Un modèle de mélange pour la classification croisée d'un tableau de données continue

Gérard Govaert, Mohamed Nadif

Un modèle de mélange pour la classification croisée d'un tableau de données continues

Gérard Govaert[1], Mohamed Nadif[2]

[1] Heudiasyc, UMR CNRS 6599
Université de Technologie de Compiègne, 60200 Compiègne
gerard.govaert@utc.fr

[2] CRIP5, Université Paris Descartes, Paris, France
mohamed.nadif@univ-paris5.fr

Résumé : Contrairement aux méthodes de classification automatique habituelles, les méthodes de classification croisée traitent l'ensemble des lignes et l'ensemble des colonnes d'un tableau de données simultanément en cherchant à obtenir des blocs homogènes. Dans cet article, nous abordons la classification croisée lorsque le tableau de données porte sur un ensemble d'individus décrits par des variables quantitatives et, pour tenir compte de cet objectif, nous proposons un modèle de mélange adapté à la classification croisée conduisant à des critères originaux permettant de prendre en compte des situations plus complexes que les critères habituellement utilisés dans ce contexte. Les paramètres sont alors estimés par un algorithme EM généralisé (GEM) maximisant la vraisemblance des données observées. Nous proposons en outre une nouvelle expression du critère bayésien de l'information, appelée BIC$_B$, adaptée à notre situation pour évaluer le nombre de blocs. Des expériences numériques portant sur des données synthétiques permettent d'évaluer les performances de GEM et de BIC$_B$ et de montrer l'intérêt de cette approche.

Mots-clés : Co-clustering, classification croisée, modèle de mélange, algorithme GEM, critère BIC.

1 Introduction

La classification automatique, comme la plupart des méthodes d'analyse de données peut être considérée comme une méthode de réduction et de simplification des données. Dans le cas où les données mettent en jeu deux ensembles I (lignes, objets, observations, individus) et J (colonnes, variables, attributs), ce qui est le cas le plus fréquent, la classification automatique en ne faisant porter la structure recherchée que sur un seul des deux ensembles, agit de façon dissymétrique et privilégie un des deux ensembles, contrairement par exemple aux méthodes factorielles comme l'analyse en composantes principales ou l'analyse factorielle des correspondances qui obtiennent simultanément des résultats sur les deux ensembles ; il est alors intéressant de rechercher *simultané-*

ment une partition des deux ensembles. Cette approche s'inscrit dans le cadre des méthodes de classification par bloc (*block clustering* ou *co-clustering*) qui cherchent à organiser la matrice de données en blocs homogènes. On se limite ici aux structures définies par un couple de partitions respectivement de I et J. Ce type d'approches a suscité récemment beaucoup d'intérêt dans divers domaines tels que celui des biopuces où l'objectif est de caractériser des groupes de gènes par des groupes de conditions expérimentales ou encore celui de l'analyse textuelle où l'objectif est de caractériser des classes de documents par des classes de mots. Un autre avantage de ces méthodes est qu'elles réduisent la matrice initiale de données en une matrice plus simple ayant la même structure. Par ailleurs, ces méthodes sont rapides, peuvent traiter des tableaux de données de grande taille, nécessitent beaucoup moins de calcul que le traitement séparé des deux ensembles et sont en conséquence intéressantes pour la fouille de données.

Les modèles de mélange de lois de probabilité (McLachlan & Peel, 2000) qui supposent que l'échantillon est formé de sous-populations caractérisées chacune par une distribution de probabilité, sont des modèles très intéressants en classification permettant d'une part de donner un sens probabiliste à divers critères classiques et d'autre part de proposer de nouveaux algorithmes généralisant par exemple l'algorithme des k-*means*. Dans le cadre de la classification croisée, il a ainsi été montré que les algorithmes de classification croisé $Crobin$ et $Croki2$ (Govaert, 1983) respectivement adaptés aux données binaires et aux tableaux de contingence peuvent être vus comme des versions classifiantes de l'algorithme *EM* associé à des modèles probabilistes de blocs latents, le premier s'appuyant sur des distributions de Bernoulli (Govaert & Nadif, 2008) et le second sur des distributions de Poisson (Govaert & Nadif, 2005). Notons toutefois que l'estimation des paramètres de ces modèles génératifs par l'algorithme EM n'a pu se faire qu'à l'aide d'une approximation de type variationnel ce qui conduit à remplacer la maximisation de la vraisemblance par la maximisation d'une vraisemblance approchée. On peut aussi remarquer que dans ces deux situations, les ensembles I et J sont traités de manière complètement symétrique.

Dans ce papier, nous proposons d'étendre ce travail à la classification croisée d'un tableau individus-variables continues. Dans cette situation, la classification croisée peut être obtenue de différentes façons. La plus simple consiste à classifier les variables et, en utilisant la matrice de données obtenue en remplaçant les classes de variables par leurs moyennes, de classifier les individus. Pour classifier alors les variables, il est possible d'utiliser des méthodes de classification comme l'algorithme des k-*means* appliquée aux données centrées en colonne, ou des méthodes hiérarchiques comme la procédure SAS Varclus qui est basée sur une analyse en composantes principales obliques (Harman, 1976) dont l'objectif est de regrouper les variables les plus corrélées. D'autres méthodes plus adaptées au problème envisagé et traitant simultanément les deux ensembles ont été proposées (Hartigan, 1975; Bock, 1979; Govaert, 1983; Arabie & Hubert, 1990; Shafiei & Milios, 2006).

L'extension des modèles de blocs latents utilisés pour les tableaux binaires et les tableaux de contingences aux tableaux continues peut se faire sans difficultés techniques en s'appuyant sur les distributions gaussiennes. Cette extension est toutefois discutable car le traitement symétrique des deux ensembles I et J est plus délicat. On peut remarquer que le même problème existe en analyse factorielle : l'analyse en composantes

principales ne traite pas de manière symétrique l'ensemble des individus et des variables contrairement à l'analyse factorielle des correspondances qui traite de la même façon les lignes et les colonnes d'un tableau de contingence. Enfin, l'utilisation du modèle des blocs latents conduit à considérer l'ensemble des individus mais aussi l'ensemble des variables comme des échantillons ce qui est plus que discutable.

Pour surmonter ces difficultés, nous proposons l'utilisation du modèle de mélange standard en intégrant la partition des variables dans le paramétrage des lois de probabilité. L'algorithme EM, ou plutôt sa version généralisée GEM, est alors applicable ce qui garantit que le critère maximisé est bien la vraisemblance et non une approximation comme pour les modèles de blocs latents. En outre, cette approche nous permet de traiter le problème du choix du nombre de blocs en s'appuyant sur une variante du critère bayésien standard de l'information (BIC) (Schwarz, 1978).

Le papier est organisé de la manière suivante. Dans la section 2 et la section 3, nous rappelons respectivement les objectifs de la classification croisée d'un tableau de données continues et l'approche de la classification à l'aide des modèles de mélange. La section 4 est consacrée à la présentation du modèle de mélange retenu pour prendre en compte notre problème de classification croisée. La section 5 présente l'algorithme GEM adapté à ce modèle. Dans la section 6, nous développons une nouvelle expression du critère bayésien adapté à notre modèle. Dans la section 7, nous étudions le comportement de notre algorithme et du critère de sélection proposé sur des données simulées et une dernière section résume les principaux aspects de ce travail.

Notations

Dans tout ce texte, on notera $\mathbf{x} = (x_{ij})$ le tableau de données associé à un ensemble de n individus I mesurés par un ensemble J de d variables continues. Une partition en g classes de l'ensemble I sera représentée par sa matrice de classification notée $(z_{ik}; i = 1, \ldots, n; k = 1, \ldots, g)$ où $z_{ik} = 1$ si i est dans la classe k et $z_{ik} = 0$ sinon. Nous adopterons la même notation $\mathbf{w} = (w_{j\ell}; j = 1, \ldots, d; \ell = 1 \ldots, m)$ pour une partition en m classes de l'ensemble J. Le cardinal de la k^{e} classe de I sera notée $z_k = \sum_{i=1}^{n} z_{ik}$ et celui de la ℓ^{e} classe de $\mathbf{w}$ par $w_\ell = \sum_{j=1}^{d} w_{j\ell}$. Par ailleurs, pour simplifier la présentation, les sommes et les produits portant sur I, J, $\mathbf{z}$ ou $\mathbf{w}$ seront indicés respectivement par les lettres i, j et k et ℓ sans indiquer les bornes de variation qui seront donc implicites. Ainsi, la somme $\sum_{i,j,k,\ell}$ portera sur toutes les lignes i allant de 1 à n, les colonnes j allant de 1 à d, les classes en ligne k allant de 1 à g et les classes en colonne ℓ de 1 à m.

2 Objectif de la classification croisée

L'objectif de la classification croisée est la recherche d'un couple de partitions $(\mathbf{z}, \mathbf{w})$ de l'ensemble des lignes et des colonnes d'un tableau de données initial $\mathbf{x}$ de manière à obtenir, après réorganisation des lignes et des colonnes du tableau $\mathbf{x}$ suivant les deux partitions, des blocs homogènes. Si on représente chacun de ces blocs homogènes $\mathbf{x}_{k\ell} = \{x_{ij} | z_{ik} = 1, w_{j\ell} = 1\}$ par leur moyenne, on obtient ainsi un résumé du tableau initial par une matrice $\mathbf{a}$ de dimension (g, m) où g et m sont les nombres de classes des deux

partitions $\mathbf{z}$ et $\mathbf{w}$ (voir figure 1).

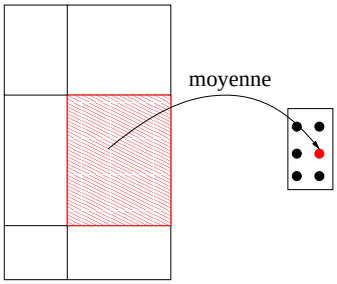

FIG. 1 – Matrice de données $\mathbf{x}$ réordonnée et son résumé $\mathbf{a}$

En prenant la distance euclidienne au carré comme mesure de déviation entre la matrice de données $\mathbf{x}$ et le résumé $\mathbf{a} = (a_{k\ell})$, le problème peut alors s'écrire comme la recherche du couple de partitions $(\mathbf{z}, \mathbf{w})$ et d'une matrice $\mathbf{a}$ minimisant le critère

$$W(\mathbf{z}, \mathbf{w}, \mathbf{a}) = \sum_{k,\ell} \sum_{i|z_{ik}=1} \sum_{j|w_{j\ell}=1} (x_{ij} - a_{k\ell})^2. \tag{1}$$

Il est facile de montrer que, pour un couple de partitions fixées, les valeurs $a_{k\ell}$ optimales sont nécessairement les moyennes des valeurs du bloc $\mathbf{x}_{k\ell}$; la matrice $\mathbf{a}$ optimale sera donc bien définie par les moyennes $a_{k\ell}$ de chaque bloc. Notons qu'en utilisant la notation matricielle des deux partitions le critère peut aussi s'écrire sous la forme

$$W(\mathbf{z}, \mathbf{w}, \mathbf{a}) = ||\mathbf{x} - \mathbf{z}\mathbf{a}\mathbf{w}^T||^2.$$

Différent algorithmes ont été proposés pour minimiser ce critère. Le premier, nommé *two-modes k-means*, a été proposé par Hartigan (1975) ; il peut être décrit de la manière suivante.

1. Choix d'une position initiale $(\mathbf{z}^{(0)}, \mathbf{w}^{(0)}, \mathbf{a}^{(0)})$.
2. Calcul de $(\mathbf{z}^{(c+1)}, \mathbf{a}^{(c+1)})$ à partir de $(\mathbf{z}^{(c)}, \mathbf{w}^{(c)}, \mathbf{a}^{(c)})$
   (a) Calcul de $\mathbf{a}^{(c+\frac{1}{2})}$: $a_{k\ell}^{(c+\frac{1}{2})} = \frac{\sum_{i,j} z_{ik}^{(c)} w_{j\ell}^{(c)} x_{ij}}{z_k^{(c)} w_\ell^{(c)}}$ $\forall k, \ell$
   (b) Calcul de $\mathbf{z}^{(c+1)}$: chaque individu i est affecté à la classe k qui minimise $\sum_{j,\ell} w_{j\ell}^{(c)} (x_{ij} - a_{k\ell}^{(c+\frac{1}{2})})^2$.
   (c) Calcul de $\mathbf{a}^{(c+1)}$: $a_{k\ell}^{(c+1)} = \frac{\sum_{i,j} z_{ik}^{(c+1)} w_{j\ell}^{(c)} x_{ij}}{z_k^{(c+1)} w_\ell^{(c)}}$ $\forall k, \ell$
   (d) Calcul de $\mathbf{w}^{(c+1)}$: chaque variable j est affectée à la classe ℓ qui minimise $\sum_{i,k} z_{i,k}^{(c+1)} (x_{ij} - a_{k\ell}^{(c+1)})^2$.
3. Répéter l'étape 2 jusqu'à la convergence.

Govaert (1983) a proposé une autre version appelé *Croeuc* plus efficace et rapide utilisant les matrices intermédiaires $\mathbf{u} = (u_{i\ell})$ et $\mathbf{v} = (v_{kj})$ définies de la façon suivante : $u_{i\ell} = \sum_{j|w_{j\ell}=1} x_{ij}/w_\ell$ et $v_{kj} = \sum_{i|z_{ik}=1} x_{ij}/z_k$. La minimisation du critère initial peut alors s'effectuer en minimisant alternativement les deux critères conditionnels suivants $W(\mathbf{z}, \mathbf{a}|\mathbf{w}) = \sum_k \sum_{i|z_{ik}=1}(u_{i\ell} - w_\ell a_{k\ell})^2$ et $W(\mathbf{w}, \mathbf{a}|\mathbf{z}) = \sum_\ell \sum_{j|w_{j\ell}=1}(v_{j\ell} - z_k a_{k\ell})^2$. Obtenant ainsi des critères d'inertie intra-classe, les deux minimisations peuvent être obtenues en appliquant l'algorithme des k-*means* à la matrice $\mathbf{u}$ de dimension $n \times m$ pour le premier critère conditionnel et à la matrice $\mathbf{v}$ de dimension $g \times d$ pour le second critère conditionnel.

À la convergence de ces algorithmes, il suffit alors de réorganiser la matrice initiale suivant les deux partitions obtenues pour mettre en évidence des blocs homogènes, chaque bloc pouvant ainsi être caractérisé par la valeur $a_{k\ell}$.

Comme les algorithmes décrits précédemment, la plupart des algorithmes développés dans ce cadre reposent sur une base géométrique et heuristique. L'objectif de ce travail est de se placer dans le cadre probabiliste afin de pouvoir interpréter de manière plus claire le critère optimisé par *Croeuc* et de pouvoir le généraliser à d'autres situations. Dans le paragraphe suivant, nous allons tout d'abord rappeler brièvement l'approche de la classification par les modèles de mélanges.

3 Classification et modèles de mélange

3.1 Modèle de mélange

Les modèles de mélanges finis de lois de probabilité sont très utilisés dans de nombreux domaines des statistiques et tout particulièrement dans le domaine de l'apprentissage non supervisé (voir par exemple l'ouvrage de McLachlan & Peel (2000)). Leur utilisation en classification automatique revient à supposer que les individus à classifier sont issus d'un modèle de mélange dont chaque composant représente une classe. Plus formellement, dans un modèle de mélange fini de lois de probabilité, on considère que les données $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ constituent un échantillon de n réalisations indépendantes d'une variable aléatoire dont la fonction de densité (pdf) peut s'écrire sous la forme :

$$f(\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{k=1}^{g} \pi_k \varphi_k(\boldsymbol{x}_i; \boldsymbol{\alpha}_k), \tag{2}$$

où g est le nombre de composants, les φ_k sont les densités de paramètre $\boldsymbol{\alpha}_k$ de chacun des composants, les π_k sont les proportions du mélange ($\pi_k \in]0,1[\ \forall k$ et $\sum_k \pi_k = 1$) et $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_g; \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_g)$ est le vecteur des paramètres du modèle de mélange.

Dans le contexte des modèles de mélange, le problème de la classification peut être traité en utilisant l'approche *estimation* (*Maximum Likelihood approach*). Celle-ci qui procède par maximisation de la vraisemblance des données observées est de loin la plus utilisée pour aborder ce problème. Elle consiste à estimer le paramètre $\boldsymbol{\theta}$ puis à en déduire une partition de I en utilisant le *principe du maximum a posteriori* (MAP). Les techniques classiques d'optimisation telles que la méthode de Newton-Raphson ou la

méthode du gradient peuvent être appliquées, mais dans notre contexte, l'algorithme EM (Dempster *et al.*, 1977) est sans aucun doute le plus utilisé en raison de ses propriétés de convergence et de sa simplicité de mise en œuvre.

3.2 Algorithme EM

L'objectif de l'algorithme EM est donc la maximisation de la log-vraisemblance $L(\boldsymbol{\theta})$. Son principe, qui repose sur la notion de données complétées, est de maximiser de manière itérative l'espérance de la log-vraisemblance *complétée* conditionnellement au paramètre courant $\boldsymbol{\theta}^{(c)}$ et aux données observées $\mathbf{x}$. Pour le modèle de mélange, les données complétées correspondent tout naturellement au vecteur $(\mathbf{x}, \mathbf{z})$ où $\mathbf{z}$ est le label de chacun des $\mathbf{x}_i$. La log-vraisemblance des données complétées, aussi appelée log-vraisemblance classifiante, s'écrit alors

$$L_c(\mathbf{z}; \boldsymbol{\theta}) = \sum_{i,k} z_{ik} \log \pi_k \varphi_k(\boldsymbol{x}_i; \boldsymbol{\alpha}_k). \tag{3}$$

L'algorithme EM est un algorithme itératif alternant une étape d'estimation *E* et une étape de maximisation *M*.

– Dans l'étape E, on calcule l'espérance conditionnelle de $L_c(\mathbf{z}; \boldsymbol{\theta})$ notée $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ qui s'écrit

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)}) = \sum_{i,k} s_{ik}^{(c)} \{\log(\pi_k) + \log \varphi_k(\boldsymbol{x}_i; \boldsymbol{\alpha}_k)\},$$

où

$$s_{ik}^{(c)} = P(z_{ik} = 1 | \mathbf{x}, \boldsymbol{\theta}^{(c)}) = \frac{\pi_k^{(c)} \varphi_k(\boldsymbol{x}_i; \boldsymbol{\alpha}_k^{(c)})}{\sum_{k'=1}^{g} \pi_{k'}^{(c)} \varphi_{k'}(\boldsymbol{x}_i; \boldsymbol{\alpha}_{k'}^{(c)})}$$

correspond à la probabilité que $\boldsymbol{x}_i$ provienne du k^{e} composant connaissant les données $\mathbf{x}$ et le paramètre $\boldsymbol{\theta}^{(c)}$. Cette étape se réduit donc au calcul des probabilités $s_{ik}^{(c)}$.

– Dans l'étape M, on calcule $\boldsymbol{\theta}^{(c+1)}$ en maximisant en $\boldsymbol{\theta}$ l'espérance conditionnelle $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$. Ce calcul dépendra de la forme de la densité des composants.

Les caractéristiques de l'algorithme EM ont été largement discutées dans de nombreux articles. Cet algorithme, qui conduit en général à des équations simples, a la propriété de faire croître la log-vraisemblance à chaque itération jusqu'à la stationnarité, fournit de bonnes estimations des paramètres dans beaucoup de circonstances et, par conséquent, est devenu un outil standard pour l'estimation du maximum de vraisemblance.

Enfin, en terme de classification non supervisée, l'algorithme EM appliqué au modèle de mélange peut être vu comme un algorithme de classification floue et peut même fournir une partition en employant le principe du MAP, vu précédemment, à partir des paramètres estimés.

Dans la suite et pour surmonter les difficultés décrites dans la section 1, nous proposons un modèle de mélange *asymétrique* prenant en compte l'objectif de la classification croisée.

4 Un modèle de mélange pour la classification croisée

Pour tenir compte du problème posé par la classification croisée, nous proposons d'utiliser le modèle de mélange classique (2) dans lequel cette fois la partition des variables $\mathbf{w}$ est considérée comme un paramètre du modèle. La densité du mélange s'écrit

$$f(\boldsymbol{x}_i;\boldsymbol{\theta}) = \sum_k \pi_k \varphi_k(\boldsymbol{x}_i;\mathbf{w},\boldsymbol{\alpha})$$

où $\varphi_k(\boldsymbol{x}_i;\mathbf{w},\boldsymbol{\alpha})$ prend la forme suivante

$$\prod_{j,\ell}\left(\frac{1}{\sqrt{2\pi\sigma_{k\ell}^2}}e^{-\frac{1}{2\sigma_{k\ell}^2}(x_{ij}-\mu_{k\ell})^2}\right)^{w_{j\ell}}.$$

Le paramètre du modèle de mélange $\boldsymbol{\theta} = (\boldsymbol{\pi},\mathbf{w},\boldsymbol{\alpha})$ est formé par les proportions $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_g)$, la partition des variables $\mathbf{w}$ et les paramètres de chaque composant $\boldsymbol{\alpha} = (\mu_{11},\ldots,\mu_{gm},\sigma_{11}^2,\ldots,\sigma_{gm}^2)$ où les $\mu_{k\ell}$ et les $\sigma_{k\ell}^2$ représentent les moyennes et les variances de chaque bloc. La log-vraisemblance s'écrit alors

$$L(\boldsymbol{\theta}) = \log f(\mathbf{x};\boldsymbol{\theta}) = \sum_i \log \sum_k \pi_k \varphi_k(\boldsymbol{x}_i;\mathbf{w},\boldsymbol{\alpha})$$

et, si nous notons $z_k = \sum_i z_{ik}$ et $w_\ell = \sum_j w_{j\ell}$ les cardinaux de chaque classe, la log-vraisemblance classifiante vérifie

$$L_c(\mathbf{z},\mathbf{w};\boldsymbol{\theta}) = \sum_{i,k} z_{ik}\log\left(\pi_k\varphi_k(\mathbf{x}_i;\mathbf{w},\boldsymbol{\alpha})\right),$$

et prend par conséquent, à la constante additive $-\frac{nd}{2}\log 2\pi$ près, la forme suivante :

$$L_c(\mathbf{z},\mathbf{w};\boldsymbol{\theta}) = \sum_k z_k\log\pi_k - \frac{1}{2}\sum_{i,j,k,\ell} z_{ik}w_{j\ell}\left(\log\sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2}(x_{ij}-\mu_{k\ell})^2\right).$$

Nous pouvons alors étendre l'écriture de la log-vraisemblance classifiante L_c, définie pour une partition $\mathbf{z}$, à la partition floue associée à $\mathbf{s} = (s_{ik}; i = 1,\ldots,n; k = 1,\ldots,g)$ matrice de classification définie par les probabilités conditionnelles.

$$L_c(\mathbf{s},\mathbf{w};\boldsymbol{\theta}) = \sum_{i,k} s_{ik}\log\left(\pi_k\varphi_k(\boldsymbol{x}_i;\mathbf{w},\boldsymbol{\alpha})\right)$$

qui peut s'écrire

$$L_c(\mathbf{s},\mathbf{w};\boldsymbol{\theta}) = \sum_k s_k\log\pi_k - \frac{1}{2}\sum_{i,j,k,\ell} s_{ik}w_{j\ell}\left(\log\sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2}(x_{ij}-\mu_{k\ell})^2\right),$$

où $s_k = \sum_i s_{ik}$.

5 Un algorithme GEM pour la classification croisée

En considérant notre modèle sous l'approche estimation, nous proposons d'utiliser l'algorithme EM, ou plus exactement, l'algorithme GEM. Partant d'une position initiale $(\mathbf{w}^{(0)}, \boldsymbol{\theta}^{(0)})$, cet algorithme va itérer les deux étapes E et M que nous allons maintenant préciser.

5.1 Étape E

Comme il a été indiqué précédemment, pour les modèles de mélange cette étape se réduit au calcul des probabilités conditionnelles a posteriori, $s_{ik}^{(c)}$

$$s_{ik}^{(c)} = \frac{\pi_k^{(c)} \varphi_k(\boldsymbol{x}_i; \mathbf{w}^{(c)}, \boldsymbol{\alpha}^{(c)})}{\sum_{k'} \pi_{k'}^{(c)} \varphi_{k'}(\boldsymbol{x}_i; \mathbf{w}^{(c)}, \boldsymbol{\alpha}^{(c)})}.$$

Calcul de s_{ik}

Ces probabilités conditionnelles peuvent s'écrire $s_{ik} = \frac{e^{S_{ik}}}{\sum_{k'} e^{S_{ik'}}}$ où

$$S_{ik} = \log(\pi_k \varphi_k(\boldsymbol{x}_i; \mathbf{w}, \boldsymbol{\alpha})).$$

Après quelques calculs algébriques, on peut montrer que le terme S_{ik} prend la forme suivante

$$\log \pi_k - \frac{1}{2} \sum_{\ell} \left(w_\ell \log \sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2} (e_{i\ell} + w_\ell (u_{i\ell} - \mu_{k\ell})^2) \right),$$

avec $u_{i\ell} = \frac{\sum_j w_{j\ell} x_{ij}}{w_\ell}$ et $e_{i\ell} = \sum_j w_{j\ell} (x_{ij} - u_{i\ell})^2$, plus facile à calculer que la probabilité initiale s_{ik}.

5.2 Étape M

La maximisation de $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ n'est pas simple et, en utilisant l'algorithme EM généralisé (*Generalized EM algorithm*, GEM) (Dempster *et al.*, 1977) pour lequel dans l'étape M on ne cherche plus à maximiser la quantité $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ mais simplement à la faire croître. Sachant que l'espérance conditionnelle $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ peut aussi s'exprimer comme la log-vraisemblance classifiante floue $L_c(\mathbf{s}^{(c)}, \mathbf{w}, \boldsymbol{\theta})$, cette fonction Q peut aussi s'écrire

$$\sum_k s_k^{(c)} \log \pi_k - \frac{1}{2} \sum_{i,j,k,\ell} s_{ik}^{(c)} w_{j\ell} \left(\log \sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2} (x_{ij} - \mu_{k\ell})^2 \right).$$

Pour faire croître cette fonction Q, nous proposons alors d'itérer jusqu'à la convergence les deux étapes suivantes : maximisation de $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ en $\mathbf{w}$ pour $\mathbf{s}$ et $\boldsymbol{\theta}^{(c)}$ fixés puis maximisation de $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ en $\boldsymbol{\theta}$ pour $\mathbf{w}$ et $\mathbf{s}$ fixés.

Calcul de $\mathbf{w}$

Cette étape consiste à maximiser $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)})$ en $\mathbf{w}$. L'expression précédente de $L_c(\mathbf{s}^{(c)}, \mathbf{w}, \boldsymbol{\theta})$ peut s'écrire

$$\sum_k s_k^{(c)} \log \pi_k + \sum_{j,\ell} w_{j\ell} T_{j\ell}^{(c)},$$

où $T_{j\ell}^{(c)} = -\frac{1}{2} \sum_{i,k} s_{ik}^{(c)} \left(\log \sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2} (x_{ij} - \mu_{k\ell})^2 \right)$. La variable j appartient à la classe maximisant $T_{j\ell}^{(c)}$ et nous obtenons

$$w_{j\ell}^{(c)} = \begin{cases} 1 & \text{si} \quad \ell = \text{argmax}_{\ell'=1,\ldots,m} \, T_{j\ell'}^{(c)} \\ 0 & \text{sinon.} \end{cases}$$

Comme pour le calcul de S_{ik}, il est facile de montrer que le terme $T_{j\ell}$ prend la forme suivante

$$-\frac{1}{2} \sum_k \left(s_k^{(c)} \log \sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2} \left(f_{jk} + s_k (v_{kj} - \mu_{k\ell})^2 \right) \right).$$

où

$$v_{kj} = \frac{\sum_i s_{ik} x_{ij}}{s_k} \quad \text{et} \quad f_{jk} = \sum_i s_{ik} (x_{ij} - v_{jk})^2.$$

Calcul de $\boldsymbol{\theta}$ *à partir de* $\mathbf{w}$ *et* $\mathbf{s}$

Cette étape consiste à maximiser $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(c)})$ en $\boldsymbol{\pi}$ et $\boldsymbol{\alpha} = (\mu_{11}, \ldots, \mu_{gm}, \sigma_{11}^2, \ldots, \sigma_{gm}^2)$. En écrivant la log-vraisemblance classifiante sous la forme

$$L_c(\mathbf{s}, \mathbf{w}; \boldsymbol{\theta}) = \sum_k s_k \log \pi_k - \frac{1}{2} \sum_{k,\ell} \left(s_k w_\ell \log \sigma_{k\ell}^2 + \frac{1}{\sigma_{k\ell}^2} \sum_{i,j} s_{ik} w_{j\ell} (x_{ij} - \mu_{k\ell})^2 \right),$$

on peut alors en déduire les valeurs suivantes $\pi_k^{(c+1)} = \frac{s_k^{(c)}}{n}$, $\mu_{k\ell}^{(c+1)} = \frac{\sum_{ij} s_{ik}^{(c)} w_{j\ell}^{(c)} x_{ij}}{s_k^{(c)} w_\ell^{(c)}}$ et $(\sigma_{k\ell}^2)^{(c+1)} = \frac{\sum_{i,j} s_{ik}^{(c)} w_{j\ell}^{(c)} \left(x_{ij} - \mu_{k\ell} \right)^2)}{s_k^{(c)} w_\ell^{(c)}}$. Ces calculs peuvent être optimisés en utilisant les valeurs v_{jk} et f_{jk} définies précédemment, ce qui permet d'accélérer cette étape. Le centre et la variance de chaque bloc sont alors $\mu_{k\ell}^{(c+1)} = \frac{\sum_j w_{j\ell}^{(c)} v_{jk}}{s_k^{(c)} w_\ell^{(c)}}$ et $(\sigma_{k\ell}^2)^{(c+1)} = \frac{\sum_j w_{j\ell}^{(c)} \left(f_{jk} + s_k^{(c)} (v_{jk} - \mu_{k\ell})^2 \right)}{s_k^{(c)} w_\ell^{(c)}}$.

5.3 Utilisation de l'algorithme

Les algorithmes GEM comme EM sont connus pour leur convergence lente, en particulier lorsque les classes sont mal séparées. Par ailleurs, ils fournissent des solutions dépendant fortement de la position initiale et produisent donc des estimateurs sous-optimaux au sens du maximum de vraisemblance. Notons que les deux inconvénients,

lenteur de la convergence et dépendance à la position initiale, peuvent être vus comme liés en pratique. En fait, il est possible que certaines positions conduisent à des convergences très lentes et que l'algorithme soit stoppé avant d'atteindre l'optimum local. Pour remédier à cette dépendance élevée de GEM à la position initiale, nous proposons d'employer la stratégie em-EM (Biernacki *et al.*, 2003) qui consiste en une première phase (em) de plusieurs exécutions courtes de EM lancées à partir de positions aléatoires et interrompues avant la convergence, suivie d'une seconde phase (EM) consistant en une exécution de EM, partant de la meilleure solution trouvée dans la première phase, menée jusqu'à la convergence. Notons qu'avec notre algorithme, différentes phases de calcul sont optimisées dans l'étape M grâce à l'utilisation des matrices de taille réduite (v_{kj}) et (f_{jk}) et qu'en conséquence l'algorithme GEM est parfaitement adapté à des données de grande taille.

En terme de classification, les probabilités s_{ik} calculées à partir des paramètres estimés par l'algorithme GEM peuvent être interprétées comme une classification floue des individus. On peut alors en déduire une partition $\mathbf{z}$ des individus en utilisant une étape de classification qui consiste à affecter chaque individu $\boldsymbol{x}_i$ à la classe maximisant la probabilité a posteriori d'appartenance s_{ik}. Avec la partition optimale $\mathbf{w}$, nous obtenons par conséquent une partition en blocs.

À partir de cet algorithme GEM, il est possible de définir une version classificatoire (CEM) (Celeux & Govaert, 1992) en remplaçant la maximisation de $L(\boldsymbol{\theta})$ par celle de $L_c(\mathbf{z}, \mathbf{w}; \boldsymbol{\theta})$. Il suffit pour cela d'introduire une phase de classification des individus après l'étape E. Dans le cas particulier où les proportions π_k sont supposées égales et que les variances sont constantes, la maximisation de la fonction $L_c(\mathbf{z}, \mathbf{w}; \boldsymbol{\theta})$ est équivalente à la minimisation du critère $W(\mathbf{z}, \mathbf{w}, \mathbf{a})$ (1) et l'algorithme ainsi obtenu GEM n'est autre que l'algorithme *Croeuc* décrit dans la section 2. Lorsqu'aucune contrainte n'est imposée au paramètre, on obtient donc un critère original permettant d'obtenir des blocs de variances différentes contrairement aux approches habituelles qui supposent, implicitement ou non, une égalité de ces variances.

6 BIC$_B$: un critère de sélection de modèle pour la classification croisée

La détermination des nombres de composants g et m peut être vue comme un problème de sélection de modèles. L'une des réponses apportées par les statisticiens dans ce cadre est l'utilisation d'un critère pénalisé. Parmi ces critères, on peut citer le critère AIC (*Akaike Information Criterion*) Akaike (1973), le critère BIC (*Bayesian Information Criterion*) Schwarz (1978) et le critère MDL (*Minimum Description Length*) Rissanen (1978). Nous avons retenu ici le critère BIC qui se place dans un contexte bayésien de sélection de modèles. Il s'agit d'un critère s'appuyant sur une pénalisation de la vraisemblance tenant compte de la complexité du modèle. Plus précisément, l'expression classique de ce critère comporte deux termes : le premier, qui correspond à une mesure de l'ajustement du modèle, est la vraisemblance ; le second mesure sa complexité. Il est bien connu que ce critère BIC est une approximation asymptotique du calcul de la vraisemblance des données conditionnellement au modèle. Malheureusement, cette

approximation n'est pas applicable ici. Elle s'appuie en effet sur l'approximation de Laplace qui impose que l'espace des paramètres soit continu. Or, dans notre modèle de mélange, le composant $\mathbf{w}$ du paramètre $\theta = (\boldsymbol{\pi}, \boldsymbol{\alpha}, \mathbf{w})$ appartient à $\mathcal{W}$, ensemble de toutes les affectations possibles des d variables en m classes qui est un ensemble discret. Notons que l'on rencontre la même difficulté avec le critère MDL. Les expériences numériques que nous avons menées et qui ne sont pas reportées ici confirment cette difficulté et ont montré que l'application directe du critère BIC conduit à une surestimation importante du nombre de classes m.

Pour tenir compte de ce problème, nous pouvons retourner à la formulation originale du critère BIC. Il est défini dans un contexte bayésien : (g, m) et $\boldsymbol{\theta}$ sont des variables aléatoires de distribution a priori $p(g, m)$ et $p(\boldsymbol{\theta}|g, m)$. Les nombres de classes retenus sont alors les valeurs g et m maximisant la probabilité a posteriori $p(g, m|\mathbf{x})$ où $\mathbf{x}$ correspond aux données disponibles. Sachant que

$$p(\mathbf{x}|g, m) = \frac{p(\mathbf{x}|g, m)p(g, m)}{p(\mathbf{x})}$$

et utilisant une distribution a priori non informative (ici, la loi uniforme sur (g, m)), ce problème est équivalent à la maximisation de $p(\mathbf{x}|g, m)$. Comme $p(\mathbf{x}, \boldsymbol{\theta}|g, m) = p(\mathbf{x}|\boldsymbol{\theta}, g, m)p(\boldsymbol{\theta})$, et en supposant que $p(\boldsymbol{\theta}) = p(\mathbf{w})p(\boldsymbol{\pi}, \boldsymbol{\alpha})$, l'expression à maximiser prend la forme suivante, souvent appelée *vraisemblance intégrée*,

$$p(\mathbf{x}|g, m) = \int_{\boldsymbol{\Theta}} \underbrace{\sum_{\mathcal{W}} p(\mathbf{x}|\boldsymbol{\theta}, g, m)p(\mathbf{w})}_{C} p(\boldsymbol{\pi}, \boldsymbol{\alpha}) d\boldsymbol{\pi} d\boldsymbol{\alpha}$$

où $\boldsymbol{\Theta}$ est l'ensemble contenant le paramètre $\boldsymbol{\theta}$ et $\mathcal{W}$ est l'ensemble de toutes les affectations possibles des n individus en g classes. Pour notre modèle de mélange, nous avons

$$C = \sum_{\mathcal{W}} \prod_{i} \sum_{k} \pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \boldsymbol{\alpha}_k) p(\mathbf{w}).$$

Sachant que l'expression $\prod_i \sum_k \pi_k \varphi_k(\mathbf{x}_i; \mathbf{w}, \boldsymbol{\alpha}_k)$ peut s'écrire (Govaert & Nadif, 2003)

$$\sum_{\mathcal{Z}} p(\mathbf{z}) g(\mathbf{x}; \mathbf{z}, \mathbf{w}, \boldsymbol{\alpha}),$$

le terme C correspond alors à la densité d'un modèle de blocs latents

$$h(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{\mathcal{Z} \times \mathcal{W}} p(\mathbf{w}) p(\mathbf{z}) g(\mathbf{x}; \mathbf{z}, \mathbf{w}, \boldsymbol{\alpha})$$

où $g(\mathbf{x}; \mathbf{z}, \mathbf{w}, \boldsymbol{\alpha}) = \prod_{i,j,k,\ell} \left(\varphi(x_{ij}; \boldsymbol{\alpha}_{k\ell})\right)^{z_{ik} w_{j\ell}}$ est un produit de distributions gaussiennes et $p(\mathbf{z})$ est défini par $\prod_{i,k} (\pi_k)^{z_{ik}}$.

Finalement, le vraisemblance intégrée s'exprime sous la forme classique

$$p(\mathbf{x}|g, m) = \int_{\mathcal{A}} h(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\alpha}) p(\boldsymbol{\pi}, \boldsymbol{\alpha}) d\boldsymbol{\pi} d\boldsymbol{\alpha}$$

où h est une densité. Pour ce modèle de blocs latents, l'approximation de Laplace peut être maintenant utilisée et nous obtenons l'approximation suivante

$$p(\mathbf{x}|g,m) \approx \log(h(\mathbf{x};\tilde{\pi},\tilde{\boldsymbol{\alpha}}) - \frac{\nu \log(nd)}{2}$$

où $\tilde{\boldsymbol{\pi}}$ et $\tilde{\boldsymbol{\alpha}}$ sont les estimations du maximum de vraisemblance de π et $\boldsymbol{\alpha}$ pour le modèle des blocs latents et ν est la dimension du paramètre $(\boldsymbol{\pi}, \boldsymbol{\alpha})$.

Cette estimation étant difficile, nous avons remplacé la log-vraisemblance $\log(h(\mathbf{x};\tilde{\pi},\tilde{\boldsymbol{\alpha}}))$ par la log-vraisemblance $L(\boldsymbol{\theta})$ obtenue à la convergence de l'algorithme GEM défini dans la section 5. En utilisant une distribution uniforme $p(\mathbf{w}) = \prod_{j,\ell}(\frac{1}{m})^{w_{j\ell}}$, nous proposons finalement d'utiliser le critère suivant

$$BIC_B(g,m) = L - d\log m - \frac{\nu}{2}\log(nd)$$

où L est la vraisemblance obtenue à la convergence de l'algorithme.

7 Expérimentations numériques

7.1 Conditions expérimentales

Dans ces premières expérimentations, nous nous sommes limité au modèle le plus simple pour lequel les proportions des classes sont égales et la variance est identique pour tous les blocs. Nous avons simulé plusieurs types de données provenant d'un mélange à 3 classes en lignes et 2 en colonnes et correspondant à trois degrés de recouvrement des classes : bien séparé, modérément séparé et mal séparé et nous avons pris à chaque fois $n = 100$ et $d = 20$. Dans le cas de la classification croisée, la notion de séparation des classes est difficile à être visualisée mais le degré de séparation peut être mesuré par le taux d'erreur de classification calculé en comparant les partitions simulées et celles obtenues en appliquant une étape de classification à partir des vrais paramètres. Dans notre expérimentation, nous avons retenu les trois taux d'erreur suivants : 5% pour les classes bien séparées (M1), 14% pour les classes modérément séparées (M2) et 22% pour les classes mal séparées.

7.2 Comportement de GEM

Si l'objectif recherché est de déterminer une partition de l'ensemble I, il est intéressant de comparer l'algorithme GEM décrit dans ce travail à l'algorithme EM appliqué au modèle de mélange gaussien diagonal ignorant la classification des variables. Pour comparer les deux partitions $\mathbf{z}$ et $\mathbf{z}'$ ainsi obtenues, le taux d'erreur, c'est-à-dire la proportion d'individus mal classifiés, peut être définie de la façon suivante : si on note C la matrice de confusion entre les deux partitions, matrice symétrique car les deux partitions ont le même nombre de classes, les classes de la partition $\mathbf{z}'$ sont renumérotées de façon à maximiser la trace de la matrice C (dans nos expériences, nous avons énuméré toutes les renumérotations) ; le taux d'erreur s'exprime alors sous la forme

TAB. 1 – Comparaison des résultats de GEM, EM et CROEUC ($n \times d = 100 \times 20$)

Taux d'erreur	Situations	GEM	EM	CROEUC
$\delta(\mathbf{z}, \mathbf{z}')$	M1	0.06	0.05	0.06
	M2	0.15	0.19	0.31
	M3	0.29	0.39	0.41

suivante $\delta(\mathbf{z}, \mathbf{z}') = 1 - \frac{1}{n} \sum_{i,k} z_{ik} z'_{ik}$. Les résultats ainsi obtenus ont été résumés dans le tableau 1.

Les principales conclusions que l'on peut tirer de cette première série d'expériences sont les suivantes :

- L'algorithme EM appliqué à l'ensemble des individus est efficace seulement quand les classes sont bien séparées. Dans le cas contraire, le fait d'utiliser un modèle s'appuyant sur un regroupement des variables en classes a largement amélioré les résultats.
- Les taux d'erreur obtenus par l'algorithme GEM se rapprochent des taux d'erreur attendus.
- Enfin, une étude détaillée effectuée en augmentant les tailles n et d, non reportée ici, montre une convergence plus rapide des taux d'erreur obtenus avec GEM.

7.3 Comportement de BIC$_B$

Pour illustrer le comportement du critère de sélection BIC$_B$, nous avons étudié ses performances à l'aide de données de tailles 50×10, 100×20 et 500×100 simulées suivant notre modèle en faisant varier g le nombre de classes en lignes de 1 à 8 et m le nombre de classes en colonne de 1 à 5. Pour toutes ces situations, trois groupes de paramètres ont été choisis de façon à obtenir les degrés de séparation $M1$, $M2$ et $M3$ définis précédemment. Les résultats obtenus ont été résumés dans les tables 2, 3 et 4. Dans ces tables, les différentes valeurs du critère ont été reportées et les meilleures valeurs ont été indiquées en gras.

Les conclusions que l'on peut tirer à partir de ces premières expériences portant sur le choix du nombre de classes sont les suivantes :

- Quand les classes sont bien ou modérément séparées (situations M1 et M2), le critère BIC$_B$ est efficace et donne de bons résultats et ses performances augmentent avec la taille des données.
- Quand les classes sont mal séparées, le critère BIC$_B$ a des difficultés et a tendance à sous-estimer le bon nombre de classes. Cependant, pour les grandes tailles (Table 4) et si le nombre de classes m est connu, BIC$_B$ donne exactement le bon nombre de classes en lignes. Cette observation a été confirmée sur de nombreuses autres simulations (non reportées ici) toujours avec des données de grande taille. Une explication potentielle de ce comportement pourrait être le remplacement des estimateurs $\tilde{\pi}$ et $\tilde{\boldsymbol{\alpha}}$ par les valeurs obtenues à la convergence de l'algorithme GEM effectué dans la construction du critère BIC$_B$ (section 4).

TAB. 2 – Valeurs de BIC$_B$ pour $n = 50$ et $d = 10$

		m				
		1	2	3	4	5
Situation	g					
	1	-1218	-1222	-1229	-1235	-1240
	2	-1212	-1200	-1209	-1217	-1225
	3	-1212	**-1182**	-1192	-1203	-1212
	4	-1214	-1187	-1200	-1210	-1221
M1	5	-1216	-1191	-1204	-1217	-1230
	6	-1219	-1196	-1211	-1225	-1238
	7	-1222	-1202	-1219	-1234	-1249
	8	-1224	-1207	-1227	-1244	-1260
	1	-1336	-1327	-1333	-1338	-1344
	2	-1333	**-1303**	-1311	-1318	-1325
	3	-1335	**-1303**	-1313	-1323	-1333
	4	-1338	-1306	-1317	-1329	-1339
M2	5	-1341	-1310	-1323	-1336	-1349
	6	-1344	-1315	-1331	-1346	-1362
	7	-1347	-1321	-1339	-1357	-1374
	8	-1350	-1327	-1348	-1367	-1385
	1	**-1440**	-1442	-1448	-1453	-1459
	2	**-1440**	-1443	-1448	-1454	-1460
	3	-1443	-1444	-1451	-1458	-1466
	4	-1446	-1450	-1458	-1466	-1475
M3	5	-1449	-1456	-1467	-1477	-1489
	6	-1452	-1462	-1476	-1489	-1502
	7	-1455	-1468	-1485	-1501	-1516
	8	-1458	-1474	-1494	-1513	-1530

TAB. 3 – Valeurs de BIC$_B$ pour $n = 100$ et $d = 20$

		m				
		1	2	3	4	5
Situation	g					
	1	-5505	-5512	-5522	-5530	-5538
	2	-5452	-5442	-5454	-5465	-5475
	3	-5453	**-5403**	-5418	-5431	-5444
	4	-5458	-5412	-5430	-5445	-5461
M1	5	-5461	-5419	-5439	-5458	-5477
	6	-5465	-5425	-5446	-5468	-5490
	7	-5469	-5432	-5457	-5480	-5505
	8	-5472	-5440	-5468	-5494	-5520
	1	-5933	-5936	-5945	-5954	-5962
	2	-5908	-5908	-5913	-5923	-5933
	3	-5910	**-5889**	-5896	-5909	-5922
	4	-5913	-5895	-5906	-5915	-5929
M2	5	-5916	-5902	-5916	-5925	-5942
	6	-5920	-5909	-5925	-5938	-5959
	7	-5923	-5915	-5936	-5951	-5972
	8	-5927	-5922	-5947	-5965	-5988
	1	-6267	-6277	-6287	-6295	-6303
	2	**-6261**	-6274	-6283	-6294	-6304
	3	-6265	-6278	-6289	-6301	-6316
	4	-6269	-6285	-6299	-6314	-6331
M3	5	-6272	-6291	-6310	-6329	-6345
	6	-6276	-6300	-6320	-6342	-6366
	7	-6280	-6306	-6325	-6354	-6384
	8	-6284	-6315	-6342	-6370	-6398

TAB. 4 – Valeurs de BIC_B pour $n = 500$ et $d = 100$

		m				
		1	2	3	4	5
Situation	g					
	1	-169191	-169176	-169205	-169232	-169255
	2	-168885	-168749	-168777	-168802	-168824
	3	-168872	**-168375**	-168407	-168434	-168460
	4	-168877	-168403	-168439	-168464	-168491
M1	5	-168882	-168408	-168447	-168483	-168511
	6	-168888	-168406	-168450	-168488	-168527
	7	-168893	-168420	-168471	-168509	-168565
	8	-168898	-168430	-168486	-168537	-168587
	1	-185587	-185603	-185634	-185663	-185687
	2	-185496	-185487	-185518	-185542	-185561
	3	-185486	**-185377**	-185413	-185440	-185467
	4	-185491	-185387	-185421	-185467	-185497
M2	5	-185498	-185404	-185449	-185490	-185532
	6	-185502	-185410	-185459	-185510	-185548
	7	-185507	-185417	-185481	-185545	-185587
	8	-185513	-185432	-185499	-185564	-185620
	1	-200695	-200726	-200760	-200789	-200814
	2	**-200634**	-200668	-200701	-200728	-200755
	3	**-200634**	-200655	-200681	-200706	-200734
	4	-200640	-200667	-200700	-200752	-200761
M3	5	-200646	-200675	-200721	-200769	-200810
	6	-200651	-200687	-200746	-200801	-200842
	7	-200656	-200706	-200777	-200831	-200867
	8	-200662	-200709	-200800	-200852	-200906

8 Conclusion

Quand les données se composent d'un grand nombre de variables définies sur un grand nombre d'individus, comme dans le contexte de la fouille de données, les algorithmes de classification croisée peuvent être une approche intéressante. En regroupant simultanément les individus et les variables, ils nous permettent de mettre en évidence des blocs homogènes auxquels on peut associer un résumé des données.

Dans ce travail, pour tenir compte de la structure des données – individus mesurés par des variables quantitatives – nous avons proposé un algorithme de classification croisé s'appuyant sur un modèle de mélange standard adapté à ce type de données. Cette approche conduit à une estimation de paramètres que nous avons effectuée sous l'approche du maximum de vraisemblance en utilisant un algorithme de type GEM. Ce modèle généralise les approches classiques en permettant d'associer à chaque bloc des variances différentes ce qui rend cette approche originale. L'algorithme obtenu est efficace et approprié aux données de grandes tailles. Nous avons en outre proposé un critère de sélection de modèles et évalué ses performances à partir d'expériences numériques. Ce critère semble efficace quand les classes sont bien ou modérément séparés. Même si ce critère a quelques problèmes pour choisir le bon nombre de blocs, on peut noter son très bon comportement pour déterminer le bon nombre de classes en lignes lorsque le nombre de classes en colonne est connu et que le nombre d'individus est assez grand.

Plusieurs aspects restent à être étudiés : améliorer le critère BIC_B en évitant le remplacement des estimateurs $\tilde{\boldsymbol{\pi}}$ et $\tilde{\boldsymbol{\alpha}}$ par les valeurs obtenues à la convergence de notre al-

gorithme ; pour le modèle proprement dit, différentes versions parcimonieuses peuvent être obtenues en imposant des contraintes sur les variances (variance égales, variances égales par ligne ou par colonne, variances obtenues comme la somme d'une variance ligne et d'une variance colonne,...). Il resterait alors à étudier de manière précise le comportement de ces différentes variantes sur des données simulées et sur des données réelles. Il serait aussi alors important d'étudier le comportement du critère BIC_B pour effectuer la sélection du modèle parcimonieux le mieux adapté aux données.

Remerciements : les auteurs remercient l'aide apportée par l'Agence Nationale de la Recherche (ANR) dans le cadre du projet ClasSel « Classification croisée et sélection de modèle ».

Références

AKAIKE H. (1973). Information theory and an extension of the maximum likelihood principle. In B. PETROV & F. CSAKI, Eds., *Second International Symposium on Information Theory*, p. 267–281, Budapest : Akademiai Kiado.

ARABIE P. & HUBERT L. J. (1990). The bond energy algorithm revisited. *IEEE Transactions on Systems, Man, and Cybernetics*, **20**, 268–274.

BIERNACKI C., CELEUX G. & GOVAERT G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics and Data Analysis*, **41**, 561–575.

BOCK H. (1979). Simultaneous clustering of objects and variables. In E. DIDAY, Ed., *Analyse des Données et Informatique*, p. 187–203 : INRIA.

CELEUX G. & GOVAERT G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, **14**(3), 315–332.

DEMPSTER A. P., LAIRD N. M. & RUBIN D. B. (1977). Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society*, **B 39**, 1–38.

GOVAERT G. (1983). *Classification croisée*. Thèse d'état, Université Paris 6, France.

GOVAERT G. & NADIF M. (2003). Clustering with block mixture models. *Pattern Recognition*, **36**, 463–473.

GOVAERT G. & NADIF M. (2005). An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(4), 643–647.

GOVAERT G. & NADIF M. (2008). Block clustering with Bernoulli mixture models : Comparison of different approaches. *Computational Statistics and Data Analysis*, **52**, 3233–3245.

HARMAN H. (1976). *Modern Factor analysis, Third Edition*. Chicago : University of Chicago Press.

HARTIGAN J. A. (1975). *Clustering Algorithms*. New York : Wiley.

MCLACHLAN G. & PEEL D. (2000). *Finite Mixture Models*. New York : Wiley.

RISSANEN J. (1978). Modeling by shortest data description. *Automatica*, **14**, 445–471.

SCHWARZ G. (1978). Estimating the number of components in a finite mixture model. *Annals of Statistics*, **6**, 461–464.

SHAFIEI M. M. & MILIOS E. M. (2006). Latent dirichlet co-clustering. In *ICDM 2006*, p. 542–551.