Identification de blocs homogènes sur des données continues

François-Xavier Jollois, Mohamed Nadif

LITA - IUT de METZ, Université de Metz,
Ile du Saulcy, 57045 METZ Cedex, France
{jollois,nadif}@iut.univ-metz.fr,

Résumé. Contrairement aux méthodes usuelles de classification ne cherchant généralement qu'une seule partition, soit des instances, soit des attributs, les méthodes de classification croisée et de classification directe fournissent des blocs de données liant des instances à des attributs. Les premières consistent à chercher simultanément une partition en lignes et une partition en colonnes. Les secondes, elles, s'appliquent directement sur les données, et permettent d'obtenir des blocs de données homogènes de toutes tailles, ainsi que des hiérarchies de classes en lignes et en colonnes. Combinant les avantages des deux méthodes, nous présentons ici une méthodologie permettant de travailler sur de grandes bases de données.

1 Introduction

Lorsque le but d'une classification est d'obtenir une structure en blocs homogènes, nombre d'utilisateurs appliquent généralement des algorithmes de classification simple sur les instances et sur les attributs séparément, les blocs résultent du croisement des partitions obtenues. Une telle méthode ne permet pas d'expliquer la relation spécifique pouvant exister entre un groupe d'instances et un groupe d'attributs. Ainsi, il est préférable d'appliquer des algorithmes de classification croisée, tel que l'algorithme *Croeuc* [Govaert, 1983, Govaert, 1995]. Celui-ci cherche simultanément une partition en lignes et une partition en colonnes, dont les centres permettent de synthétiser les données sous forme d'une matrice d'information de taille réduite. Une deuxième façon d'aborder le problème de la classification simultanée est d'utiliser un algorithme de classification directe, comme *Two-way splitting* [Hartigan, 1975], qui cherche à obtenir des blocs de données homogènes et de toutes tailles. On peut aussi citer les travaux de Marcotorchino [Marcotorchino, 1987] sur la sériation.

Malgré sa rapidité et son efficacité de traiter des tables de grande taille, l'algorithme *Croeuc* présente un défaut majeur ; il nécessite la connaissance des nombres de classes en lignes et en colonnes. Par contre, l'algorithme *Two-way splitting* s'affranchit de cette hypothèse mais sa complexité rend son utilisation impossible sur des données de grande taille. Nous présentons donc ici une combinaison de ces deux algorithmes afin de pallier les inconvénients de chacun.

Dans un premier temps, nous décrivons l'algorithme de classification croisée *Croeuc*. Ensuite, nous présentons l'algorithme *Two-way splitting*. Puis, nous décrivons la combinaison de ces deux algorithmes, et nous illustrons cette démarche par une application sur des données simulées. Enfin, nous concluons sur l'intérêt de cette méthode, ainsi

que sur les travaux futurs à réaliser.

2 L'algorithme *Croeuc*

Dans la suite, la matrice des données est définie par $\mathbf{x} = \{(x_i^j); i \in I \text{ et } \in J\}$, où I est l'ensemble des n objets (lignes, observations), et J est l'ensemble des d attributs (colonnes, attributs). On cherche à optimiser un critère $W(\mathbf{z},\mathbf{w},\mathbf{g})$, où $\mathbf{z} = (z_1,\ldots,z_s)$ est une partition de I en s classes, $\mathbf{w} = (w_1,\ldots,w_m)$ est une partition de J en m classes et $\mathbf{g} = (g_k^\ell)$ est une matrice $s \times m$, qui peut être vue comme un résumé de la matrice de données $\mathbf{x}$. Une définition plus précise de ce résumé et du critère W dépendra de la nature des données. La recherche des partitions optimales $\mathbf{z}$ et $\mathbf{w}$ est effectuée en utilisant un algorithme itératif. Govaert (1983, 1995) a proposé quelques algorithmes qui réalisent une classification croisée sur des tables de contingence, et plus généralement sur des tableaux de données binaires, continues ou catégorielles. Parmi tous les algorithmes, nous avons choisi *Croeuc* qui a été développé pour des données continues. En prenant la somme des distances euclidiennes au carré comme une mesure de la déviation entre la matrice $\mathbf{x}$, et la structure décrite par $\mathbf{z}$, $\mathbf{w}$ et $\mathbf{g}$, le problème est de trouver une paire de partition $(\mathbf{z},\mathbf{w})$ et le paramètre $\mathbf{g}$ correspondant, tel que le critère suivant soit minimisé :

$$W(\mathbf{z},\mathbf{w},\mathbf{g}) = \sum_{k=1}^{s}\sum_{\ell=1}^{m}\sum_{i\in z_k}\sum_{j\in w_\ell}(x_i^j - g_k^\ell)^2, \tag{1}$$

où g_k^ℓ est le centre du bloc $\mathbf{x}_k^\ell$. Il est facile de voir que pour $\mathbf{z}$ et $\mathbf{w}$ fixés, la valeur optimale de g_k^ℓ est donnée par la moyenne de tous les x_i^j du bloc $\mathbf{x}_k^\ell$. Les différentes étapes de *Croeuc* sont :

1. Démarrer avec des paramètres initiaux qui peuvent être choisis au hasard ($\mathbf{z}^0$, $\mathbf{w}^0$, $\mathbf{g}^0$).
2. Calculer ($\mathbf{z}^{(q+1)}$, $\mathbf{w}^{(q+1)}$, $\mathbf{g}^{(q+1)}$) en démarrant de ($\mathbf{z}^{(q)}$, $\mathbf{w}^{(q)}$, $\mathbf{g}^{(q)}$):
   (a) Calculer ($\mathbf{z}^{(q+1)}$, $\mathbf{w}^{(q)}$, $\mathbf{g}'$) à partir de ($\mathbf{z}^{(q)}$, $\mathbf{w}^{(q)}$, $\mathbf{g}^{(q)}$).
   (b) Calculer ($\mathbf{z}^{(q+1)}$, $\mathbf{w}^{(q+1)}$, $\mathbf{g}^{(q+1)}$) en démarrant de ($\mathbf{z}^{(q+1)}$, $\mathbf{w}^{(q)}$, $\mathbf{g}'$).
3. Répéter l'étape 2 jusqu'à la convergence

Dans les étapes 2(a) et 2(b), pour trouver $\mathbf{z}^{q+1}$ et $\mathbf{w}^{q+1}$ optimaux, nous cherchons à minimiser alternativement les deux critères suivants

$$\begin{cases} W(\mathbf{z},\mathbf{g}/\mathbf{w}) = \displaystyle\sum_{k=1}^{s}\sum_{i\in z_k}\sum_{\ell=1}^{m}\#w_\ell(u_i^\ell - g_k^\ell)^2 \quad \text{avec} \quad u_i^\ell = \frac{\sum_{j\in w_\ell} x_i^j}{\#w_\ell} \\ W(\mathbf{w},\mathbf{g}/\mathbf{z}) = \displaystyle\sum_{\ell=1}^{m}\sum_{j\in w_\ell}\sum_{k=1}^{s}\#z_k(v_k^j - g_k^\ell)^2 \quad \text{avec} \quad v_k^j = \frac{\sum_{i\in z_k} x_i^j}{\#z_k} \end{cases}$$

où # représente la cardinalité. L'étape 2(a) est effectuée par l'application de l'algorithme k-means, utilisant la matrice $n \times m$ (u_i^ℓ), la distance euclidienne et les valeurs moyennes des blocs. Alternativement, l'étape 2(b) est obtenue par l'application de l'algorithme k-means utilisant cette fois-ci la matrice $s \times d$ (v_k^j). Ainsi, à la convergence,

nous obtenons des blocs homogènes en réorganisant les lignes et les colonnes selon les partitions $\mathbf{z}$ et $\mathbf{w}$. De cette manière, chaque bloc $\mathbf{x}_k^\ell$, défini par les éléments x_i^j pour $i \in \mathbf{z}_k$ et $j \in \mathbf{w}_\ell$, est caractérisé par g_k^ℓ. Il est évident que cet algorithme est exactement l'algorithme k-means, quand nous restreignons la recherche à une seule des deux partitions.

L'intérêt de cet algorithme a été mis en évidence par comparaison avec k-means appliqué séparément sur les instances et les attributs [Nadif *et al.*, 2002]. Par sa simplicité et sa rapidité, cet algorithme peut s'appliquer sur des données comparables de grande taille. Malheureusement, il requiert la connaissance du nombre de classes en lignes et en colonnes.

3 L'algorithme *Two-way splitting*

Lorsque les données sont directement comparables d'un attribut à un autre, Hartigan (1972) propose un algorithme divisif, *Two-way splitting*, qui choisit à chaque étape entre une division des instances et une division des attributs. Ce choix se base sur la réduction maximum de l'hétérogénéité du groupe d'instances ou de variables divisé. Afin de respecter les contraintes hiérarchiques imposées pour cet algorithme, les divisions effectuées à une étape ne sont jamais remises en cause aux étapes suivantes. Cet algorithme ne nécessite pas de savoir à l'avance le nombre de blocs que l'on veut obtenir. Il peut être décrit succinctement de la manière suivante :

1. Fixer un seuil minimum de variance T, et démarrer avec les instances dans une seule classe et les attributs dans une seule classe.
2. Calculer les variances moyennes de chaque classe en lignes et de chaque classe en colonnes. Les lignes et les colonnes ayant une variance inférieure à T ne sont pas prises en compte. Ainsi, une classe en lignes ne contenant que des objets avec une variance inférieure à T ne sera plus découpée. De même en colonnes.
3. Choisir la classe en lignes ou en colonnes qui a la plus grande variance.
4. Découper cette classe en deux en utilisant une variante de k-means, en ne retenant que les blocs où la variance est supérieure à T.
5. Recommencer à partir de l'étape 2, jusqu'à ce que toutes les variances de chaque bloc soit inférieure au seuil fixé par l'utilisateur.

Cet algorithme permet de mettre en évidence des structures plus fines que celles de *Croeuc*. Notons que nous disposons de plus d'une hiérarchie en lignes et une hiérarchie en colonnes.

Pour illustration de cet algorithme, nous avons choisi de l'appliquer sur des données binaires, de taille $n = 20$ et $d = 10$. Afin de prendre le seuil le plus approprié, nous avons testé les différentes valeurs de 0.1 à 1.0, par pas de 0.1. Les moyennes des variances de chaque bloc sont reportées, pour chaque seuil, dans la figure 1. A partir de 0.3, *Two-way splitting* ne sépare plus les données, et garde un seul bloc. C'est pourquoi nous avons choisi de retenir un seuil de 0.2.

Dans la figure 1, nous présentons la matrice réordonnée selon les blocs et les hiérarchies en lignes et en colonnes fournis par *Two-way splitting*. Il est très clair que cet algorithme propose un découpage en blocs clair.

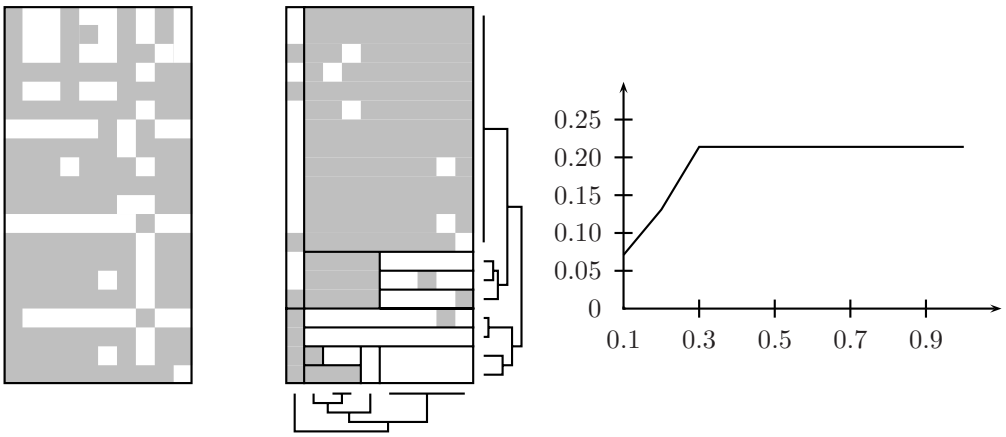

FIG. 1 – *Matrice initiale et réordonnée selon les résultats de Two-way splitting (avec les arbres hiérarchiques associés) et variances moyennes pour chaque seuil, pour le tableau binaire.*

Malheureusement, cet algorithme nécessite le calcul à chaque étape d'un grand nombre de variances, ce qui rend son utilisation sur des données de grande taille inadaptée. De plus, le choix du seuil n'est pas automatique, et nécessite soit une connaissance préalable de celui-ci, soit un test de plusieurs valeurs candidates.

4 Combinaison des algorithmes et application

Puisque l'algorithme *Two-way splitting* travaille sur des attributs comparables, il est possible de le combiner avec un algorithme de type *Croeuc*. En effet, ce dernier propose comme résultat une matrice d'information réduite, contenant les moyennes de chaque bloc. Nous obtenons donc un tableau sur lequel *Two-way splitting* peut parfaitement s'appliquer. Ainsi, nous allons utiliser le schéma de la figure 2 pour les combiner.

De plus, grâce à cette combinaison, nous pouvons nous affranchir du problème du nombre de classes pour *Croeuc*. En effet, lors d'une exécution, il sera alors judicieux de choisir s et m assez grands. Et ensuite, nous pourrons appliquer *Two-way splitting* sur les centres obtenus et avoir ainsi une structure claire de la matrice d'information (et donc des données de base).

Pour illustrer la méthodologie présentée, nous avons décidé de l'appliquer sur des données simulées. En effet, il a été démontré que le critère optimisé par *Croeuc* provient d'un modèle de mélange Gaussien croisé [Govaert et Nadif, 2002, Govaert et Nadif, 2003, Nadif *et al.*, 2002], où les données de chaque bloc suivent une loi normale de centre μ_k^ℓ et de variance $({\sigma_0}^2)$ qui est commune à tous les blocs. Nous avons choisi $n = 5000$, $d = 500$, des proportions égales (en lignes et en colonnes), chaque valeur x_i^j est obtenu avec une loi normale $N(\mu,1)$, où μ est le centre du bloc auquel x_i^j appartient. Nous avons pris comme structure de base la matrice présentée dans la figure 3 (à gauche).

Pour résumer la matrice de données, nous utilisons donc *Croeuc*. Nous avons choisi

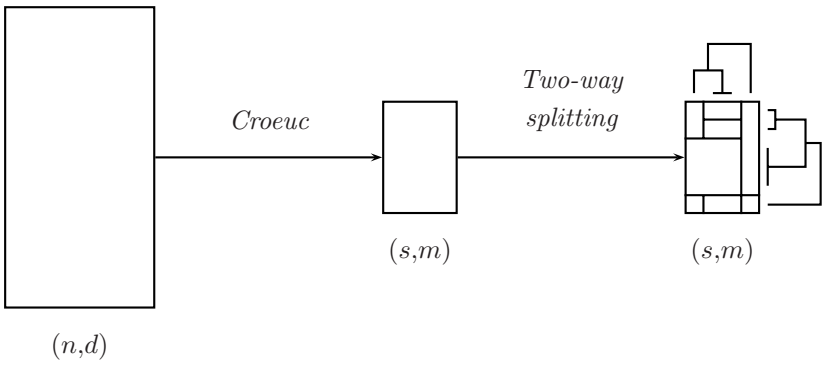

FIG. 2 – *Schéma illustratif décrivant l'utilisation des algorithmes Croeuc et Two-way splitting combinés ensemble.*

de prendre $s = 10$ et $m = 5$, afin d'avoir une sous-matrice réduite très simple à utiliser. Il est bien sûr possible de choisir d'autres nombres de classes, en fonction de la granularité désirée.

Lorsque nous avons lancé *Two-way splitting*, nous avons vu que pour un seuil de 1, il fournit un découpage en bloc approprié, et identique à ceux fournis par les seuils inférieurs (0.1, 0.2, 0.3, ..., 0.9). Nous avons donc décidé de garder ce seuil de 1. Dans la figure 3, on voit très bien que la structure proposé par *Two-way splitting* est la même que celle simulée.

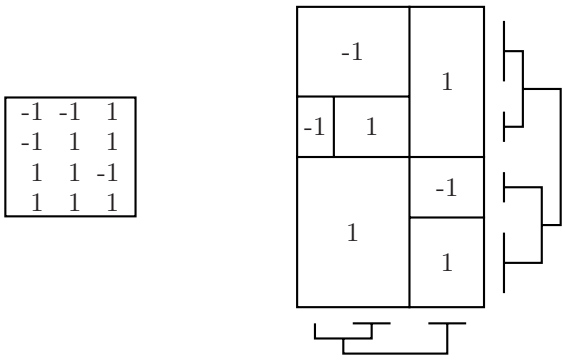

FIG. 3 – *Structure simulée (à gauche) et structure obtenue à l'aide de Two-way splitting appliqué sur les noyaux donnés par Croeuc (à droite).*

5 Conclusion et Perspectives

Après avoir présenté un algorithme de classification croisée, *Croeuc*, nous avons présenté un algorithme de classification directe, *Two-way splitting*. Profitant des avan-

tages des deux méthodes, nous avons proposé une méthodologie d'identification de blocs homogènes en utilisant *Two-way splitting* sur les résultats de *Croeuc*. Cette méthode nous permet de réduire les informations dans une sous-matrice et ensuite de découvrir à l'intérieur de celle-ci une structure en blocs homogènes des attributs et des instances. Les premiers résultats obtenus sont très encourageants et nous persuadent de l'intérêt évident d'une telle démarche. De plus, celle-ci est applicable dans le cas où les variables sont comparables (comme les données de type biopuces, données textuelles, tableaux de pourcentage, ...). Nous envisageons donc de tester cette méthode sur des données réelles, et de plus comparer celle-ci avec notre méthode hybride hiérarchique HBCM [Jollois *et al.*, 2003, Nadif *et al.*, 2002].

Références

[Govaert et Nadif, 2002] G. Govaert et M. Nadif. Block clustering on continuous data. In *Proc. of the Workshop on Clustering High Dimensional Data and its applications at the Second SIAM international Conference on Data Mining*, pages 7–16, 2002.

[Govaert et Nadif, 2003] G. Govaert et M. Nadif. Clustering with block mixture models. *Pattern Recognition*, pages 463–473, 2003.

[Govaert, 1983] G. Govaert. Classification croisé. Thèse d'Etat, Université de Paris 6, France, 1983.

[Govaert, 1995] G. Govaert. Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24:437–458, 1995.

[Hartigan, 1975] J. Hartigan. Direct splitting. In *Clustering Algorithms*, chapter 14, pages 251–277. John Wiley & Sons, New York, 1975.

[Jollois *et al.*, 2003] F.-X. Jollois, M. Nadif, et G. Govaert. Classification croisée sur des données binaires de grande taille. *Extraction et Gestion des Connaissances, RSTI série RIA-ECA*, 17(1-2-3):213–218, 2003.

[Marcotorchino, 1987] F. Marcotorchino. Block seriation problems: A unified approach. *Applied Stochastic Models and Data Analysis*, 3:73–91, 1987.

[Nadif *et al.*, 2002] M. Nadif, G. Govaert, et F.-X. Jollois. A hybrid system for identifying homogenous blocks in large data sets. In *Second Euro-Japanese Workshop on Stochastic Risk Modelling for Finance, Insurance, Production and Reliability, 16-19 septembre, Chamonix, France*, pages 324–333, 2002.

Summary

In contrast to classical clustering methods, which search only a row or column partition, block and direct clustering methods give a reorganized data into homogeneous blocks. The first one consists in searching simultanously rows and columns partitions. The second one works directly on data, and permits to obtain any sized homogenous data blocks. Combining advantages of both methods, we present here a methodology available for large data bases.