens de longueurs compris entre $(k - 1)\rho$ et $k\rho$, parmi lequel le lien minimal est choisi, fusionnant deux classes de Φ. Si Λ reste vide, une nouvelle classe de P est initialisée.

2. DESCRIPTION DE L'ALGORITHME

La procédure ALM représentée ci-dessous reproduit les éléments essentiels de l'algorithme. On y apportera quelques commentaires : Λ (ligne 2) est l'ensemble des liens initialisé à $\{\emptyset\}$; Λ s'accroît (l. 13) lorsque la distance d (X, Y) ($X \in P_m$, $y \notin P_m$) est inférieure à la meilleure distance déjà trouvée entre Y et P_m et contenue dans D (Y) (initialisée à chaque itération principale à ∞ en l.4) et D1 (Y) est l'autre extrémité de cette distance. Λ est amputé en l.14 du lien (Y, X) et en l.8 de tous les liens entre Ψ_l et P_m devenus caducs depuis la fusion de Ψ_l et P_m(l.23). L'élément minimum dans Λ doit être recherché (l. 20). Au départ (l. 2) chaque élément de X s'associe avec une classe de P. A chaque itération principale impliquant un changement de seuil, l'ancienne partition P correspondant à l'itération précédente est appelée $\Phi(\Psi_1, \ldots \Psi_t)$ (l.4). La deuxième itération principale opère tant que Φ ne devient pas vide. Or, celle-ci se voit amputée d'une classe Ψ_l (en ls 6 et 23) dès qu'une de ces dernières est affectée à P_m. En l. 19 dès que Λ est non vide, un Ψ_l est affecté à P_m (l. 22), sinon Ψ_l affecté (l. 18) à $\{\emptyset\}$, provoque la fin de l'itération (l. 7) et l'initialisation d'une nouvelle classe P_m. En l. 9, $V(B_r(X, k\rho) - P_m)$ signifie un sous-ensemble de R^p qui contient tout point de $B_r(X, k\rho)$ n'appartenant pas à P_m. On verra dans le point suivant une structure permettant de rendre ce sous-ensemble V le plus petit possible.

```
1       Procedure ALM(x,n,p);
2  debut Λ←{∅};P_i←X_i pour i=1,...,n;k←0
3     A : Répéter jusque |P|=1
4       debut k←k+1; m←0; Φ←P;D←∞;
5             B : tant que Φ ≠ {∅} faire
6             début   Soit ψ_1∈Φ; m←m+1; P_m←ψ_1; Φ←Φ←ψ_1
7                   C : tant que ψ_1≠∅ faire
8                       début si D1(X)∈P_m Alors Λ←Λ-{X,D1(X)}
9                             pour tout x∉ψ_1 faire
10                          début pour tout Y∈V[B_r(X,kρ)-P_m] faire
11                                début Si d(X,Y)<D(y) Alors
12                                      début D(Y)←d(Y,Y);X*←D1(Y);
13                                            D1(Y)←X    ;
14                                            Λ←Λ∪(X,Y)-(Y,X*)
15                                      fin
16                                fin
17                          fin
18                          ψ_1←∅;
19                          Si Λ≠∅ Alors Λ
20                          début Soit d(α,β)=(min d(X,Y))
21                                et soit β⊂ψ_1∉P_m
22                                P_m←P_m∪ψ_1;
23                                Φ←Φ-ψ_1
24                           fin
                      fin
                fin
          fin
    fin
```

Fig. 2 : Algorithme ALM

3.- STRUCTURES ADEQUATES POUR X ET Λ

3.1. PAVAGE P(ρ) SUR E

Comme souligné en 2, V $[B_r(x, k_\rho) - P_m]$ doit être minimisé. La meilleure structure dans ce sens est la réalisation d'un pavage linéaire sur E (Lehert [17]) consistant à découper R^p (ou du moins ses dimensions principales), en cellules cubiques C_I ($I = (i_1, \ldots, i_p)$) telles que

$$C_I = \left\{ X \in E \text{ tel que } \left\lfloor \frac{x_j}{\rho} \right\rfloor = i_j \right\}$$

Les élements de C_I sont reliés entre eux par une liste linéaire $L(C_I)$ ou rassemblés séquentiellement. C_I est accessible via une méthode de hachage délivrant une adresse de rangement initiale h(I) sur base de la clé I. Un pavage initial de côté ρ, P(ρ) étant construit sur E, on remarque facilement que si $x \in C_I$, $B(X, \rho) \subset A(I)$ tel que $A(I) = \left\{ C_j \text{ tel que } \max_{k = 1, \ldots, p} |i_k - j_k| \leqslant 1 \right\}$

De même, transformant P(ρ) en P(kρ) on a $B(X, k\rho) \subset A(I)$: de cette manière, construisant un pavage initial P(ρ) et le modifiant en P(kρ) à chaque itération principale (l. 5), on obtient B(X, kρ) en accès direct en appelant par hachage sur la clé l'ensemble A(I). De plus, la liste $L(C_I)$ sera amputée de tous les points rentrés dans un P_m (l. 6 et 22) puisque ceux-ci ne sont plus à considérer : cette opération se fera de manière particulièrement aisée dès le début du bloc (l. 9) jusque dans celui-ci, tout nouveau point de P_m est examiné. A(I) est plus grand que B(X, kρ) : pour un point qui appartient à A(I) - B(X, kρ), il est nécessaire de calculer sa distance d(X, Y) : néanmoins si $d < k\rho$, il n'est pas désirable de tenir compte de ce lien dans Λ : en effet, ce lien sera probablement caduc lors d'une itération suivante, alors qu'introduit dans Λ; il réclame deux mises à jour probablement en pure perte. Il vaut mieux négliger ce lien et le recalculer à l'itération suivante. De plus, la taille de Λ augmente ce qui, on le verra, en diminue les performances d'accès.

3.2. STRUCTURE SUR Λ

Λ ne contient donc à tout moment de l'itération k que des liens de longueur compris entre $(k-1)\rho$ et kρ. Concernant Λ, une structure est nécessaire en vue d'extraire le minimum et effectuer des ajoutes et des suppressions : une structure de heap tree (Aho et al [18]) s'applique particulièrement à ce problème : le minimum sera obtenu au sommet du heap. En l. 8 la suppression requiert un vecteur d'état Id tel que Id(X) = true si X a déjà été classé dans P (avec toute une classe Ψ_I). Id peut être mis à jour en même temps que les listes $L(C_I)$ en début de bloc (l. 9).

4.- CONCLUSIONS

Après la description de l'algorithme, il semble important d'aborder le calcul de sa complexité. Cette étude est en cours. A ce stade du travail, de multiples expériences ont été réalisées en vue de comparer les approches similaires de Bentley et Friedman [1] et de Rohlf [13] (notées respectivement BF etR).

Malgré que les essais sur les 3 méthodes aient été réalisés sur trois systèmes différents, il est possible de comparer leur temps moyen d'exécution par rapport au temps de résolution des mêmes données par l'algorithme de base de Prim, programmé par Dykstra [5] et choisis par Bentley et Friedman [1] et Rohlf [13] comme mesure unitaire.

Le tableau suivant donne les valeurs de n, pour chaque dimension m, à partir desquelles l'algorithme devient plus rapide que la version de base de Prim.

	BF	R	PR
2	250	150	122
3	260	200	149
4	340	350	210
5	445	-	380
6	645	-	734
7	920	-	-
8	1400	-	-

Or, la méthode BF est démontrée opérer en O(n log n) en temps moyen et cette complexité a été expérimentalement vérifiée.

Un comportement O(n log log n) a été expérimentalement observé par Rohlf pour des données uniformes ou multivariées gaussiennes. Quant à la méthode proposée ici, un ajustement statistique semble mettre en évidence un comportement empirique O(n log log n) très proche de O(n). Puisque dans le graphique donnant la complexité en ordonnée en fonction de n, le point (0, 0) est commun aux trois méthodes, les comportements asymptotiques O(n log n) et O(n log log n) des méthodes sont supposées connues, et enfin les points d'inter-sections avec les courbes de complexité de PRIM-Dykstra étant connues par le tableau précédent, on en déduit une supériorité des performances en temps moyen de notre méthode par rapport aux autres pour des dimensionalités $m \leqslant 5$; passée cette dimension, les résultats de BF sont plus performants (cf. fig. IV.4.2.).

Enonçons enfin les caractéristiques principales de l'algorithme mises en évidence expérimentalement : parmi les autres méthodes, ce dernier se caractérise par une plus grande rapidité, requérant (résultat expérimental) O(n log log n) opérations pour des données dans des espaces de faibles dimensionalités ($m < 5$). Le temps d'exécution est particulièrement faible lorsque la répartition des points dans les zones occupées est peu concentrée et lorsque la plus grande arête du MST à mettre en évidence est de longueur faible par rapport aux étendues du nuage de points F dans R^p. Le choix plus délicat du côté du pavage la rend probablement plus fluctuante sur le plan des performances que celle de Rohlf.

La méthode de Bentley et Friedman considérablement plus lente en moyenne pour les faibles dimensionalités, assure des performances moins fluctuantes, indépendantes de la concentration des points dans R^p.

NOTE

(1) L'écart d'une partition est la distance minimale entre deux objets situés dans deux classes différentes.

BIBLIOGRAPHIE

[1] BENTLEY, J.L. and FRIEDMAN, J.M., "Fast Algorithms for constructing minimal spanning trees in coordonate spaces". I.E.E.E. Trans. on Computers, Vol. C-27, 97-104.

[2] BOUCKAERT, A., "Computer diagnose of goiters : Classification and differential diagnoses". J. Chronic. Dis 24, pp. 299-310 (1971).

[3] DELATTRE, M. and HANSEN, P., "Bicriterion Cluster Analysis". I.E.E.E. Transactions on pattern analysis and machine intelligence, Vol. Pami-2, n. 4 (1980).

[4] DIDAY, E., "Optimisation en Classification Automatique et Reconnaissance des Formes". Revue Fr. Automatique, Infor. Rech. Oper., 6 (1972).

[5] DIJKSTRA, E.W., "A short introduction to the art of programming". Technolog. Université Einshoven, Rep. EWD 316, pp. 64-70 (1971).

[6] FISHER, W.D., "Clustering and Aggregation in Economics", Baltimore, John Hopkins Press (1968).

[7] GOWER, J.C. and ROSS, J.S., "Minimum spanning trees and single linkage cluster analysis". Applied statistics, 18, pp. 54-64 (1969).

[8] HARTIGAN, J.A., "Clustering algorithms". New York, Wiley (1975).

[9] HAUTALUOMA, J., "Syndromes, antecedents and outcomes of psychoses : a cluster analysis study". J. Connret. Clin. Psychol., 37, pp. 332-344 (1971).

[10] HWANG, F.K., "An 0(n log n) algorithm for rectilinear minimal spanning trees". JACM Vol. 26, 2, pp. 177-182 (1979).

[11] MAC QUEEN, J.B., "Some Methods for classification and Analysis of Multivariate observations" Proceedings of the fifth berkeley symposium on mathematical statistics and probability, 1, pp. 281-297 (1967).

[12] PRIM, R.C., "Shortest connection matrix network and some generalizations". Bell system techn. Journal, 36, pp. 1389-1401 (1957).

[13] ROHLF, F.J., "A probabilistic Minimum spanning tree algorithm". Inf. Proc. Letters, 8, pp. 44-49 (1978).

[14] SANTALO, L.A., "Integral geometry and geometric probability". Encyclopedia of mathematics and its applications, vol. 1, Addison Wesley (1976).

[15] SHAMOS, M.I. and HOEY, D., "Closest point problems". Proceedings 6th annual. I.E.E.E. symposium on fondations of computer science, pp. 151-162 (1975).

[16] WEINER, J.S. and HUIZINGER, J. Eds, "The assesment of population affinities" in man Oxford U.P. New York (1972).

[17] LEHERT, Ph. and DEVIJVER, P., "Clustering in 0(n) expected time by connected components". 5th international conference on pattern recognition (I.E.E.E.), Miami (1980).

[18] AHO, HOPCROFT and ULLMAN J., "The design and analysis of computer Algorithms". Wiley N.Y. (1974).