Outil de classification et de visualisation de grands volumes de données mixtes

Christophe CANDILLIER*, Noureddine MOUADDIB**

*Entreprise GÉOBS SA, 8 avenue des Thébaudières, 44800 SAINT-HERBLAIN
christophe.candillier@lina.univ-nantes.fr
http://c.candillier.free.fr/
**LINA (Laboratoire d'Informatique de Nantes Atlantique)
2 rue de la Houssinière, 44322 Nantes cedex 3
mouaddib@lina.univ-nantes.fr
http://www.sciences.univ-nantes.fr/lina/

Résumé. Nous avons conçu un outil de classification de données original que nous détaillons dans le présent article. Cet outil comporte un module de création de résumés et un module d'affichage. Le module de création de résumés prend en charge les données mixtes (qualitatives et quantitatives) ainsi que les grands volumes de données en utilisant une méthode de classification incrémentale et agglomérative originale. Le module de visualisation permet une lecture aisée des résumés grâce à une interface graphique évoluée permettant la présentation et l'exploration des résumés sous forme d'une hiérarchie de profils ou d'un tableau de profils. Chaque profil donne de manière claire les informations importantes relatives au résumé de données correspondant. La lecture de la hiérarchie et du tableau est aussi grandement facilitée par le choix d'un ordre optimal pour la présentation des variables et des résumés.

1 Introduction

Nous discuterons d'abord de l'algorithme de classification utilisé, de ses avantages et de ses inconvénients. Nous nous intéresserons ensuite à la visualisation des résumés produits, cela comprendra le calcul de l'ordre optimal des résumés et des variables ainsi que la visualisation sous la forme d'une hiérarchie des profils des résumés et sous la forme d'un tableau de profils. Finalement, nous illustrerons le fonctionnement de l'outil par l'analyse des données socioprofessionnelles de Paris et sa petite couronne.

2 Outil de Classification

2.1 Préliminaires

Les outils de classification sont divers et variés, ils ont pour but de regrouper les individus les plus semblables dans une même classe (Jain et al. 1999, Berkin 2002). Les deux principales familles sont les méthodes par partitionnement et les méthodes hiérarchiques. Les premières construisent directement les partitions et cherchent ensuite à les améliorer. Les dernières peuvent être scindées entre les méthodes par agglomération qui créent une

hiérarchie ascendante en procédant par regroupements successifs et les méthodes par division qui créent une hiérarchie descendante en procédant par divisions successives. A ces deux familles, ils convient d'ajouter des familles de méthodes historiquement plus récentes et qui se recoupent parfois avec les anciennes familles. Les plus connues sont les méthodes par densité qui groupent les objets situés dans des zones de fortes densités et les méthodes par grille qui appliquent une grille multi-niveaux (ou un maillage multi-niveaux) dans l'espace des données et travaillent sur les cellules créées par le maillage.

Dans notre cas, nous nous intéressons plutôt aux stratégies permettant la prise en compte des grands volumes de données. Ces stratégies sont principalement les suivantes.

L'échantillonnage a été et reste la méthode la plus utilisée. Le principe est simple : à partir de l'échantillon réalisé, on utilise la méthode de classification de son choix. Une fois les classes déterminées sur l'échantillon, le reste des données est lu en une seule fois, chaque individu étant affecté à la classe la plus proche. Il est utilisé par CLARA (Kaufman et Rousseeuw 1990), CLARANS (Ng et Han 1994), CURE (Guha et al. 1998).

Une autre stratégie est la création d'un arbre hiérarchique de classes de manière descendante et incrémentale. Chaque nouvel individu descend de la classe racine jusqu'à la classe la plus basse dans laquelle il sera incorporé. Lors de la descente, divers opérateurs de sélection, de création, de division et de fusion de classes peuvent être appliqués. C'est la méthode utilisée par COBWEB (Fisher 1987) et BIRCH (Zhang et al. 1996).

Une autre technique est la création de bulles de données (Data Bubbles, Breunig et al. 2001) qui sont des résumés de données. Chaque bulle possède un centre, ces derniers sont initialisés soit par les résumés issus de BIRCH soit par un échantillon. Les bulles de données sont ensuite « remplies » par les individus en les assignant aux bulles les plus proches. Certains individus trop distants des bulles de données restent non affectés.

Une autre technique est l'utilisation d'un graphe de voisinage qui permet l'emploi de méthodes de partition de graphe rapides telles CHAMELEON (Karypis et al. 1999). Cependant, le principal obstacle vient de la construction du graphe de voisinage qui est très coûteuse si le nombre de dimensions est supérieur à trois.

Dans le cadre des méthodes de classification basées sur la densité en deux dimensions, l'utilisation d'un R-tree permet l'indexation des objets spatiaux et la recherche rapide d'objets compris dans un certain rayon. C'est cet index spatial qu'utilisent DBSCAN (Ester et al. 1996) et OPTICS (Ankerst et al. 1999).

Une autre stratégie est la création d'intervalles. Chacune des dimensions est partagée en intervalles et chaque individu est ensuite affecté aux intervalles auxquels il appartient. Les algorithmes de classification travaillent alors à partir de ces intervalles. Cette méthode est utilisée par CLIQUE (Agrawal et al. 1998) et WAVECLUSTER (Sheikholeslami et al. 1998).

La création de cellules est une extension de la création d'intervalles. Chaque dimension étant partagée en intervalles, cela crée de fait un découpage de l'espace en cellules. Chaque individu est ensuite affecté à la cellule à laquelle il appartient. Les cellules voisines sont connectées et les cellules vides sont supprimées. Les algorithmes de classification travaillent alors à partir de ces cellules. Cette méthode est utilisée par DENCLUE (Hinneburg et Keim 1998) et STING (Wang et al. 1997). Il faut cependant noter que le nombre de cellules croit exponentiellement avec le nombre de dimensions. L'utilisation de cette stratégie est ainsi limitée à un faible nombre de dimensions.

Les stratégies décrites précédemment traitent les grands volumes de données de façons très diverses. Cependant il existe certaines catégories intéressantes comme la famille des

méthodes incrémentales. Celles-ci doivent réceptionner un flux de données avec l'interdiction de conserver en mémoire la totalité ou une part importante des données reçues. On peut résumer ce principe par le respect des contraintes suivantes : la lecture des données se fait en une seule passe, les individus sont pris en compte un par un, le processus peut à tout instant être stoppé et redémarré, un résultat temporaire est disponible à tout moment. Les méthodes incrémentales sont ainsi particulièrement adaptées aux entrepôts de données. En effet, la mise à jour de la classification s'effectue par l'ajout des nouvelles données, sans avoir à recommencer le traitement depuis le début. Parmi les algorithmes incrémentaux, on peut citer BIRCH et COBWEB déjà vus précédemment.

2.2 Méthode de la Classification Ascendante Approximative (CAA)

Pour mettre au point notre algorithme, nous sommes partis des contraintes que doit respecter un algorithme incrémental prenant en charge de grands volumes de données. Nous avons aussi défini que l'algorithme ne devait pas utiliser plus de *k* classes (ou résumés) tout en permettant la création de nouvelles classes pour les individus atypiques. Ainsi comme l'a défini (Charikar et al. 1997), le principal problème que doit résoudre notre algorithme incrémental à *k* classes maximum est le suivant : pour une classification en *k* classes existantes, l'arrivée d'un nouvel individu à classer se traduit soit par l'inclusion de l'individu dans une classe existante, soit par la création d'une nouvelle classe pour le contenir et la fusion de deux classes existantes afin de maintenir le nombre de classes égale à *k*. La nécessité de réaliser des fusions place notre algorithme dans la catégorie des algorithmes ascendants (agglomératifs) dont fait partie la Classification Ascendante Hiérarchique (CAH). En étudiant le fonctionnement de la CAH, nous sommes arrivés à la conclusion que l'algorithme de la CAH répond à notre problème moyennant quelques adaptations. La CAH ignore la distinction entre les individus et les classes car les individus sont considérés comme des classes ayant un seul individu. Ainsi, lors de l'ajout d'un nouvel individu, il n'y a plus de choix à faire : l'individu est incorporé dans une nouvelle classe, puis les deux classes les plus semblables sont fusionnées. Cette méthode à l'avantage de ne pas centrer le problème sur le nouvel individu arrivant, par contre la recherche des deux classes les plus semblables peut être coûteuse avec un algorithme naïf.

- Etape 1 : La liste des résumés est vide: $R = \varnothing$
- Etape 2 : Tant qu'il y un individu *o* à classer faire :
  - Créer une nouvelle classe *C* pour cet individu : $C \leftarrow \{o\}$
  - Ajouter cette classe à la liste de résumés : $R \leftarrow R \cup \{C\}$
  - Si il y a plus de *k* classes Alors
    - Trouver les deux classes C_x et C_y les plus semblables :
      $d(C_x, C_y) = \min(d(X, Y))$ avec $X \in R$ et $Y \in R$
    - Les fusionner en une nouvelle classe : $C_n \leftarrow C_x \cup C_y$
    - Ajouter la nouvelle classe et enlever les deux anciennes classes :
      $R \leftarrow R \cup \{C_n\} - \{C_x, C_y\}$
  - Sinon ne rien faire
- Fin Tant Que

FIG. 1 – Algorithme de la CAA

Nous définissons notre algorithme, la Classification Ascendante Approximative (CAA), comme indiqué (FIG. 1) avec pour paramètre k le nombre de résumés maximum et d la mesure de distance ou de dissimilarité choisie entre les classes. Comme il s'agit de résumer l'information, nous avons systématiquement utilisé la distance de Ward qui représente la perte d'inertie, celle-ci étant une bonne représentation de l'information. Ainsi en fusionnant les classes ayant la distance de Ward la plus faible, nous minimisons la perte d'information.

Une variante de cet algorithme, la Classification Ascendante Hiérarchique Approximative (CAHA), reprend la CAA en y ajoutant une ultime étape : lorsqu'il n'y a plus de nouveaux individus à ajouter, on continue à fusionner les classes les plus semblables jusqu'à obtenir la classe contenant tous les individus. Pendant cette opération, on mémorise la hiérarchie et les nœuds de cette hiérarchie. Cette dernière étape permet de construire l'arbre de classification des résumés. Il s'agit en fait d'une CAH classique réalisée sur les résumés avec l'utilisation de la même distance d.

Le choix du nombre de résumés k permet de paramétrer la précision de l'algorithme. En prenant k aussi grand ou plus grand que le nombre d'individus à traiter, la CAHA est équivalente à la CAH. En effet, durant l'étape 2, il n'y a aucune fusion et chaque résumé ne contient qu'un seul individu. Le choix de k offre donc une alternative entre un algorithme rapide et approximatif et un algorithme lent et précis. Pour fixer la valeur de k, nous sommes partis du principe qu'un utilisateur souhaite une analyse globale de la totalité des données en moins de 20 classes. Nous établissons donc le niveau de résumés à un niveau 5 fois supérieur en prenant $k=100$ résumés afin d'obtenir 20 classes acceptables à partir des résumés tout en gardant un traitement rapide.

Les informations nécessaires à l'analyse sont calculées à chaque fusion réalisée par l'algorithme. Pour chaque variable de chaque résumé nous avons les informations suivantes : la valeur minimum, la valeur maximum, la valeur moyenne et l'écart type. La valeur maximum s'obtient en prenant la valeur maximum des deux résumés à fusionner et la valeur minimum s'obtient de manière équivalente. La valeur moyenne et l'écart type se déduisent simplement de la somme des valeurs, de la somme des carrés des valeurs et du poids (ou du nombre d'individus) qui sont obtenus par les sommations des valeurs respectives des deux résumés à fusionner. D'autres informations telles que la médiane ou les quartiles ne sont malheureusement pas calculables rapidement car elles nécessitent de réordonner la totalité des valeurs et donc de stocker tous les individus en mémoire, ce que nous nous interdisons dans le cadre d'un algorithme prenant en charge de grands volumes de données.

La complexité de l'algorithme peut être appréhendée via la matrice des distances permettant de calculer la distance minimale entre les résumés. Cette matrice symétrique a une taille $k \times k$. Il est nécessaire de calculer k distances lors de l'ajout d'un nouveau résumé et encore k distances lors de la fusion. Par ailleurs, la recherche de la distance minimale nécessite le parcours de toute la matrice soit k^2 accès. De fait, ces étapes étant réalisées approximativement n fois durant l'algorithme, la complexité temporelle de l'algorithme est $o\left(n\left(2k + k^2\right)\right) = o\left(nk^2\right)$, la recherche de la distance minimum étant l'étape la plus coûteuse. La complexité temporelle de l'algorithme est donc bien linéaire par rapport au nombre n d'individus. Afin d'optimiser la recherche de la distance minimale, nous avons utilisé un index (arbre binaire équilibré) sur la matrice des distances avec la distance pour clef. Ainsi, la recherche de la distance minimale est très rapide et s'effectue en $o(\log(k))$ au lieu de $o\left(k^2\right)$. En contrepartie le coût d'insertion et de suppression des distances dans l'index est de $o(\log(k))$ au lieu de $o(1)$. Il y a k insertions lors de l'ajout et k insertions et $2k$ suppressions

lors de la fusion. La complexité finale est ainsi $o(n(4k\log(k)+\log(k))) = o(nk\log(k))$, les étapes les plus coûteuses étant les accès à l'index. La complexité finale est toutefois moindre suivant le nombre *k* de résumés. Lorsque *k=n*, la complexité temporelle est $o(n^3)$ pour l'algorithme standard et $o(n^2\log(n))$ pour l'algorithme standard optimisé.

Lorsque le nombre *k* de résumés utilisés est faible par rapport au nombre de classes « réelles », la CAA peut subir un fort effet de « dérive » qui est dû à l'ordre « mauvais » dans lequel les individus sont ajoutés. Cela se traduit à la fin de la CAA par le fait que certains individus sont plus proches d'un autre résumé que du résumé auquel ils appartiennent. Pour éliminer cette dérive, il faudrait introduire les individus les plus proches entre eux d'abord pour finir par les individus les plus éloignés. Malheureusement, cela nécessiterait de calculer la matrice des distances entre tous les individus, ce qui reviendrait finalement à réaliser une CAH sur la totalité des données. Cependant, une solution simple est l'utilisation des k-moyennes en post-traitement de la CAA qui permet de réduire rapidement le nombre d'individus mal classés tout en améliorant la qualité globale des résumés. Son principal inconvénient est que chaque itération des k-moyennes nécessite le parcours de la totalité des individus. Nous établirons les modalités de l'utilisation des k-moyennes dans la partie expérimentation.

2.3 Prétraitements nécessaires

Le premier prétraitement nécessaire est la disjonction des variables qualitatives. Il existe beaucoup de types de distances sur des données mixtes (Mazlack et Coppock 2002). Cependant, une pratique courante est de considérer la distance finale comme la somme d'une distance adaptée aux variables qualitatives et d'une distance adaptée aux variables quantitatives. Nous adoptons une démarche similaire en transformant par disjonction chaque variable qualitative en autant de variables quantitatives « filles » qu'il y a de modalités et en pondérant chacune des variables filles. Ainsi pour chaque individu, la valeur d'une variable fille représentant une modalité vaudra 1 si l'individu possède cette modalité et 0 sinon. La moyenne d'une variable fille dans un résumé correspond donc à la fraction d'individus possédant cette modalité et les valeurs minimum et maximum dans le résumé ne peuvent être que 0 ou 1. L'intérêt de cette méthode est d'intégrer de façon très transparente les données qualitatives tout en utilisant des méthodes uniquement valables pour les données quantitatives. Cependant, afin que les variables ayant beaucoup de modalités ne soient pas avantagées par rapport aux autres variables, nous pondérons chacune des *m* variables filles issues d'une même variable qualitative par *1/m* pour que le poids global de ses variables filles fasse 1 et soit donc égal au poids d'une seule variable.

Le second prétraitement est la normalisation des données qui consiste simplement à centrer-réduire toutes les données (y compris les variables filles créées à partir des variables qualitatives). Pour cela, nous devons calculer la moyenne et l'écart type de chaque variable. Cette opération est réalisée préalablement en une seule lecture de la totalité des données. Une solution alternative est de calculer ces informations sur un échantillon. Dans tous les cas, la moyenne et l'écart type de chaque variable sont indiqués à la CAA qui réalise la normalisation des individus lors de leur lecture.

2.4 Expérimentations et comparaison avec les k-moyennes

Pour mesurer les performances de la CAA, nous l'avons comparée avec l'algorithme standard des k-moyennes. Nous avons pris comme indicateur l'inertie perdue par les résumés (FIG. 2). D'après les tests effectués sur des données réelles, la CAA peut se comporter moins bien que les k-moyennes lorsque le nombre de résumés est trop faible (notamment à cause d'un fort effet de dérive). Par contre, lorsque l'on augmente suffisamment le nombre de résumés demandés, la CAA devient alors meilleure car d'une part, les k-moyennes sont davantage pénalisées par un mauvais choix des partitions de départ (les *k* premiers individus soumis à l'algorithme) tandis que la CAA s'en affranchit rapidement par le mécanisme des fusions et d'autre part, l'effet de dérive s'atténue. Une autre alternative pour corriger l'effet de dérive est l'utilisation des k-moyennes pendant un très faible nombre d'itérations (entre une et trois itérations) sur les résumés trouvés par la CAA. Sur les zones IRIS (Ilots Regroupés pour l'Information Statistique) de Paris et sa petite couronne pour 27 variables socioprofessionnelles, la CAA seule est meilleure que les k-moyennes à partir de *k=100.* Sur les données simulées, la CAA est par contre bien meilleure car l'effet de dérive est très faible et les k-moyennes sont très pénalisées par un mauvais choix des partitions de départ.

Données	Nb de classes réelles	Nb de résumés	Inertie totale	CAA	k-moy 5 itérs	CAA + k-moy 5 itérs	k-moy 20 itérs	CAA + k-moy 20 itérs
Paris PC	Inconnu	20	73953	33255	31885	28787	27366	27113
Paris PC	Inconnu	50	73953	23111	25282	20201	21087	19552
Paris PC	Inconnu	100	73953	16350	20920	14787	17187	14464
Paris PC	Inconnu	200	73953	10935	16580	10278	14908	10218
Paris PC	Inconnu	400	73953	7225	12144	6980	11301	6965
Simulées	50	100	500000	11623	73397	11595	73373	11592
Simulées	400	100	500000	281071	374464	280757	360495	280757
Simulées	10000	100	500000	407919	427235	404225	416338	403617

FIG. 2 - Comparaison des qualités des résumés

3 Outil de visualisation

3.1 Méthode d'optimisation de l'ordre des variables et des résumés

L'optimisation des variables et l'optimisation des résumés sont des problèmes similaires. Dans la suite, nous utiliserons le terme « individu » pour désigner les résumés ou les variables selon le problème d'optimisation concerné. Différentes heuristiques plus ou moins efficaces permettent de trouver un ordre assez bon. La plus basique est le tri en fonction de la moyenne des valeurs normalisées d'un individu (Eisen et al. 1998) dont l'intérêt est d'être très simple et rapide, mais elle n'est efficace que si les variables sont toutes corrélées positivement. Une autre méthode est le tri selon le premier axe de projection d'une Analyse en Composantes Principales. Cependant, cette méthode n'est efficace que si les variables sont toutes corrélées. Une autre façon d'aborder le problème est de le définir de la façon suivante : parmi tous les ordres possibles, l'ordre optimal est celui qui minimise la somme des distances entre les individus consécutifs. Ce problème d'ordonnancement (Belloni et Lucena 2001) est connu sous le nom de Sequential Ordering Problem (SOP) ou Linear

Ordering Problem (LOP) qui est une variante du problème du voyageur de commerce, Traveling Salesman Problem (TSP). Ce problème étant NP-complet, la recherche de la solution exacte nécessite l'évaluation des *n!* ordres différents des *n* individus, ce qui n'est pas réalisable pour *n* supérieur à 20. On utilise alors des heuristiques. Toutefois, ce problème peut être simplifié par la création d'une hiérarchie binaire sur les individus. Dans ce cas, il existe des algorithmes capables de résoudre le problème de manière exacte (Bar Joseph et al. 2002) avec une complexité de $o(n^3)$ en temps et $o(n^2)$ en espace.

Ainsi, pour trouver l'ordre optimal des résumés, nous utilisons directement cet algorithme sur la hiérarchie des résumés en utilisant la même distance que dans l'outil de classification. Pour trouver l'ordre optimal des variables, il serait naturel de réaliser la classification ascendante hiérarchique conjointe pour créer une hiérarchie sur les variables. Cependant, cela est impossible pour les grands volumes de données. Nous utilisons donc à la place la matrice de corrélation des variables, beaucoup plus compacte. Le calcul de la matrice de corrélation peut se faire en une seule lecture de la base des données normalisées. Nous calculons donc la matrice de corrélation au cours de la classification des données au fur et à mesure de l'arrivée des individus lors de la CAA. Nous réalisons ensuite une CAH sur la matrice de corrélation en utilisant la distance de Ward. Puis nous appliquons l'algorithme de recherche de l'ordre optimal sur la hiérarchie des variables.

3.3 Visualisation de la hiérarchie de résumés

L'une des représentations les plus utilisées pour décrire une variable quantitative est l'histogramme des fréquences mais il a l'inconvénient d'être plutôt volumineux. Une représentation plus synthétique est la boîte à moustaches qui donne sur un seul axe les informations telles que les valeurs minimale, maximale et médiane et les quartiles. D'autre part, pour décrire plusieurs variables sur un même graphique pour une même information, il est possible d'utiliser la représentation en coordonnées parallèles (Siirtola 2000, Yang et al. 2003). Celle-ci fait correspondre à chaque variable un axe vertical sur lequel est représenté l'information par un point. La ligne reliant les points de proche en proche constitue la représentation de cette information pour toutes les variables dans le graphique. Cependant, pour des raisons de lisibilité, les variables doivent être similaires ou à défaut, être normalisées. En combinant la boîte à moustaches et la représentation en coordonnées parallèles, il est ainsi possible de visualiser simultanément les informations de toutes les variables.

Nous avons fait évoluer cette dernière représentation pour la rendre plus simple et plus accessible en supprimant toutes les informations chiffrées du graphique, le résultat final étant le profil de résumé (*FIG.* 3). Pour cela, nous avons défini pour chaque axe une origine, la moyenne générale de la variable (*MG*), et une échelle, l'écart type général de la variable (σG). Ainsi, chaque information sur l'axe s'exprime de manière centrée-réduite et une valeur X représentant une information quelconque (moyenne, maximum, minimum, médiane…) a sa position sur l'axe $P(X) = \frac{X - MG}{\sigma G}$. Pour la lecture, nous avons réciproquement $X = (P(X) \times \sigma G) + MG$. Dans une légende globale, les valeurs de *MG* et σG sont indiquées pour chaque variable. Dans le profil, les informations suivantes sont représentées pour chaque variable : la moyenne dans le résumé (représentée par le point de couleur), l'écart type dans le résumé (représenté par les barres verticales de part et d'autre du point de

couleur), le minimum dans le résumé (représenté par la délimitation inférieure de la zone grisée), le maximum dans le résumé (représenté par la délimitation supérieure de la zone grisée).

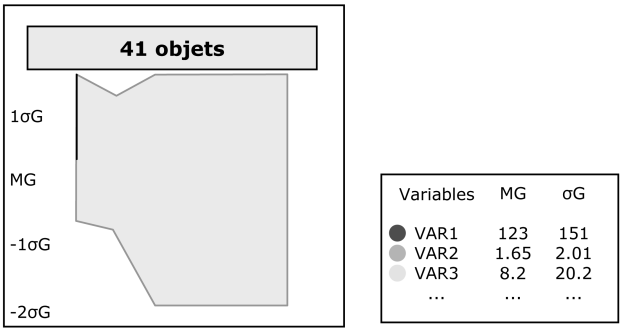

FIG. *3 - Profil d'un résumé et la légende*

Finalement, on représente la hiérarchie de manière incomplète (*FIG.* 4), seuls les noeuds les plus élevés étant représentés par leur profil intégré au sein de la hiérarchie. Par ailleurs, afin d'assurer une lecture globale rapide de la hiérarchie, il est créé un regroupement intermédiaire auquel est associée une couleur. Les classes appartenant à un même groupe sont alors associées à une variation de cette couleur. Nous disposons ainsi de deux niveaux de lecture pouvant être cartographiés si les données traitées ont une composante géographique.

3.2 Visualisation des classes de résumés sous forme de tableau

Un inconvénient de la représentation précédente est qu'elle ne permet pas la comparaison aisée des valeurs d'une même variable au sein des différents résumés alors que c'est une étape importante de l'analyse effectuée par l'utilisateur. Par contre, cela est permis par le tableau des profils de classes qui est historiquement la représentation la plus utilisée pour la visualisation des classes sous forme textuelle (Jambu 1999). Pour chaque classe et pour chaque variable, la case correspondante dans le tableau indique la valeur moyenne dans la classe. Cela permet une lecture dans les deux sens : une lecture horizontale des profils des classes et une lecture verticale des profils des variables. Une variante de ce tableau sous forme graphique est le résultat associé à la double classification des données (Eisen et al. 1998). Dans chaque case, la moyenne est représentée par une couleur allant du rouge (valeur positive maximum) au vert (valeur négative maximum) en passant par le noir (zéro qui est la moyenne de chaque variable normalisée). La lecture du tableau est alors immédiate. Un autre tableau adapté à la lecture des classes de données mixtes est celui utilisé pour la représentation des objets symboliques (Bock et Diday 1999), l'intervalle des valeurs est indiqué pour les variables quantitatives tandis que la fréquence de chaque modalité est représentée pour les variables qualitatives.

Nous avons repris l'idée de base du tableau pour les variables quantitatives (*FIG.* 5) tout en augmentant sensiblement les informations disponibles dans chaque case et permettant

deux niveaux de lecture. Le premier niveau est une lecture graphique globale et rapide permettant de connaître la moyenne de la variable pour la classe et sa représentativité. La moyenne est codée sous forme symbolique : un cercle rouge plein si la moyenne est plus de *1,5* écarts types généraux au dessus de la moyenne générale, un cercle rouge vide entre *0,5* et *1,5*, un cercle blanc entre *-0,5* et *0,5*, un cercle bleu vide entre *-0,5* et *-1,5* et un cercle bleu plein en dessous de *-1,5*. La taille de l'intervalle (écart entre la valeur minimum et maximum) est codée par les symboles suivants : rien, la valeur moyenne de la classe est considérée comme représentative car la taille de l'intervalle des valeurs est inférieure ou égale à *0,5* écarts types généraux, « <> », elle est assez représentative (entre *0,5* et *1,5*) et « !! », elle est peu représentative (plus de *1,5*). Le deuxième niveau de lecture donne les informations chiffrées de la moyenne, du minimum et du maximum de la variable dans la classe.

4 Application à l'analyse des catégories socioprofessionnelles de Paris et sa petite couronne

Les données sont les 2800 zones IRIS de Paris (département 75) et sa petite couronne (départements 92, 93, 94). Elles sont décrites par des variables quantitatives (le pourcentage de population dans chaque catégorie socioprofessionnelle) et une variable qualitative TYPE indiquant la nature de la zone (H=habitat, A=administratif et D=divers). La CAHA a été réalisée avec *k*=100 résumés et chaque IRIS a été pondéré initialement par la population y résidant. En choisissant le regroupement des résumés en 5 classes, nous obtenons les résultats suivants (*FIG.* 4, *FIG.* 5, *FIG.* 6), le nombre d'objets étant la population résidante.

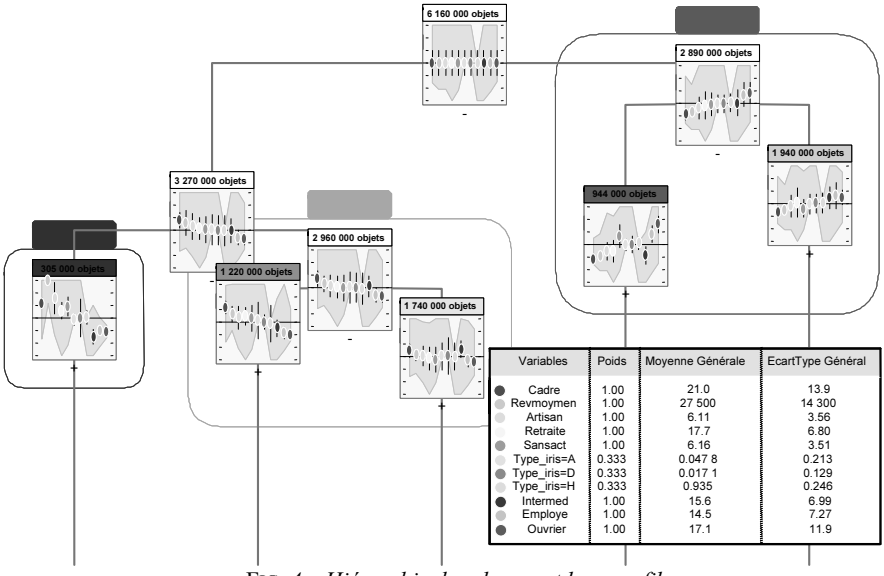

FIG. *4 – Hiérarchie des classes et leur profil*

Cette classification met en évidence deux types d'IRIS : ceux des catégories aisées (en vert) à très aisées (en bleu) et ceux des catégories moins aisées (en rouge). On remarquera aussi que pour chaque variable et chaque classe la moyenne est peu représentative du fait de la taille relativement importante des intervalles des valeurs. Les conclusions sur une classe ne seront donc valables que « en général » et non applicables à chaque IRIS de cette classe. L'interprétation de la classe en rouge foncé est la suivante : les IRIS de cette classe comportent généralement plus d'ouvriers, d'employés et de personnes sans activité que la moyenne, moins de cadres et d'intermédiaires que la moyenne et les revenus sont plutôt faibles. Géographiquement, les IRIS de cette classe sont surtout présents dans la zone au nord de Paris et de façon plus dispersée au sud. On remarque aussi que le type d'IRIS (Administratif, Divers ou Habitat) n'est ici pas discriminant entre les classes.

	Cadre	Revmoyen	Artisan	Retraite	Sansact	Type_iris=A	Type_iris=D	Type_iris=H	Intermed	Employe	Ouvrier
	MG=21.1 EC=13.9	MG=27 500 EC=14 300	MG=6.11 EC=3.56	MG=17.7 EC=6.80	MG=6.16 EC=3.51	MG=0.047 8 EC=0.213	MG=0.017 1 EC=0.129	MG=0.935 EC=0.246	MG=15.6 EC=6.99	MG=14.5 EC=7.27	MG=17.1 EC=11.9
Classe 1 305 000 objets	78.5 37.2 0.00	131 000 88 900 43 100	50.0 10.9 2.25	50.0 21.0 0.00	16.8 9.05 0.00	1.00 0.002 94 0.00	1.00 0.001 34 0.00	1.00 0.995 0.00	21.0 8.69 0.00	15.9 8.15 0.00	11.4 4.79 0.00
Classe 2 1 220 000 objets	60.0 36.4 0.00	73 200 35 800 18 700	18.9 7.21 0.00	40.0 20.2 0.00	18.1 7.23 0.00	1.00 0.026 3 0.00	1.00 0.000 204 0.00	1.00 0.973 0.00	25.0 13.5 0.00	100 9.09 0.00	30.2 6.17 0.00
Classe 3 1 740 000 objets	100 27.4 0.00	116 000 28 600 9 330	14.3 6.24 0.00	31.9 17.1 0.00	19.0 5.50 0.00	1.00 0.013 6 0.00	1.00 0.001 24 0.00	1.00 0.984 0.00	39.3 18.7 0.00	33.3 12.8 0.00	42.8 12.0 0.00
Classe 4 944 000 objets	27.9 4.90 0.00	38 300 14 500 0.00	13.4 4.45 0.00	29.5 15.0 0.00	75.0 8.15 0.00	1.00 0.002 76 0.00	1.00 0.000 254 0.00	1.00 0.997 0.00	20.7 11.4 0.00	35.2 20.0 0.00	85.7 35.8 0.00
Classe 5 1 940 000 objets	50.0 11.8 0.00	47 100 20 500 0.00	42.8 5.42 0.00	100 18.0 0.00	13.8 4.99 0.00	1.00 0.007 86 0.00	1.00 0.000 335 0.00	1.00 0.991 0.00	100 18.1 0.00	81.7 18.8 0.00	46.3 22.6 0.00

: très en-dessous de la moyenne : en- : proche de la moyenne : au- : très au-dessus de la moyenne

" " : moyenne représentative

FIG. *5 – Tableau des profils de classes*

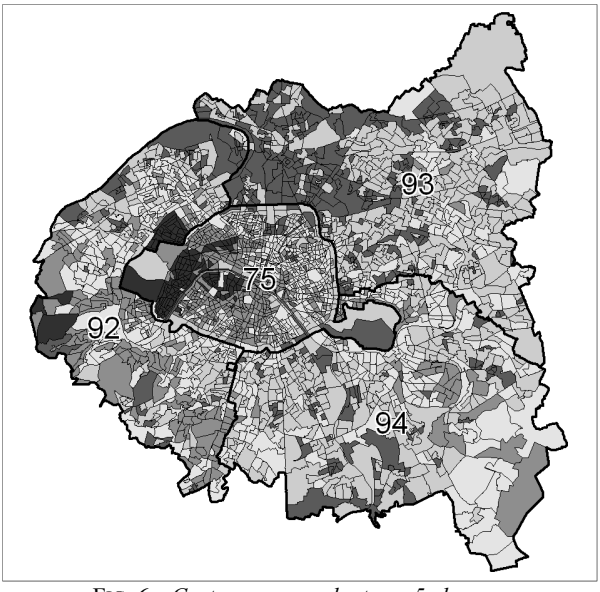

FIG. *6 – Carte correspondant aux 5 classes*

5 Conclusion

Notre outil est robuste et nécessite très peu de paramètres. Son principal inconvénient actuel est le paramétrage de la distance via des pondérations qui ne sont pas forcément adaptées. C'est pourquoi nous envisageons de remplacer l'utilisation de la distance euclidienne pondérée par la distance de Mahalanobis dans le calcul de la distance de Ward. Un autre point problématique est le fait que l'ordre optimal des résumés n'entraîne pas forcément un ordre optimal des classes construites par la hiérarchie. Nous étudions donc la possibilité de calculer l'ordre des classes de la hiérarchie durant son exploration.

Références

Agrawal R., Gehrke J., Gunopulos D., Raghavan P. (1998), Automatic subspace clustering of high dimensional data for data mining applications. In ACM SIGMOD International Conference on Management of Data, pages 94-105, 1998.

Ankerst M., Breunig M., Kriegel, Sander J. (1999), OPTICS: Ordering points to identify the clustering structure. ACM SIGMOD International Conference on Management of Data, 28(2):49-60, Juin 1999.

Bar-Joseph Z., Demaine D., Gifford D., Hamel A., Jaakkola T., Srebro (2002), K-ary clustering with optimal leaf ordering for gene expression data. In International Workshop on Algorithms in Bioinformatics, WABI, 2002.

Belloni A., Lucena A. (2001), Lagrangian Based Heuristics for the Linear Ordering Problem. In Metaheuristics International Conference, MIC, 16-20 juillet 2001.

Berkhin P. (2002), Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

Bock H. H., Diday E. (2000), Analysis of Symbolic Data. Springer, 2000. ISBN 3-540-66619-2.

Breunig M., Kriegel H., Kröger P., Sander J. (2001), Data bubbles: quality preserving performance boosting for hierarchical clustering. SIGMOD Record (ACM Special Interest Group on Management of Data), 30(2):79-90, 2001.

Charikar M., Chekuri C., Feder T., Motwani R. (1997), Incremental clustering and dynamic information retrieval. In 29th Symposium on Theory of Computing, pages 626-635, 1997.

Eisen M. B., Spellman P. T., Brown P. O., Botstein D. (1998), Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences USA, 95(25):14863-14868, Décembre 1998.

Ester M., Kriegel H. P., Sander J., Xu X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise. In 2nd International Conference on Knowledge Discovery and Data Mining KDD, page 226-231, 1996.

Fisher D. (1987), Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.

Guha S., Rastogi R., Shim K. (1998), CURE: an efficient clustering algorithm for large databases. In ACM SIGMOD International Conference on Management of Data, pages 73-84, 1998.

Hinneburg A., Keim D. (1998), An efficient approach to clustering in large multimedia databases with noise. In Knowledge Discovery and Data Mining, KDD, pages 58-65, 1998.

Jain A. K., Murty M. N., Flynn P. J. (1999), Data clustering: a review. ACM Computing Surveys, 31(3):264-323, 1999.

Jambu M. (1999), Méthodes de base de l'analyse de données. Eyrolles, 1999. ISBN 2-212-05256-1.

Karypis G., Han E., Kumar V. (1999), Chameleon: Hierarchical clustering using dynamic modeling. IEEE Computer: Special Issue on Data Analysis and Mining, 32(8):68-75, 1999.

Kaufman L., Rousseeuw P. J. (1990), Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, New York, 1990.

Mazlack L., Coppock S. (2002), Using soft computing techniques to integrate multiple kinds of attributes in data mining. In 5th International FLINS Conference, Computational Intelligent Systems for Applied Research, pages 137-144, Septembre 2002.

Ng R. T., Han J. (1994), Efficient and effective clustering methods for spatial data mining. In 20th International Conference on Very Large Data Bases, VLDB, pages 144-155, Septembre 1994.

Sheikholeslami G., Chatterjee S., Zhang A. (1998), WaveCluster: A multi-resolution clustering approach for very large spatial databases. In 24th International Conference on Very Large Data Bases, VLDB, pages 428-439, 1998.

Siirtola H. (2000), Direct manipulation of parallel coordinates. In Conference on Human Factors in Computing Systems,ACM CHI, volume 2 of Interactive posters, pages 119-120, 2000.

Wang W., Yang J., and Muntz R. STING: A statistical information grid approach to spatial data mining. In 23th International Conference on Very Large Data Bases, VLDB, pages 186-195, Athens, Greece, 1997.

Yang J., Ward M. O., Rundensteiner E. A. (2003), Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. Computers and Graphics, 27(2):265-283, Avril 2003.

Zhang T., Ramakrishnan R., Livny M. (1996), BIRCH: an efficient data clustering method for very large databases. In ACM SIGMOD International Conference on Management of Data, pages 103-114, 1996.

Summary

We conceived an original data clustering tool that we present in the following article. This tool comprises a summary module and a display module. The summary module handles mixed data (both qualitative and quantitative) and large quantities of data by using an original incremental and agglomerative clustering algorithm. The display module allows users to understand easily summaries thanks to its evolved graphic design. Summaries are either described through a hierarchy of profiles or a table of profiles. Each profile gives clear information about its corresponding summary. Readings of the hierarchy or the table are eased by the choice of an optimal order for both summaries and variables.