Revue des méthodes pour la classification jointe des lignes et des colonnes d'un tableau

Vincent Brault, Aurore Lomet

Journal de la Société Française de Statistique
Vol. 156 No. 3 (2015)

Revue des méthodes pour la classification jointe des lignes et des colonnes d'un tableau

Title: Methods for co-clustering: a review

Vincent Brault[1] et Aurore Lomet[2]

Résumé : La classification croisée vise à identifier une structure sous-jacente existant entre les lignes et colonnes d'un tableau de données. Cette revue bibliographique présente les différents points de vue abordés depuis cinquante ans pour définir cette structure et propose pour chacun un éventail non exhaustif des algorithmes et applications associés. Enfin, les questions encore ouvertes sont abordées et une méthodologie est proposée dans la partie discussion pour analyser des données réelles.

Abstract: Co-clustering aims to identify block patterns in a data table, from a joint clustering of rows and columns. This problem has been studied since 1965, with recent interests in various fields, ranging from graph analysis, machine learning, data mining and genomics. Several variants have been proposed with diverse names: bi-clustering, block clustering, cross-clustering, or simultaneous clustering. We propose here a review of these methods in order to describe, compare and discuss the different possibilities to realize a co-clustering following the user aim.

Mots-clés : classification croisée (co-clustering), classification croisée par blocs (block clustering), classification imbriquée, classification avec chevauchement (biclustering), critère de sélection
Keywords: Cross classification, co-clustering, block clustering, biclustering, selection criterion
Classification AMS 2000 : ,

1. Introduction

La classification non supervisée (simple, Jain et al., 1999; Gan et al., 2007) est une méthode visant à identifier la structure en groupes homogènes interprétables de l'ensemble des lignes (ou des colonnes) d'un tableau de données. Elle est réalisée sur une seule dimension du tableau et chaque classe est alors composée d'éléments se ressemblant et différents de ceux des autres classes (Celeux et Robert, 1993; Maugis et al., 2009).

Dès 1965 (Good, 1965), une extension de ce procédé a été proposée dans le but de réaliser une classification jointe des lignes et des colonnes d'un tableau. L'objectif de cette analyse est l'identification d'une structure sous-jacente en blocs homogènes qui existe entre ces deux ensembles ; on parle de *classification croisée*.

Depuis, la classification croisée a été développée sous plusieurs variantes et son intérêt s'est considérablement accru ces dernières années avec l'arrivée de nombreuses applications comme l'analyse de données marketing (Berkhin, 2006), textuelles (Dhillon et al., 2003), la génomie (Jagalur et al., 2007; Hedenfalk et al., 2001), les systèmes de recommandations (Shan et Banerjee,

[1] AgroParisTech/UMR INRA MIA 518.
E-mail : vincent.brault@agroparistech.fr
[2] CEA, LIST, F-91191 Gif-sur-Yvette, France.
E-mail : aurore.lomet@cea.fr

 ISSN: 2102-6238

2008), les données archéologiques (Leredde et Perin, 1980) ou bien encore en sociologie où des données politiques ont été étudiées par différents auteurs pour retrouver des groupes de politiciens et des enjeux politiques (Wyse et Friel, 2010; Hartigan, 1975). En 2006, elle a d'ailleurs été l'objet d'un challenge, Netflix, dont l'un des objectifs était de réussir à classer simultanément les utilisateurs de cette société de location de dvd et les films (Bennett et Lanning, 2007).

Dans cet article, nous commençons par une définition générale de la classification croisée et l'introduction de quelques notations (section 2). Puis, nous présentons différentes méthodes ainsi que les algorithmes et applications associés. Ces modèles diffèrent notamment par leurs hypothèses de classification allant des plus contraintes aux plus souples : la classification par blocs qui organise l'ensemble des lignes et des colonnes en blocs homogènes (section 3), la classification imbriquée (section 4) et celle avec chevauchement (section 5). Pour finir, nous discutons l'ensemble de ces méthodes et plus généralement la classification croisée (section 6).

2. Définition générale

La classification croisée a pour objectif d'identifier la structure en blocs dont les données sont homogènes et contrastées de celles des autres blocs d'un tableau de données à partir d'une classification jointe des lignes et des colonnes. Une classe est une collection d'éléments du tableau semblables entre eux et dissemblables des éléments des autres classes.

La matrice de données $\mathbf{x} = (x_{ij})_{n\times d}$ est définie par l'ensemble $I = \{1,\dots,n\}$ des lignes des n observations et l'ensemble $J = \{1,\dots,d\}$ des colonnes des d variables. Chaque élément x_{ij} de la matrice représente la valeur binaire, discrète ou continue prise par l'observation i pour la variable j. Avec ces notations, l'objectif de la classification croisée est de définir un ensemble $U = (U_\kappa)_{\kappa=1,\dots,u}$ de u blocs $U_\kappa = (Z_\kappa, W_\kappa)$ où $Z = \{Z_1,\dots,Z_u\}$ est un ensemble de sous-groupes de lignes de I et $W = \{W_1,\dots,W_u\}$ un ensemble de sous-groupes de colonnes de J pouvant être le produit cartésien d'une partition Z de I et d'une partition W de J (*block clustering*), le produit cartésien d'une hiérarchie Z de I et d'une hiérarchie W de J, une partition de $I\times J$, une hiérarchie de $I\times J$ ou simplement un sous-groupe de $I\times J$ (*biclustering*). Les matrices $\mathbf{z} = (z_{ik})_{n\times g}$ et $\mathbf{w} = (w_{j\ell})_{d\times m}$ représentent respectivement les appartenances aux classes en ligne et en colonne où g est le nombre de classes en ligne et m le nombre de classes en colonne. Le bloc U_κ est de forme rectangulaire si les lignes et colonnes sont réordonnées suivant les ensembles Z et W dans la matrice initiale. Par ailleurs, dans la plupart des méthodes présentées dans ce papier, le nombre total u de blocs recherchés est généralement supposé connu et fixé a priori. Une discussion sur ce choix est présentée en section 6.3. Pour les autres méthodes, ce choix est inclus dans l'approche et précisé dans leur présentation.

Par exemple, dans le cas du challenge Netflix, l'ensemble des lignes I représentent les utilisateurs, l'ensemble des colonnes J les films et les éléments du tableau $\mathbf{x}$ les notes (de 1 à 5) attribuées à chaque film par chaque utilisateur. L'objectif de la classification croisée est alors de regrouper les individus ayant un même profil suivant une partition $\mathbf{z}$ avec les films notés de façon analogue suivant une partition $\mathbf{w}$.

 ISSN: 2102-6238

3. Classification croisée par blocs

3.1. Définition

La classification croisée par blocs considère une classification simultanée des lignes et des colonnes utilisant deux partitions (*block clustering*). L'objectif est d'identifier la structure en g blocs homogènes en ligne et en m blocs homogènes en colonne comme représentée en figure 1. Cette dernière représente un tableau de données continues en niveaux de gris avec à gauche le tableau original, au centre le tableau organisé suivant les partitions en ligne et en colonne, à droite le tableau résumé des blocs où sont représentées dans chaque case les moyennes estimées des blocs. Ce type de classification croisée peut s'appliquer aux données binaires (où le tableau résumé contient non plus la moyenne mais la majorité de 1 ou 0), continues ou de contingence se présentant sous la forme d'un tableau à double entrées.

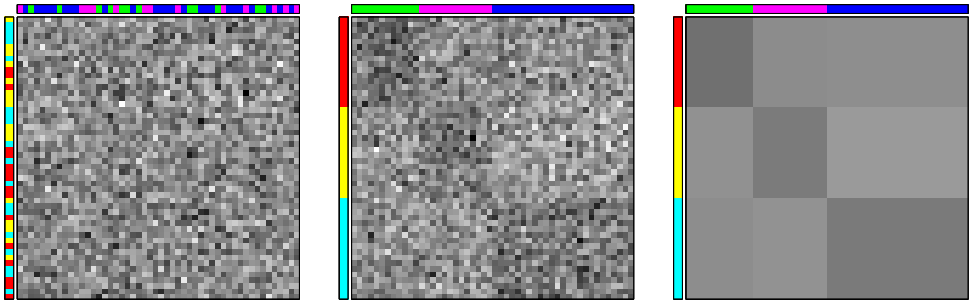


FIGURE 1. *Représentation d'un tableau de données continues en niveaux de gris : à gauche le tableau original, au centre le tableau organisé suivant les partitions en ligne et en colonne, à droite le tableau résumé des blocs (où sont représentées dans chaque case les moyennes estimées des blocs).*

En reprenant les notations précédentes, la détection de blocs homogènes de la matrice $\mathbf{x}$ peut être obtenue en partitionnant les lignes en g classes et les colonnes en m classes. L'objectif est de trouver les partitions Z des n observations en g classes et W des d variables en m classes. Les blocs sont dans ce cas définis par $U = Z \times W$ avec $u = g \times m$.

Pour réaliser une telle classification croisée, certains auteurs (par exemple Lerman et Leredde, 1977) proposent une méthode dont l'algorithme est appliqué séparément et indépendamment sur les deux ensembles mais avec une analyse simultanée des résultats. Ils réalisent ainsi deux classifications simples et donnent une interprétation des blocs. D'autres comme Tishby et al., 1999 réalisent une classification sur un ensemble puis utilisent cette classification pour effectuer celle sur le second. Il s'agit alors d'effectuer sur une dimension du tableau une classification simple ; puis, de réaliser la classification de la seconde dimension en prenant en compte les résultats de la première. Enfin, plusieurs auteurs (Govaert, 1983; Yoo et Choi, 2010; Kluger et al., 2003; Seldin et Tishby, 2010) réalisent une classification simultanée des lignes et des colonnes via différentes méthodes exposées en sections 3.2 et 3.3.

La classification croisée par blocs est une méthode d'apprentissage non supervisée, on distingue deux grandes familles de méthodes mathématiques :

– la reconstruction de matrices (*matrix reconstruction based*), où le problème est formalisé par une approximation de matrice et des mesures de dissimilarité, utilisant différents ensembles

 ISSN: 2102-6238

de contraintes (Govaert, 1977, 1995; Banerjee et al., 2007),
– des modèles probabilistes (*model-based*), dont les modèles de mélange qui utilisent des variables latentes pour définir les classes en ligne et en colonne (Govaert et Nadif, 2003; Shan et Banerjee, 2008; Wyse et Friel, 2010).

3.2. *Méthodes par reconstruction de matrices*

Les méthodes par reconstruction de matrices se basent sur le choix de mesures de distance ou plus généralement sur le choix de mesure de distorsion entre les données. On distingue ainsi plusieurs courants :
– les méthodes « CRO » (Govaert, 1983, 1989),
– les méthodes de *non-negative matrix factorization* (Seung et Lee, 2001),
– la tri-factorisation non-négative (Long et al., 2005),

3.2.1. *Méthodes « CRO »*

Proposées par Govaert (1983), les méthodes « CRO » rassemblent trois algorithmes *Croeuc, Crobin* et *Croki2* qui diffèrent par le type de données observées (quantitatives, binaires ou de contingence respectivement). L'objectif de ces algorithmes est de déterminer le couple de partitions de I et J qui minimise (ou maximise selon le cas) un critère mesurant la qualité d'un couple de partitions et dépendant du type de données.

Dans le cas des données quantitatives, le critère à minimiser par l'algorithme *Croeuc* est celui des moindres carrés. En reprenant les notations précédentes, ce critère s'écrit :

$$F(Z,W) = \sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{k=1}^{g}\sum_{\ell=1}^{m} z_{ik}w_{j\ell}(x_{ij}-\bar{x}_{k\ell})^2 \quad ,$$

où n,d,g,m sont respectivement le nombre de lignes, de colonnes, de classes en ligne et de classes en colonne, z_{ik} et $w_{j\ell}$ sont respectivement les valeurs des matrices indicatrices de l'ensemble des lignes et des colonnes et $\bar{x}_{k\ell} = \frac{1}{n_k n_\ell}\sum_{i,j} z_{ik}w_{j\ell}x_{ij}$ la moyenne empirique du bloc (k,ℓ) avec n_k (respectivement n_ℓ) le nombre de lignes (respectivement de colonnes) dans la classe k (respectivement la classe ℓ).

Cette méthode comme les suivantes suppose que le nombre de classes en ligne et en colonne est connu. Les valeurs de g et m sont donc fixées par l'utilisateur avant l'application de la méthode. Cette question de la détermination du nombre de classes est discutée en section 6.3.

Dans le cas d'un tableau de données binaires, chaque bloc est associé à la valeur 0 ou 1. La table où chaque bloc est représenté par la valeur la plus fréquente est alors appelé le tableau résumé (dans le cas des données continues et de contingence, le tableau résumé est alors la moyenne de chaque bloc). Le critère à minimiser par l'algorithme *Crobin* est alors la mesure entre le tableau initial et le tableau résumé. Dans le cas des données de contingence, l'algorithme *Croki2* cherche à minimiser la perte d'information résultant du regroupement, ce qui est équivalent à déterminer les partitions maximisant le χ^2 de contingence du tableau. Cela revient à maximiser la dépendance entre la partition des lignes et la partition des colonnes.

 ISSN: 2102-6238

Ces algorithmes itératifs définissent une suite de couples de partitions à partir d'un couple initial. Pour ce faire, on fixe une des partitions et on cherche la meilleure partition de l'autre ensemble. Puis, cette dernière étant fixée, on cherche alors la meilleure partition du premier ensemble. Ces étapes sont réalisées jusqu'à convergence des partitions vers un optimum local. Il est nécessaire de lancer ces algorithmes un nombre suffisant de fois avec différentes partitions initiales afin d'éviter les optima locaux.

Se basant sur ces critères, Bock (1979) propose des algorithmes similaires pour des données quantitatives et Dhillon et al. (2003) pour des données de contingence. D'autres types d'algorithmes ont aussi été proposés comme l'algorithme génétique de Hansohm (2002) ou l'algorithme séquentiel de Podani et Feoli (1991) qui cherche à maximiser une différence de distances du χ^2 et qui est appliqué à des données binaires en écologie.

3.2.2. Méthodes de non-negative matrix factorization (NMF)

La NMF [1] (Seung et Lee, 2001), comme les méthodes suivantes utilisant des matrices, permet d'exprimer le problème de la classification croisée dans un cadre algébrique reposant sur l'approximation matricielle sous certaines contraintes appropriées. On suppose que la matrice des observations peut s'écrire sous la forme de deux ou trois matrices (suivant la méthode) qui peuvent être vue comme des variables latentes (aussi vues en section 3.3.1 pour le modèle de blocs latents) utilisées pour l'organisation du tableau en blocs homogènes. L'objectif est alors de réaliser une classification croisée sans prendre en considération d'informations supplémentaires sur les données.

La NMF est généralement utilisée dans les domaines de la reconnaissance de formes et du *text mining*. Les données sont le plus souvent des données de comptage (nombre de mots dans différents textes par exemple) et des données binaires.

L'objectif de cette méthode est de factoriser la matrice des données $\mathbf{x}$ en deux matrices positives $\mathbf{a}$ et $\mathbf{b}$ dont tous les éléments sont positifs de taille $n \times d$ en minimisant :

$$\min_{\mathbf{a}\geq 0, \mathbf{b}\geq 0} ||\mathbf{x} - \mathbf{a}\mathbf{b}^{\mathsf{T}}||^2 \quad ,$$

où $\mathbf{a} \in \mathbb{R}_+^{n\times g}, \mathbf{b} \in \mathbb{R}_+^{d\times g}$ sont deux matrices arbitraires non-négatives et $^{\mathsf{T}}$ est la notation de transposée. Pour ce faire, on optimise la distance entre $\mathbf{x}$ et les matrices non-négatives. La convergence de cet algorithme est assurée, et l'unicité de la solution peut être acquise en imposant une contrainte que la longueur euclidienne du vecteur colonne de la matrice $\mathbf{a}$ soit égale à 1.

Une méthode similaire, la *non-negative matrix tri-factorization* (NTF), vise à factoriser la matrice initiale $\mathbf{x}$ en $\mathbf{z}\mathbf{a}\mathbf{w}^{\mathsf{T}}$ où $\mathbf{a} \in \mathbb{R}_+^{g\times m}$ est une matrice positive, $\mathbf{z}$ et $\mathbf{w}$ sont les matrices binaires représentant les appartenances des observations aux classes en ligne et en colonne respectivement que l'on retrouve également dans la méthode *CRO*. Le produit de ces trois matrices peut être vu comme un résumé de la matrice initiale $\mathbf{x}$. Comme précédemment, l'objectif de cette méthode est de minimiser l'écart quadratique :

[1] Cette méthode est implémentée dans le package *NMF* pour le logiciel R, https://cran.r-project.org/web/packages/NMF/index.html.

 ISSN: 2102-6238

$$\min_{\mathbf{z}\geq 0,\mathbf{a}\geq 0,\mathbf{w}\geq 0} ||\mathbf{x}-\mathbf{z}\mathbf{a}\mathbf{w}^{\top}||^2 \ ,$$

où $\mathbf{a}$ est une matrice arbitraire et $\mathbf{z}$ et $\mathbf{w}$ sont deux matrices des appartenances aux classes.

Cette minimisation peut être réalisée par un algorithme itératif alternant des procédures d'optimisation des moindres carrés. Cette fonction d'optimisation est convexe en $\mathbf{z}$, $\mathbf{a}$, $\mathbf{w}$ respectivement mais pas en considérant l'ensemble de ces trois termes. L'algorithme converge par conséquent vers un minimum local.

3.2.3. Méthodes de non-negative block value decomposition (NBVD)

Pour pallier le problème de convergence locale de la méthode NTF, Long et al. (2005) proposent la méthode de *non-negative block value decomposition* qui a la même fonction objectif que la méthode NTF mais sous certaines contraintes multiplicatives de mise à jour telles que :

$$\begin{aligned} z_{ij} &<- z_{ij}\frac{(\mathbf{x}\mathbf{w}\mathbf{a}^T)_{ij}}{(\mathbf{z}\mathbf{a}\mathbf{w}^T\mathbf{w}\mathbf{a}^T)_{ij}} \ , \\ a_{ij} &<- a_{ij}\frac{(\mathbf{z}^T\mathbf{x}\mathbf{w})_{ij}}{(\mathbf{z}^T\mathbf{z}^T\mathbf{a}\mathbf{w}^T\mathbf{w})_{ij}} \ , \\ w_{ij} &<- w_{ij}\frac{(\mathbf{x}^T\mathbf{z}\mathbf{a})_{ij}}{(\mathbf{w}\mathbf{a}^T\mathbf{z}^T\mathbf{z}\mathbf{a})_{ij}} \ . \end{aligned}$$

Cette décomposition de la matrice initiale n'est pas unique. Elle ne permet donc pas de fournir directement une classification croisée des classes : une normalisation du produit des matrices $\mathbf{z}\mathbf{a}$ par $\mathbf{z}\mathbf{a}\mathbf{w}^T$ est notamment nécessaire afin d'obtenir les classes. Cette méthode testée empiriquement sur quelques jeux de données montre une amélioration de la précision de la classification par rapport aux méthodes classiques de NMF. Par ailleurs, pour cette méthode comme pour les autres présentées dans cette section, la principale difficulté consiste à choisir a priori une mesure de distance appropriée. Comme pour la NMF, ce choix arbitraire influence bien sûr les résultats.

3.3. Méthodes par modèle probabiliste

La seconde famille de méthodes est basée sur l'utilisation de modèles probabilistes nécessitant des hypothèses portant sur l'origine des données. Ces hypothèses permettent la conception d'un modèle statistique où les paramètres sont alors estimés sur la base des données fournies.

En classification simple (ici présentée sur l'ensemble des lignes), une des approches les plus classiques et performantes est l'utilisation des modèles de mélange (Celeux et Govaert, 1992; McLachlan, 1982; Banfield et Raftery, 1993). Les données sont supposées être générées par un mélange de distributions de probabilité où chaque composante k du mélange représente une classe et dont la vraisemblance du modèle est définie par :

$$p(\mathbf{x};\theta) = \prod_i \sum_k \pi_k \varphi(x_i;\alpha_k)$$

 ISSN: 2102-6238

où $\pi = (\pi_1, \ldots, \pi_k, \ldots, \pi_g)$ sont les paramètres de proportion des classes ; $\theta = (\pi, \alpha)$ est la notation résumé de l'ensemble des paramètres du modèle et $\varphi(x_i; \alpha_k)$ la densité des éléments de la matrice connaissant leur appartenance aux classes. La matrice des données est alors supposée être un échantillon indépendant et identiquement distribué $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ où $\mathbf{x}_i$ est un vecteur en ligne. Généralement, ces densités suivent une même famille paramétrique. L'estimation des paramètres et, par conséquent, celle des classes sont souvent réalisées par l'algorithme EM (*Expectation-Maximisation* Dempster et al., 1977) [2].

Pour la classification croisée, plusieurs modèles probabilistes dont certains s'inspirant de la classification simple et des modèles de mélange sont proposés pour représenter les structures en blocs homogènes des données.

3.3.1. Modèle de blocs latents, algorithmes d'estimation et extensions

Inspirés par les modèles de mélange pour la classification simple, Govaert et Nadif (2003) proposent le modèle de blocs latents qui est une extension du modèle de mélange classique. Cette méthode considère que (i) les matrices d'affectation aux classes en ligne $\mathbf{z}$ et colonne $\mathbf{w}$ sont des variables latentes à estimer, (ii) chaque élément du tableau est issu d'une distribution de probabilité conditionnée par son appartenance à sa classe en ligne et en colonne et (iii) les appartenances aux classes sont supposées a priori indépendantes et conditionnellement à la connaissance de ces appartenances, les éléments du tableau sont supposés indépendants.

La matrice d'affectation aux classes en ligne $\mathbf{z} = (z_{ik}; i = 1, \ldots, n; k = 1, \ldots, g)$ représente la partition des lignes en g classes. Les éléments z_{ik} de cette matrice sont égaux à 1 si la ligne i appartient à la classe k et 0 sinon. De même pour les colonnes, la matrice d'affectation aux classes $\mathbf{w} = (w_{j\ell}; j = 1, \ldots, d; \ell = 1, \ldots, m)$ représente la partition des colonnes en m classes. La vraisemblance de ce modèle s'écrit :

$$p(\mathbf{x}; \theta) = \sum_{(\mathbf{z},\mathbf{w}) \in \mathscr{Z} \times \mathscr{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,\ell} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell} \varphi(x_{ij}, \alpha_{k\ell})^{z_{ik} w_{j\ell}} \quad ,$$

où $\mathscr{Z}$, $\mathscr{W}$ représentent les ensembles de toutes les affectations possibles $\mathbf{z}$ de I et $\mathbf{w}$ de J ; $\pi = (\pi_1, \ldots, \pi_k, \ldots, \pi_g)$ et $\rho = (\rho_1, \ldots, \rho_\ell, \ldots, \rho_m)$ sont respectivement les vecteurs de probabilités d'appartenance aux classes en ligne et en colonne ; et $\theta = (\pi, \rho, \alpha = (\alpha_{k\ell}))$ est la notation résumée de tous les paramètres du modèle. Ce modèle possède trois déclinaisons en fonction du type de données. La probabilité conditionnelle $\varphi(x_{ij}, \alpha_{k\ell})$ est alors une gaussienne pour la modélisation de données continues (Govaert et Nadif, 2009), de Bernoulli pour les données binaires (Govaert et Nadif, 2008) et de Poisson pour les données de contingence (Govaert et Nadif, 2007) [3].

Pour estimer les paramètres de ces modèles, l'algorithme EM n'est pas directement applicable comme en classification simple. En effet, le calcul de la vraisemblance du modèle de blocs latents est un problème combinatoire nécessitant $g^n \times m^d$ opérations du fait de l'interdépendance des données (où n, d, g et m sont respectivement le nombre de lignes, de colonnes, de classes en

[2] Cette méthode est implémentée dans les packages *Mclust* et *RMixmod* pour le logiciel R, https://cran.r-project.org/web/packages/mclust/index.html, https://cran.r-project.org/web/packages/Rmixmod/index.html.

[3] Cette méthode est implémentée dans le package *blockcluster* pour le logiciel R et dans le logiciel *Mixmod*, https://cran.r-project.org/web/packages/blockcluster/index.html, http://www.mixmod.org/?lang=fr.

 ISSN: 2102-6238

ligne et de classes en colonne). Pour pallier ce problème, Govaert et Nadif (2008) proposent un algorithme d'estimation à partir de l'algorithme EM en incluant une approximation variationnelle : les probabilités a posteriori des partitions sont supposées indépendantes. Dans chaque cas du modèle de blocs latents (gaussien, de Bernoulli ou de Poisson), un algorithme VEM (*Variational Expectation Maximization*) a ainsi été développé pour estimer les paramètres du modèle. Muni de cette hypothèse simplificatrice, cet algorithme se base sur la vraisemblance L_c des données complétées $(\mathbf{x},\mathbf{z},\mathbf{w})$ définies par :

$$\begin{aligned} L_c(\mathbf{x},\mathbf{z},\mathbf{w};\theta) &= \log p(\mathbf{x},\mathbf{z},\mathbf{w};\theta) \\ &= \sum_{i,k} z_{ik}\log\pi_k + \sum_{j,\ell} w_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell} z_{ik}w_{j\ell}\log p_{k\ell}(x_{ij};\alpha) \quad , \end{aligned}$$

où $p_{k\ell}$ est la densité de distribution du bloc (k,ℓ). L'objectif est alors de maximiser le critère F_c pour obtenir l'approximation variationnelle de la vraisemblance $\tilde{L}$ avec :

$$\tilde{L}(\theta) = \underset{\mathbf{s},\mathbf{t}}{\operatorname{argmax}}\, F_c(\mathbf{s},\mathbf{t},\theta) \quad ,$$

où

$$F_c(\mathbf{s},\mathbf{t},\theta) = L_c(\mathbf{s},\mathbf{t},\theta) + H(\mathbf{s}) + H(\mathbf{t}) \quad ,$$

avec $s_{ik} = p_{\mathbf{z}}(z_{ik}=1)$, $t_{j\ell} = p_{\mathbf{w}}(w_{j\ell}=1)$,
$L_c(\mathbf{s},\mathbf{t};\theta) = \sum_{i,k} s_{ik}\log\pi_k + \sum_{j,\ell} t_{j\ell}\log\rho_\ell + \sum_{i,j,k,\ell} s_{ik}t_{j\ell}\log p_{k\ell}(x_{ij};\alpha_{k\ell})$, $H(\mathbf{s}) = -\sum_{i,k} s_{ik}\log s_{ik}$ et $H(\mathbf{t}) = -\sum_{j,\ell} t_{j\ell}\log t_{j\ell}$.

L'algorithme VEM donne en sortie l'approximation variationnelle de la vraisemblance $\tilde{L}$, les estimations des paramètres $\hat{\theta}$ et les matrices de probabilités des appartenances aux classes $\mathbf{s}$ et $\mathbf{t}$. Pour obtenir les matrices d'appartenances $\mathbf{z}$ et $\mathbf{w}$ (également utilisées dans les méthodes *CRO* et NTF, présentées précédemment), Govaert et Nadif (2008) proposent d'appliquer la règle du maximum a posteriori (MAP).

D'autres algorithmes issus de l'algorithme EM ont aussi été proposés pour l'estimation des paramètres du modèle de blocs latents pour pallier le problème de dégénérescence des classes de l'algorithme VEM. En effet, dans certains cas, l'algorithme VEM retourne des classes vides. Pour résoudre ce problème, une régularisation bayésienne de cet algorithme nommé V-Bayes a été proposée par Keribin et al. (2012, 2013) pour les données binaires et de contingence. D'autres algorithmes comme CEM (*Classification Expectation Maximization*) ont aussi été développés par ces mêmes auteurs. À la différence de l'algorithme VEM, ce dernier cherche à maximiser la vraisemblance des données complétées. L'algorithme SEM (*Stochastic Expectation Maximization* Keribin et al., 2010) estime quant à lui la vraisemblance à partir d'un échantillonneur de Gibbs.

Le modèle de blocs latents et l'algorithme VEM ont été, par ailleurs, utilisés par Lashkari et Golland (2009) qui l'emploient comme modèle génératif et par Jagalur et al. (2007) pour étudier des données d'expression de gènes.

Plusieurs approches bayésiennes basées sur ce modèle ont été récemment développées. Shan et Banerjee (2008) et Van Dijk et al. (2009) proposent une approche bayésienne pour estimer les paramètres. Le premier utilise une approximation variationnelle alors que le second emploie un algorithme basé sur l'échantillonneur de Gibbs plus coûteux en temps de calculs. Meeds et Roweis (2007) montrent quant à eux que ces modèles bayésiens prennent facilement en compte les données manquantes et sont robustes pour des forts taux de données manquantes.

 ISSN: 2102-6238

Enfin, Deodhar et Ghosh (2007) étendent le modèle pour la prise en compte d'attributs sur les lignes et les colonnes pour une analyse croisée des clients et des produits en marketing. Shafiei et Milios (2006) proposent quant à eux un modèle probabiliste autorisant le chevauchement des blocs pour les distributions régulières de la famille exponentielle.

3.3.2. Autres modèles probabilistes

Rooth (1995) propose un modèle probabiliste pour la classification en blocs de données de contingence avec une structure en diagonale et définit un algorithme utilisant des formules similaires aux formules d'estimation de Baum-Welch pour les chaînes de Markov cachées.

Hartigan (2000) étudie la classification croisée des votes aux congrès des États-Unis. Les sénateurs sont partitionnés en blocs et les mesures législatives sont partitionnées en types. Pour ce faire, il propose un modèle probabiliste pour réaliser une classification croisée de données binaires et la probabilité de chaque bloc et type de la partition finale est estimée par une méthode de Monte Carlo par chaînes de Markov.

Un autre modèle qui diffère de par son application et par son type est proposé par Nowicki et Snijders (2001). Ce modèle de graphes aléatoires (*Stochastic Block Model*) construit un modèle de mélange pour les relations entre les objets et identifie les classes latentes via une inférence a posteriori. Ce modèle pour graphe suppose que les sommets sont répartis en plusieurs classes (latentes) et que la distribution de probabilité de la relation en deux sommets ne dépend que de la classe à laquelle ils appartiennent. Ces auteurs illustrent leur méthode [4] par un exemple tiré des réseaux sociaux (*Kapferer's tailor shop*). Un état de l'art sur ce modèle est proposé par Matias et Robin (2014).

Kemp et al. (2006) présentent un modèle infini relationnel qui découvre des structures stochastiques de données relationnelles sous la forme d'observations binaires. Ce modèle non paramétrique bayésien est appliqué à quatre types de problèmes : classer les objets et leurs caractéristiques, découvrir des systèmes de parenté, découvrir la structure de données politiques et pour des ontologies d'apprentissage.

4. Classification imbriquée

La condition que les blocs de U soient le résultat du produit cartésien d'une partition de I et d'une partition de J peut être trop contraignante. Sur la figure 2 est représentée une matrice réorganisée sous les conditions des modèles précédents. Les blocs de la cinquième classe en colonne sont distingués par les classes en lignes alors qu'ils pourraient être regroupés en une seule colonne commune à toutes les lignes. Le modèle serait ainsi plus parcimonieux. Dans le cas de Netflix, cela signifie que les films de cette classe plaisent de la même façon à tout le monde tandis que les appréciations d'autres films vont dépendre des utilisateurs.

Par ailleurs, il peut parfois être intéressant de ne chercher que les blocs les plus contrastés en ne classant pas les autres cases. Par exemple, Ben-Dor et al. (2003) ont fait ressortir un groupe de gènes ayant des comportements similaires face à une mutation de la tumeur du cancer du sein.

[4] Certaines méthodes relatives à ce modèle sont implémentées dans le logiciel *Wmixnet* http://www.agroparistech.fr/mia/productions:logiciel:wmixnet

 ISSN: 2102-6238

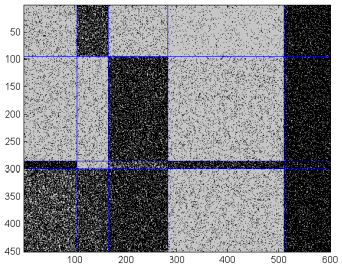


FIGURE 2. *Représentation d'une matrice réorganisée par le modèle des blocs latents.*

Dans cette perspective, la classification imbriquée (appelée aussi hiérarchique) autorise pour deux blocs $U_k = (Z_k, W_k)$ et $U_{k'} = (Z_{k'}, W_{k'})$ une hiérarchie possible dans les groupes de classification :

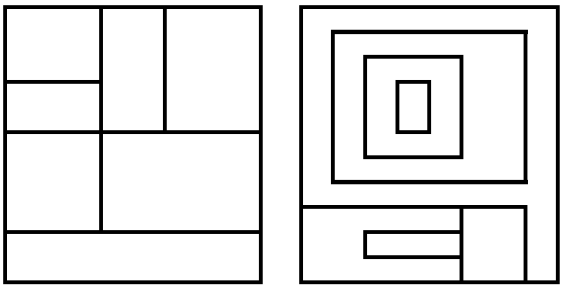
– Soit sur les lignes et/ou les colonnes (figure 3 (a)) :

$$\begin{aligned} Z_k \cap Z_{k'} \neq \emptyset \quad &\Rightarrow \quad Z_k \subseteq Z_{k'} \text{ ou } Z_{k'} \subseteq Z_k \\ \text{et } W_k \cap W_{k'} \neq \emptyset \quad &\Rightarrow \quad W_k \subseteq W_{k'} \text{ ou } W_{k'} \subseteq W_k. \end{aligned}$$

– Soit sur les blocs entièrement (figure 3 (b)) :

$$U_k \cap U_{k'} \neq \emptyset \quad \Rightarrow \quad U_k \subseteq U_{k'} \text{ ou } U_{k'} \subseteq U_k.$$

La différence avec la section 3 étant que l'inclusion stricte est désormais autorisée. Dans cette partie, nous abordons le cas de la classification imbriquée sur les lignes et/ou les colonnes puis la classification imbriquée sur les blocs.



(a) (b)

FIGURE 3. *Représentation schématique de deux classifications imbriquées : (a) est une classification imbriquée sur les lignes et les colonnes et (b) est une classification imbriquée sur les blocs.*

 ISSN: 2102-6238

4.1. *Classification imbriquée sur les lignes et les colonnes*

La première configuration consiste à faire des découpes successives horizontalement ou verticalement pour séparer des blocs en deux parties et est appelée *Direct Splitting* (voir Hartigan, 1975, Chapitre 14). Toutes les données du tableau seront donc classées. Dans la suite, nous proposons deux points de vue.

4.1.1. *Algorithme déterministe*

Pour trouver une telle classification, des algorithmes déterministes sont proposés. Hartigan (1975) introduit ceux de *fractionnement direct* (*Splitting Algorithm*) cherchant à séparer les blocs tout en minimisant la variance de chacun des deux nouveaux blocs obtenus ; il s'arrête lorsque chaque bloc possède une variance inférieure à une valeur donnée par l'utilisateur. L'algorithme de *One-Way Splitting Algorithm* considère pour chaque bloc des vecteurs colonnes (ou lignes) alors que l'algorithme *Two-Way Splitting Algorithm* considère toutes les cases des blocs de la même façon. Ces algorithmes ont été testés sur le pourcentage de votes pour le candidat présidentiel républicain dans six états du sud des Etats-Unis pour six années différentes (tableau 1, à gauche pour l'algorithme *One-Way Splitting* en colonne et à droite pour l'algorithme *Two-Way Splitting*) pour une variance maximale de 25. Nous pouvons voir pour le tableau de gauche que trois états (KY, MD et MO) ont été regroupés puis que les trois autres se différencient sauf sur certaines années ; il n'est pas nécessaire dans ce cas de diviser le bloc inférieur car la variance est calculée pour chaque colonne du bloc. En revanche, l'algorithme *Two-Way Splitting* est obligé de séparer certaines années car il compare toutes les cases du bloc.
Duffy et Quiroz (1991) proposent une amélioration de ces algorithmes en autorisant les permutations des lignes et des colonnes.

TABLE 1. *Classification par l'algorithme One-Way Splitting (à gauche) et l'algorithme Two-Way Splitting (à droite) du pourcentage de votants pour le candidat républicain aux présidentielles pour* 6 *années dans* 6 *états du sud des États-Unis : Mississippi (MS), Caroline du Sud (SC), Louisiane (LA), Kentucky (KY), Maryland (MD) et Missouri (MO). Contrairement à l'algorithme One-Way Splitting qui ne regarde que la variance sur chaque colonne d'un bloc, l'algorithme Two-Way Splitting est obligé de procéder à de nouvelles divisions.*

	32	36	40	60	64	68
MS	4	3	4	25	87	14
SC	2	1	4	49	59	14
LA	7	11	14	29	57	23
KY	40	40	42	54	36	44
MD	36	37	41	46	35	42
MO	35	38	48	50	36	45

	32	36	40	60	68	64
MS	4	3	4	25	14	87
SC	2	1	4	49	14	59
LA	7	11	14	29	23	57
KY	40	40	42	54	44	36
MD	36	37	41	46	42	35
MO	35	38	48	50	45	36

4.1.2. *Algorithme probabiliste*

D'un point de vue probabiliste, Roy et Teh (2008) proposent un modèle appelé *processus Mondrian* basé sur une généralisation du processus de Dirichlet (Robert, 2006) utilisé dans le cadre des mélanges simples. Il cherche alors à estimer les découpes par des algorithmes MCMC et

 ISSN: 2102-6238

Metropolis-Hasting. Wang et al. (2011) expliquent que ce modèle permet de diminuer le nombre de coupures en comparaison au modèle des blocs latents. Roy et Teh (2008) ont testé leurs algorithmes sur les échanges commerciaux et diplomatiques entre 24 pays. Ils ont pu retrouver que les pays échangeaient essentiellement avec leurs voisins.

4.2. Classification imbriquée par blocs

La classification imbriquée par bloc ne regarde plus les lignes, ni les colonnes dans leur globalité. Sur la figure 4 est représenté un tableau avec deux regroupements différents. La figure 4 (a) représente le tableau initial, la figure 4 (b) un regroupement par une méthode de fractionnement sans permutation des lignes ou des colonnes et la figure 4 (c) un regroupement imbriqué par blocs. Pour isoler le groupe de 2, la méthode par fractionnement sépare l'ensemble des 1 contrairement à la méthode imbriquée par bloc. Dans ce type de classification, certaines données peuvent ne pas être utilisées.

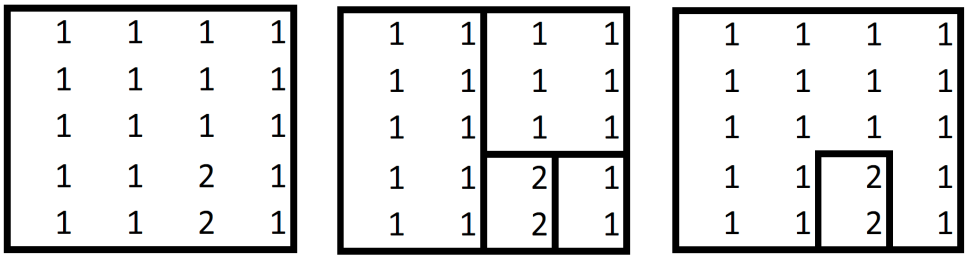


FIGURE 4. *Différence entre un regroupement de fractionnement et imbriqué : à gauche le tableau initial, au milieu une partition de fractionnement et à droite un regroupement imbriqué.*

Dans le cas de la classification imbriquée restreinte (figure 5 (b)), on oblige les blocs à être imbriqués c'est-à-dire que pour tout $\kappa \geq 1$, $U_\kappa \subseteq U_{\kappa-1}$ où U_0 représente la partition initiale alors que dans le cas non restreint, on peut avoir différents groupes de blocs (figure 5 (a)).

Pour la classification non restreinte, Hartigan (1975, Chapitre 15) propose l'algorithme *Two-Way Joining Algorithm* reprenant le principe de fractionnement direct mais adapté aux blocs plutôt qu'aux lignes.

La classification restreinte peut être traitée comme dans le cas de la recherche d'un bloc atypique (voir section 5) où à chaque itération κ, le bloc U_κ obtenu à la κ ème itération est vu comme la matrice à étudier.

5. Classification avec chevauchement

Le dernier type de classification possible est celle par chevauchement (figure 6) qui est la moins restreinte ; le but étant la recherche d'un ou plusieurs blocs atypiques sans classer l'ensemble du tableau contrairement aux méthodes exposées dans la section 3. Cette classification est proche de la méthode imbriquée (section 4.2) à la différence que les blocs peuvent se chevaucher. La seule

 ISSN: 2102-6238

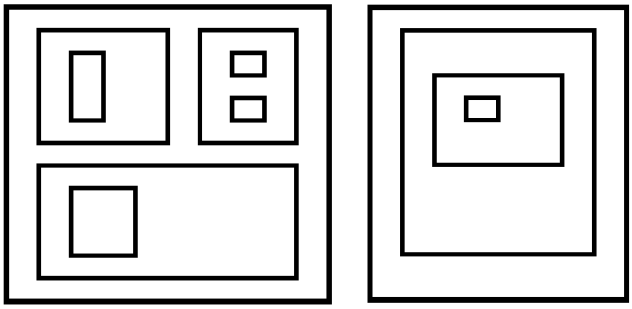


FIGURE 5. *Différence schématique entre un regroupement imbriqué non restreint (à gauche) et restreint (à droite).*

condition pour deux blocs U_κ et $U_{\kappa'}$ est qu'ils ne soient pas identiques :

$$U_\kappa \cap U_{\kappa'} \neq \emptyset \Rightarrow U_\kappa \neq U_{\kappa'}.$$

Une représentation schématique représentant tous les blocs d'un seul tenant n'est possible que si la réorganisation des lignes et des colonnes s'y prêtent. Souvent, lorsqu'en permutant les lignes et les colonnes, l'un des blocs est bien de forme rectangulaire, les autres sont fragmentés (comme pour les blocs magenta et jaune de la figure 6).

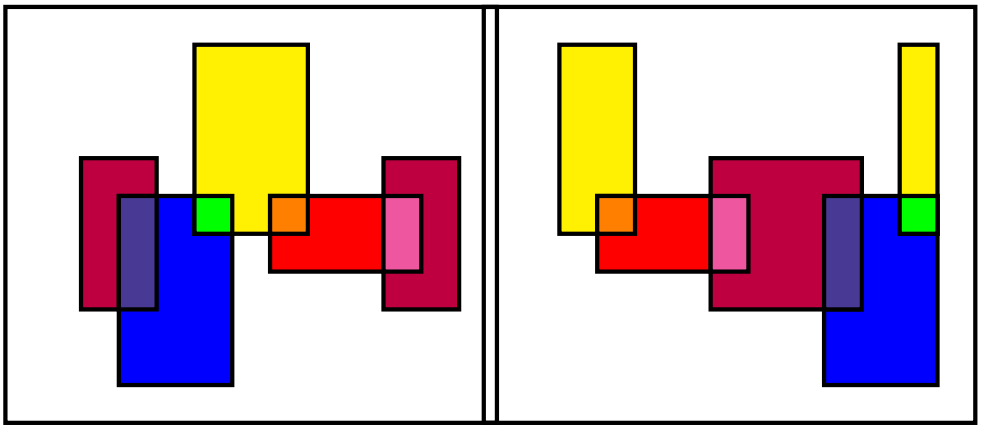


FIGURE 6. *Représentations schématiques d'une classification avec chevauchement (chaque bloc est symbolisé par une couleur) suivant deux réorganisations des colonnes. Aucune permutation ne permet d'avoir une représentation de chacun des quatres blocs d'un seul tenant en même temps.*

Le plus souvent, les blocs recherchés sont homogènes comme dans les classifications précédentes (voir figure 7 (b)) ou ayant une progression arithmétique (figure 7 (a)) ou géométrique sur les lignes et/ou les colonnes mais dans certains cas, la recherche se fait sur une structure ordonnée (comme pour l'algorithme *OPSM* présenté en section 5.4). Madeira et Oliveira (2004) proposent dans leur revue bibliographique une section dédiée aux différentes configurations possibles pour les blocs.

 ISSN: 2102-6238

Sur la figure 7 sont représentées deux parties de matrices avec deux blocs se chevauchant soit parce que leur intersection peut appartenir aussi bien à l'un des blocs qu'à l'autre (comme sur la (a)), soit car ils ont une partie différente (comme sur la (b)). Les classifications vues jusqu'à présent empêchent ce chevauchement.

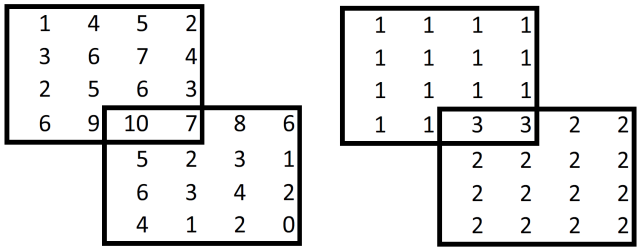


FIGURE 7. *Deux classifications avec chevauchement. Celle de gauche est composée de blocs suivant une progression arithmétique sur les lignes et colonnes avec l'intersection faisant le lien entre les deux. Celle de droite comporte des blocs à valeurs constantes avec l'intersection considérée comme un bloc à part.*

De nombreux algorithmes sont proposés, Prelić et al. (2006) font une comparaison de cinq d'entre eux (*SAMBA*, *ISA*, *OPSM*, *xMotif* et les algorithmes de Cheng et Church, 2000) et Hanczar et Nadif (2010) proposent une méthode basée sur du bootstrap utilisant ces algorithmes.
Dans cette section, nous nous limitons à certains d'entre eux et mettons en appendice A un tableau non exhaustif contenant des algorithmes dédiés en particulier à la classification par chevauchement. D'abord, la δ-classification sera traitée pour mieux comprendre les problèmes algorithmiques sous-jacents provenant de ce relâchement des hypothèses. Dans un second temps, une vision probabiliste sera abordée et enfin, des approches spécifiques pour le cas binaire et le cas de données ordonnées seront présentées.

5.1. δ-classification

Pour répondre au problème de classification par chevauchement, Cheng et Church (2000) introduisent une fonction de coût H des résidus quadratiques d'un bloc $U_\kappa = (Z_\kappa, W_\kappa)$:

$$H(U_\kappa) = \frac{1}{|Z_\kappa||W_\kappa|} \sum_{i \in Z_\kappa, j \in W_\kappa} (x_{ij} - x_{iW_\kappa} - x_{Z_\kappa j} + x_{Z_\kappa W_\kappa})^2$$

avec

$$x_{iW_\kappa} = \frac{1}{|W_\kappa|} \sum_{j \in W_\kappa} x_{ij},\ x_{Z_\kappa j} = \frac{1}{|Z_\kappa|} \sum_{i \in Z_\kappa} x_{ij} \text{ et } x_{Z_\kappa W_\kappa} = \frac{1}{|Z_\kappa||W_\kappa|} \sum_{i \in Z_\kappa, j \in W_\kappa} x_{ij}.$$

Ce critère étant nul pour des blocs contenant des progressions arithmétiques en ligne et en colonne, le minimiser reviendrait dans beaucoup de cas à sélectionner des blocs ne contenant qu'une seule case. Cheng et Church (2000) proposent donc plusieurs algorithmes cherchant le plus grand bloc U_κ tel que $H(U_\kappa) \leq \delta$ où $\delta > 0$ est un paramètre donné. On parle de δ-*classification*. L'algorithme *Brute-Force Deletion and Addition* calcule pour chaque bloc U possible sa valeur

 ISSN: 2102-6238

$H(U)$ en commençant par la matrice initiale puis tous les blocs avec une ligne ou une colonne en moins, puis avec deux et ainsi de suite jusqu'à obtenir le plus grand bloc de coût inférieur à δ. Cet algorithme est optimal mais la complexité étant de l'ordre $\mathcal{O}(2^{nd})$, il ne peut s'appliquer qu'à des petites matrices.
Deux versions plus rapides basées respectivement sur un algorithme de type *forward* et *backward* consistent à partir de la matrice initiale en supprimant une ligne ou une colonne en faisant diminuer la fonction de coût le plus vite possible (algorithme *Multiple Node Deletion*) ou de partir de rien et d'ajouter à chaque étape une ligne ou une colonne pour faire augmenter le moins possible la fonction de coût (algorithme *Multiple Node Addition*). Ces algorithmes ont une complexité de l'ordre $\mathcal{O}((nd)^2)$ mais convergent vers des maxima locaux.
Enfin, Cheng et Church (2000) proposent un algorithme type *forward*/*backward* alternant les deux précédents. Une fois un premier bloc U_1 obtenu, les valeurs des cases de U_1 sont aléatoirement tirées par un tirage aléatoire et la recherche d'un nouveau bloc recommence jusqu'à obtenir le nombre u de blocs désiré. Cet algorithme a été testé sur l'expression du cycle cellulaire de la levure Saccharomyces cerevisiae.
Sur la même hypothèse, Yang et al. (2002) proposent l'algorithme *FLOC* (pour *FLexible Overlapped Clustering*) utilisant la même procédure de type *forward*/*backward* mais directement sur les u blocs désirés (plutôt que d'en sélectionner un premier, puis un second...). Comme les valeurs des cases des blocs obtenus ne sont plus remplacées par un tirage aléatoire, cet algorithme est moins dépendant du hasard que celui de Cheng et Church (2000). De plus, il possède une complexité en $\mathcal{O}(ndu)$ et est moins aléatoire.
Toutefois, les algorithmes de δ-classification sont peu robustes face aux bruits, préférant, dans le cas de données simulées, des blocs plus grands et incluant ceux souhaités (voir Prelić et al., 2006).

5.2. *Modèle probabiliste*

Dans leur article, Lazzeroni et Owen (2000) proposent une méthode probabiliste [5] pour trouver les u blocs (avec u inconnu). Ce modèle suppose que les blocs $U_\kappa = (Z_\kappa, W_\kappa)$ peuvent éventuellement se chevaucher et que chaque bloc possède une structure propre, c'est-à-dire qu'il existe une fonction $f_{\theta_\kappa}(i,j)$ rattachée à ce bloc (par exemple, $f_{\theta_\kappa}(i,j)$ est constante sur tout un bloc). Ce modèle traite ainsi tous les cas évoqués en début de section 5. Une case x_{ij} appartient à un, plusieurs ou aucun bloc et est le résultat du tirage aléatoire d'une variable X_{ij} dont la loi dépend des effets cumulés des blocs auxquels elle appartient :

$$X_{ij} = \sum_{\kappa=1}^{u} f_{\theta_\kappa}(i,j) z_{i\kappa} w_{j\kappa} + \varepsilon_{ij}$$

avec $z_{i\kappa}$ (resp $w_{j\kappa}$) valant 1 si la $i$ème ligne appartient à Z_κ (resp la $j$ème colonne appartient à W_κ) et ε_{ij} des variables aléatoires indépendantes de mêmes lois (le plus souvent gaussienne centrée). Ce modèle prend en compte le cas de la figure 7 (b) en considérant les $f_{\theta_\kappa}(i,j)$ constant par blocs. Contrairement à la section 3.3.1, nous ne fixons aucune contrainte sur les $z_{i\kappa}$ et les $w_{j\kappa}$ qui pourront, par exemple, valoir un pour toute une ligne de la matrice $\mathbf{z}$ ou la matrice $\mathbf{w}$.

[5] Cette méthode est implémentée dans le package *biclust* pour le logiciel R : http://cran.r-project.org/web/packages/biclust/index.html

 ISSN: 2102-6238

Dans le cas d'un bruit symétrique, l'idée est alors de sélectionner les blocs et paramètres minimisant les moindres carrés :

$$\sum_{i=1}^{n}\sum_{j=1}^{d}\left(x_{ij}-\sum_{\kappa=1}^{u} f_{\theta_\kappa}(i,j)z_{i\kappa}w_{j\kappa}\right)^2.$$

Ceci n'est pas réalisable en un temps raisonnable sachant de plus que u est inconnu. Lazzeroni et Owen (2000) proposent un algorithme itératif cherchant le κème bloc à partir des $\kappa-1$ précédents en minimisant par rapport au κ ème bloc et au paramètre θ_κ :

$$\sum_{i=1}^{n}\sum_{j=1}^{d}\left(x_{ij}^{\kappa-1}-f_{\theta_\kappa}(i,j)z_{i\kappa}w_{j\kappa}\right)^2$$

où

$$x_{ij}^{\kappa-1}=x_{ij}-\sum_{\kappa'=1}^{\kappa-1} f_{\theta_{\kappa'}}(i,j)z_{i\kappa'}w_{j\kappa'}$$

représente la matrice où les valeurs des blocs précédents ont été enlevées (si une case x_{ij} dépend uniquement des blocs déjà trouvés, $x_{ij}^{\kappa-1}$ est le résultat du tirage de ε_{ij}). L'algorithme s'arrête lorsque les cases $x_{ij}^{\kappa-1}$ du dernier bloc obtenu sont les résultats de tirages des variables aléatoires ε_{ij}.
Lazzeroni et Owen (2000) testent leurs algorithmes sur des données nutritionnelles de 961 aliments, ce qui a permis de séparer certains groupes d'aliments suivant différentes variables nutritionnelles (par exemple, certains parmesans, parties du boeuf et parties du poulet ont été séparés car possédant plus de protéines et moins de cholestérol que d'autres aliments) ; ils ont aussi étudié les valeurs mensuelles de taux de change entre différents pays séparant 71 mois où le dollar l'emportait sur 11 autres devises par exemple ; ou encore sur des données d'expression de gènes de levures suivant différentes conditions expérimentales mettant en évidence 34 blocs se chevauchant avec certains gènes se retrouvant dans 18 groupes.

5.3. Samba

Dans le cas de matrices binaires, Tanay et al. (2002) définissent un algorithme [6] probabiliste appelé *SAMBA*. Le tableau représente le graphe bipartie des relations entre les lignes et les colonnes. Ainsi, les cases x_{ij} valent 1 si les sommets i et j sont reliés et 0 sinon. Le but est de chercher un sous-graphe à la fois grand et dense.
L'idée de base est de supposer que chaque case est le résultat d'un tirage aléatoire d'une loi de Bernoulli de paramètre p la proportion de cases valant 1 dans la matrice. L'objectif est de sélectionner un bloc U_κ contenant une grande proportion de 1 tout en étant le plus grand possible. En première approche, Tanay et al. (2002) proposent d'utiliser une fonction de coût H_p calculant, pour chaque bloc $U_\kappa=(Z_\kappa,W_\kappa)$, la valeur de la queue d'une loi binomiale de paramètres $|Z_\kappa||W_\kappa|$ et p à partir de la somme des cases du bloc (c'est-à-dire le nombre de cases valant 1) :

$$H_p(U_\kappa)=\sum_{\ell=\sum_{x\in U_\kappa}x}^{|Z_\kappa||W_\kappa|}\binom{|Z_\kappa||W_\kappa|}{\ell}p^\ell(1-p)^{1-\ell}. \tag{1}$$

[6] La page dédiée est http://acgt.cs.tau.ac.il/samba/

 ISSN: 2102-6238

Pour sélectionner des blocs ayant un grand nombre de 1, un petit calcul montre que p doit être plus petit que $1/2$. Sous cette condition, la fonction de coût ne favorise pas les blocs monocases (comme celle de la section 5.1) car elle choisit un bloc de 4 cases contenant 3 cases avec des 1 (où le critère vaudrait $1-p$) plutôt qu'un bloc d'une seule case contenant 1 par exemple (où le critère vaudrait p). En revanche, la structure globale de la matrice n'influence pas le choix du bloc car pour tout p plus petit que $1/2$, les fonctions H_p ont le même comportement. De plus, cet algorithme calculant la valeur pour chaque bloc, sa complexité est de l'ordre de $\mathcal{O}\left(2^{nd}\right)$.
Pour contourner ce problème, Tanay et al. (2002) supposent que les cases x_{ij} sont le résultat de tirages aléatoires de lois de Bernoulli de paramètre p_{ij}, ces paramètres représentant la fraction de sous-graphes bipartis contenant l'arête (i,j) et ayant le même degré que le graphe associé à la matrice totale. L'estimation de ces paramètres est faite par simulation de Monte-Carlo et sera d'autant plus proche de 1 que les sommets de l'arrête seront reliés avec d'autres sommets. En adaptant le critère (1) et une fois les p_{ij} obtenus, l'algorithme *SAMBA* recherche le sous-graphe désiré avec une complexité de $\mathcal{O}\left(d2^{\max_j x_{1:n,j}}\right)$. L'algorithme sera d'autant plus rapide que le nombre de colonnes sera faible et qu'il n'y aura pas trop de 1 dans celles ci.
L'algorithme a été testé sur les données de levure mettant en évidence le lien entre des gènes bien connus et d'autres qui n'étaient pas encore classifiés.

5.4. *Algorithme OPSM*

Dans le cas de données où une relation d'ordre $\prec$ est possible (comme des données continues), Ben-Dor et al. (2003) proposent un algorithme recherchant un bloc avec une progression ordonnée sur les colonnes. Dans ce modèle, le bloc $U_\kappa = (Z_\kappa, W_\kappa)$ est supposé être généré en deux temps. Soit s un entier positif et $W_\kappa = \{j_1, \ldots, j_s\}$ un groupe ordonné parmi tous les groupes possibles de tailles s de $\mathcal{P}(\{1,\ldots,d\})$ l'ensemble des parties de $\{1,\ldots,d\}$. Chaque ligne i a la même probabilité π d'appartenir à Z_κ et, dans ce cas, les cases de la ligne i correspondantes aux colonnes de W_κ vérifient $x_{ij_1} \prec x_{ij_2} \prec \ldots \prec x_{ij_s}$. Connaître W_κ permet l'obtention rapide d'un ensemble contenant Z_κ, il suffit de conserver les lignes respectant l'ordre demandé. En pratique, la variable W_κ est inconnue et il est impossible de faire dans un temps raisonnable le même raisonnement pour chaque groupe de taille s.
L'algorithme *OPSM* (pour *order-preserving submatrices*) cherche les indices de W_κ en commençant par les plus extrêmes, c'est-à-dire les indices $W_\kappa^{(1)} = \{j_1, j_s\}$ puis en alternant les petits et grands indices ($W_\kappa^{(2)} = \{j_1, j_2, j_s\}$, $W_\kappa^{(3)} = \{j_1, j_2, j_{s-1}, j_s\}$, $W_\kappa^{(4)} = \{j_1, j_2, j_3, j_{s-1}, j_s\}$,...).
L'idée est que les "grands sauts" entre les valeurs sur les colonnes sont d'autant plus visibles que les indices sont séparés dans W_κ et la suite des groupes $Z_\kappa^{(c)}$ associés à chaque $W_\kappa^{(c)}$ est décroissante. L'inconvénient est que cet algorithme est sensible à l'estimation du premier groupe $W_\kappa^{(1)}$.
De plus, les auteurs soulignent que l'obligation d'ordre est trop rigide (ceci le rend très sensible au bruit, voir Prelić et al., 2006) et suggèrent un allègement de la condition en incluant une probabilité que certaines valeurs peuvent rompre la relation.
Cet algorithme a été testé sur des données du cancer du sein (Hedenfalk et al., 2001). L'algorithme a extrait de la matrice de taille 3226×22 une sous-matrice de taille 347×4.

 ISSN: 2102-6238

6. Discussion

6.1. Unité d'intérêt

En classification simple, les données sont généralement sous la forme d'un tableau où chaque ligne représente un individu et où chaque colonne représente une variable mesurée pour chaque individu. L'objectif de la classification est de regrouper les individus qui sont alors les unités d'intérêt.

En classification croisée, le but est de partitionner un tableau de données en blocs homogènes de lignes et de colonnes. Se pose alors la question de l'unité d'intérêt et ce choix reste encore à ce jour une question ouverte : est-ce le nombre de vecteurs en ligne, en colonne, le nombre d'éléments du tableau ou le tableau ?

Cette question a une importance notamment lors de l'étude asymptotique des propriétés des modèles probabilistes employés, lors de la construction des critères et lors de l'étude de la consistance des critères de sélection (Keribin et al., 2012; Lomet et al., 2012b). En effet, par défaut du fait de l'interdépendance entre les lignes et les colonnes, l'unité d'intérêt est le tableau dans sa globalité. Cependant, lors de l'écriture des critères asymptotiques, l'unité est le nombre d'éléments dans le tableau.

6.2. Évaluation des méthodes

L'évaluation objective des techniques d'apprentissage non supervisé sur des données réelles est difficile. Plusieurs points de vue sont légitimes et mesurer la pertinence d'une classification par sa capacité à retrouver un étiquetage particulier n'est donc pas nécessairement approprié. Ainsi, l'utilisation de données simulées est une technique traditionnelle pour l'évaluation expérimentale des algorithmes de classification car elle apporte une connaissance a priori sur les données. Un ensemble de jeux de données contrôlés permet de tester les limites d'un algorithme quant à sa capacité à retrouver une structure donnée dans des cas plus ou moins favorables.

L'absence de jeux de données de référence et de consensus sur l'évaluation des algorithmes amène chaque auteur (par exemple Banerjee et al., 2007; Good, 1965; Mariadassou et al., 2010) à proposer son propre protocole dont la reproductibilité est souvent limitée, du fait de la brièveté de la description et l'absence de mise à disposition des données. De plus, la qualité des résultats est difficilement évaluable car la difficulté intrinsèque du problème de classification croisée est inconnue du lecteur. En effet, si en classification simple, une analyse en composantes principales permet de visualiser le degré de mélange des classes, en classification croisée, cette approche est plus difficile du fait de la dualité des lignes et des colonnes du tableau de données. Lomet et al. (2012c) proposent un protocole de simulation dont la mesure de la difficulté est exprimée par le risque conditionnel de Bayes : le risque de Bayes est conditionné au tableau observé. Ces auteurs ont mis à disposition des jeux de données simulées binaires, continues et de contingence de différentes difficultés et tailles.

6.3. Choix du nombre de blocs

L'ensemble des méthodes de classification croisée présentées dans les sections précédentes pose le problème majeur du choix de modèle, et en particulier de la sélection du nombre de blocs.

 ISSN: 2102-6238

L'objectif de cette sélection est d'obtenir une classification pertinente, à un niveau de granularité approprié.

Ce problème est bien connu et abondamment traité en classification simple (Fraley et Raftery, 1998; Burnham et Anderson, 2004; Biernacki et al., 2000). Les solutions existantes en classification simple, basées sur des indices, des heuristiques ou des critères d'information, ont été ou peuvent être adaptées à la classification croisée.

6.3.1. Indices de sélection adaptés de la classification simple

Charrad et al. (2010) étendent à la classification croisée sept indices utilisés en classification simple : l'indice de Dunn (Dunn, 1974), l'indice de Baker et Hubert (Baker et Hubert, 1975), l'indice de Davies et Bouldin (Davies et Bouldin, 1979), l'indice de Calinsky et Harabsaz (Caliński et Harabasz, 1974), l'indice Silhouette de Rousseeuw (Rousseeuw, 1987), l'indice de Hubert et Levin (Hubert et Levin, 1976) et l'indice de Krzanowski et Lai (Krzanowski et Lai, 1988). Leur proposition consiste à appliquer ces différents indices de classification simple indépendamment sur les lignes et sur les colonnes du tableau de donnés, puis de calculer un indice global par une combinaison linéaire de ces deux derniers. Charrad et al. (2010) proposent également un huitième indice appelé la méthode de la différentielle ou résolution graphique s'appuyant sur l'algorithme `Croki2` (Govaert, 1983). L'objectif de cet indice est de maximiser le critère du χ^2 présenté en section 3.2.1. Ce dernier croit lorsque le nombre de classes en ligne ou en colonne augmente puis stagne ou croit plus lentement. Le nombre de classes sélectionné par l'indice proposé par Charrad et al. (2010) correspond alors à celui où la valeur du χ^2 ne croit plus ou peu. Ces huit indices ont été testés sur six tables de contingence simulées de taille 200×100 et dont le nombre de classes diffère. Dans ces expériences, tous les indices sélectionnent majoritairement les bons nombres de classes, avec un léger avantage pour la méthode différentielle et l'indice de Baker et Hubert. Comme toutes les approches basées sur l'évaluation des deux classifications du tableau, en lignes et en colonnes, ces indices ne tiennent pas compte de la structure croisée du problème. Le choix de la pondération de la combinaison linéaire est alors critique, et Charrad et al. (2010) n'apportent pas de solution à ce nouveau problème.

Rocci et Vichi (2008) proposent une extension de l'indice de Calinski-Harabasz (CH) qui est le rapport entre la dispersion inter-blocs et la dispersion intra-blocs. Cet indice est le plus performant d'une étude comparative de 30 indices en classification simple (Milligan et Cooper, 1985). Dans les expériences sur données simulées de Rocci et Vichi (2008), cet indice obtient de bons résultats pour des classes très séparées.

Schepers et al. (2008) proposent une extension de l'indice de Silhouette qui est appliqué pour des classifications simples basées sur la distance euclidienne comme les k-moyennes (k-*means*). Cette extension est une moyenne pondérée par le nombre de lignes et de colonnes des indices de Silhouette calculés sur l'ensemble des lignes et des colonnes. Ces auteurs font la remarque que ces deux derniers indices sont de simples extensions d'indices utilisés pour les k-moyennes et par conséquent ne prennent pas pleinement en compte la nature spécifique des données et des modèles en classification croisée.

 ISSN: 2102-6238

6.3.2. Sélection de modèle en classification croisée utilisant un modèle probabiliste

L'utilisation de modèles statistiques permet de considérer le problème du choix du nombre de classes comme un problème de sélection de modèles. Des critères de sélection généralement employés dans le domaine de la statistique peuvent donc être appliqués et dérivés au cas de la classification croisée.

Utilisant le modèle de blocs latents présentés précédemment en section 3.3.1, Van Dijk et al. (2009) emploient deux critères empiriques de sélection de modèles pour choisir le nombre de classes : le facteurs de Bayes ou le critère AIC3. Le facteur de Bayes sélectionne le modèle dont la vraisemblance marginale est maximale. Ce facteur se base sur l'estimation des distributions marginales pour plusieurs nombres de classes en ligne et en colonne qui sont très coûteuses en calculs. Van Dijk et al. (2009) appliquent aussi le critère AIC3 en se basant sur les résultats des simulations de Andrews et Currim (2003). La pénalité de ce critère tient compte du nombre de paramètres du modèle, du nombre de classes et du nombre d'étiquettes des appartenances aux classes estimées (en reprenant les notations de l'article $n(g-1)+d(m-1)+Kgm+(g-1)+(m-1)$ où K est la dimension des paramètres de chaque bloc). Les auteurs n'étudient pas les performances des critères sur des données simulées, mais ils argumentent la pertinence des classes obtenues sur des données réelles.

Wyse et Friel (2010) ont développé [7] une approche bayésienne du modèle de blocs latents dont le choix du nombre de classes est un paramètre intégré dans le modèle. Ils emploient une approche bayésienne utilisant un algorithme MCMC (*Markov Chain Monte Carlo*) incluant une distribution a priori sur le nombre de classes.

Récemment, Keribin et al. (2012) et Lomet et al. (2012b,a) ont montré que le critère exact ICL (*Integrated Completed Likelihood*), initialement développé pour la classification simple par modèle de mélange (Biernacki et al., 2010), peut être calculé exactement dans le cas du modèle de blocs latents binaires et continues afin de sélectionner le nombre de classes. Le critère ICL évalue la logvraisemblance complète intégrée du modèle de blocs latents. Supposant l'indépendance a priori des paramètres du modèle de blocs latents binaires et des distributions a priori sur ces derniers, ces auteurs développent un critère de sélection exact. Ils dérivent une version asymptotique de ce critère et en déduisent un critère BIC en supposant la consistance des paramètres estimés et la convergence des distributions conditionnelles des étiquettes vers les vraies partitions. Ces critères [8] ont été testés sur des jeux de données simulées et réelles et apportent une solution satisfaisante pour la sélection de modèle. Pour les petites tailles de tableaux, le critère exact ICL est le plus performant pour retrouver le nombre de classes utilisé pour la simulation des données alors que, pour les tableaux de grandes dimensions, le critère BIC donne de meilleurs résultats.

7. Conclusion

En statistique, les données sont souvent présentées sous la forme d'un tableau représentant par exemple, des objets en ligne et leurs caractéristiques en colonne. Pour traiter ces données, il existe

[7] Les programmes sont disponibles sur la page http://www.ucd.ie/statdept/jwyse.

[8] Ces critères sont proposés dans le package *blockcluster* disponible à l'adresse http://cran.r-project.org/web/packages/blockcluster/index.html du logiciel R.

 ISSN: 2102-6238

plusieurs techniques exploratoires dont l'objectif est de les expliquer et de les résumer. Parmi ces méthodes, la classification croisée permet de mettre en évidence leur structure en blocs : des classes en ligne et en colonne.

La structure latente du tableau, mise en évidence par une classification croisée, dépend de l'objectif de l'exploration des données et de la nature de ces dernières. En effet, le tableau peut être soit classé en créant les classes les lignes et les colonnes (la classification par blocs), soit présenté un ou plusieurs blocs atypiques (le chevauchement), soit présenté des blocs imbriqués (la classification imbriquée). La classification croisée par blocs est généralement utilisée pour l'analyse de données textuelles ou issue de la sociologie, du marketing, de l'imagerie et dans certains cas de la génomie alors que le chevauchement et la classification imbriquée sont généralement employées en génomie pour la détermination d'un groupe de gènes caractéristiques, par exemple.

Quelque soit le type de structure latente des données, plusieurs méthodes et algorithmes sont possibles. Comme il n'existe toujours pas de consensus général pour en évaluer la qualité et ce, malgré ces cinquante ans d'histoire, il est préférable d'en utiliser plusieurs et de comparer les résultats obtenus. Une comparaison entre différentes classifications peut alors être réalisée par la pertinence des classes obtenues et/ou par des critères et indices.

 ISSN: 2102-6238

Annexe A: Tableau récapitulatif

Pour une vue plus globale, nous proposons un tableau récapitulatif des différents algorithmes existant :

Nom	Méthode	Données	Type de critères pour la classification	Classification	Exemples d'applications proposées dans les articles	Référence
Corki2	Par blocs	Contingence	Déterministe	Simultanée	Géographie, données textuelles	Govaert (1983)
Croeuc	Par blocs	Continues	Déterministe	Simultanée	Cadre général, données de recommandation, écologie	Govaert (1983)
Crobin	Par blocs	Binaires	Déterministe	Simultanée	Cadre général, sociologie	Govaert (1983)
NMF	Par blocs	Tout type	Déterministe	Simultanée	Cadre général	Seung et Lee (2001)
NBVD	Par blocs	Tout type	Déterministe	Simultanée	Données textuelles	Long et al. (2005)
ONMF	Par blocs	Tout type	Déterministe	Simultanée	Données textuelles	Yoo et Choi (2010)
VEM, CEM	Par blocs	Tout type	Probabiliste	Simultanée	Cadre général, génétique, sociologie, données textuelles, données de recommandation	Govaert et Nadif (2009, 2008, 2007)
SEM	Par blocs	Tout type	Probabiliste	Simultanée	Cadre général, génétique, sociologie, données textuelles, données de recommandation	Keribin et al. (2010)
V-Bayes	Par blocs	Tout type	Probabiliste	Simultanée	Cadre général, génétique, sociologie	Keribin et al. (2012, 2013)
Algorithme MCMC	Par blocs	Tout type	Probabiliste	Simultanée	Cadre général, génétique, sociologie	Wyse et Friel (2010)
Brute-Force Deletion and Addition	Un seul bloc	Tout type	Déterministe	Simultanée	Génétique	Cheng et Church (2000)
Finding a Given Number of Biclusters	Chevauchement	Tout type	Déterministe	Simultanée	Génétique	Cheng et Church (2000)
FLOC	Chevauchement	Tout type	Déterministe	Simultanée	Génétique	Yang et al. (2002)
ISA (Iterative Signature Algorithm)	Chevauchement	Binaire	Déterministe	Alternée	Génétique	Ihmels et al. (2004)
Lazzeroni and Owen	Chevauchement	Tout type	Probabiliste	Simultanée	Génétique, biologie, économie	Lazzeroni et Owen (2000)
Multiple Node Deletion	Un seul bloc	Tout type	Déterministe	Simultanée	Génétique	Cheng et Church (2000)
Node Addition	Un seul bloc	Tout type	Déterministe	Simultanée	Génétique	Cheng et Church (2000)
One-Way splitting	Imbriquée	Tout type	Déterministe	Alternée	Sociologie	Hartigan (1975)
OPSM	un seul bloc	Ordonnées	Probabiliste	Simultanée	Génétique	Ben-Dor et al. (2003)
Permutation-Based Block Clustering	Par blocs	Tout type	Déterministe	Alternée	Sociologie	Duffy et Quiroz (1991)
Ping-Pong	Un seul bloc	Binaires	Déterministe	Simultanée	Marketing	Oyanagi et al. (2001)
Processus Mondrain	Imbriquée	Multinomiale	Probabiliste	Simultanée	Génétique, Sociologie	Roy et Teh (2008)
Samba	Chevauchement	Binaires	Probabiliste	Simultanée	Génétique	Tanay et al. (2002)
Single Node Deletion	Un seul bloc	Tout type	Déterministe	Simultanée	Génétique	Cheng et Church (2000)
Two-Way splitting	Imbriquée	Tout type	Déterministe	Alternée	Sociologie	Hartigan (1975)
xMotif	Chevauchement	Binaire	Probabiliste	Simultanée	Génétique	Murali et Kasif (2003)

La colonne "classification" signifie si cette dernière a été fait de façon simultanée ("simultanée") ou si les algorithmes alternent entre une classification sur les lignes et les colonnes ("Alternée")

 ISSN: 2102-6238

Références

Andrews, R. L. et Currim, I. S. (2003). A comparison of segment retention criteria for finite mixture logit models. *Journal of Marketing Research*, pages 235–243.

Baker, F. B. et Hubert, L. J. (1975). Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 70(349) :31–38.

Banerjee, A., Dhillon, I., Ghosh, J., et Merugu, S. (2007). A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8 :1919–1986.

Banfield, J. D. et Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, pages 803–821.

Ben-Dor, A., Chor, B., Karp, R., et Yakhini, Z. (2003). Discovering local structure in gene expression data : the order-preserving submatrix problem. *Journal of computational biology*, 10(3-4) :373–384.

Bennett, J. et Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35.

Berkhin, P. (2006). *A survey of clustering data mining techniques*. Springer.

Biernacki, C., Celeux, G., et Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7) :719–725.

Biernacki, C., Celeux, G., et Govaert, G. (2010). Exact and Monte Carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, 140(11) :2991–3002.

Bock, H. (1979). Simultaneous clustering of objects and variables. *Analyse des données et Informatique*, pages 187–203.

Burnham, K. P. et Anderson, D. R. (2004). Multimodel inference understanding aic and bic in model selection. *Sociological methods & research*, 33(2) :261–304.

Caliński, T. et Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1) :1–27.

Celeux, G. et Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3) :315–332.

Celeux, G. et Robert, C. (1993). Une histoire de discrétisation. *La Revue de Modulad*, 11 :7–44.

Charrad, M., Lechevallier, Y., Saporta, G., et Ben Ahmed, M. (2010). Détermination du nombre de classes dans les méthodes de bipartitionnement. In *17ème Rencontres de la Société Francophone de Classification*, pages 119–122, Saint-Denis de la Réunion.

Cheng, Y. et Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)*, page 93.

Davies, D. L. et Bouldin, D. W. (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2) :224–227.

Dempster, A. P., Laird, N. M., Rubin, D. B., et al. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38.

Deodhar, M. et Ghosh, J. (2007). Simultaneous co-clustering and modeling of market data. In *Proceedings of the Workshop for Data Mining in Marketing (DMM 2007)(Leipzig, Germany). IEEE Computer Society Press, Los Alamitos, CA*.

Dhillon, I. S., Mallela, S., et Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM.

Duffy, D. E. et Quiroz, J. (1991). A permutation-based algorithm for block clustering. *Journal of Classification*, 8(1) :65–91.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1) :95–104.

Fraley, C. et Raftery, A. E. (1998). How many clusters ? Which clustering method ? Answers via model-based cluster analysis. *The Computer Journal*, 41(8) :578–588.

Gan, G., Ma, C., et Wu, J. (2007). *Data clustering : theory, algorithms, and applications*, volume 20. Siam.

Good, I. J. (1965). *Categorization of classification*. Mathematics and Computer Science in Biology and Medicine. Her Majesty's Stationery Office.

Govaert, G. (1977). Algorithme de classification d'un tableau de contingence. In *First international symposium on data analysis and informatics*, pages 487–500, Versailles. INRIA.

Govaert, G. (1983). *Classification croisée*. PhD thesis, Thèse d'état, Université Pierre et Marie Curie.

Govaert, G. (1989). Classification croisée. *Modulad*, 4 :9–36.

Govaert, G. (1995). Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24(4) :437–458.

 ISSN: 2102-6238

Govaert, G. et Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36 :463–473.
Govaert, G. et Nadif, M. (2007). Clustering of contingency table and mixture model. *European Journal of Operational Research*, 183 :1055–1066.
Govaert, G. et Nadif, M. (2008). Block clustering with Bernoulli mixture models : Comparison of different approaches. *Computational Statistics and Data Analysis*, 52 :3233–3245.
Govaert, G. et Nadif, M. (2009). Un modèle de mélange pour la classification croisée d'un tableau de données continues. In *CAP'09, 11e conférence sur l'apprentissage artificiel*.
Hanczar, B. et Nadif, M. (2010). Bagging for biclustering : Application to microarray data. In Balcázar, J., Bonchi, F., Gionis, A., et Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6321 of *Lecture Notes in Computer Science*, pages 490–505. Springer Berlin Heidelberg.
Hansohm, J. (2002). Two-mode clustering with genetic algorithms. In *Classification, automation, and new media*, pages 87–93. Springer.
Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition.
Hartigan, J. A. (2000). Bloc voting in the United States senate. *Journal of Classification*, 17(1) :29–49.
Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bitter, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., et Raffeld, M. (2001). Gene-expression profiles in hereditary breast cancer. *New Eng. J. Med.*, 344 :539–548.
Hubert, L. J. et Levin, J. R. (1976). A general statistical framework for assessing categorical clustering in free recall. *Psychological bulletin*, 83(6) :1072–1080.
Ihmels, J., Bergmann, S., et Barkai, N. (2004). Defining transcription modules using large-scale gene expression data. *Bioinformatics*, 20(13) :1993–2003.
Jagalur, M., Pal, C., Learned-Miller, E., Zoeller, R. T., et Kulp, D. (2007). Analyzing in situ gene expression in the mouse brain with image registration, feature extraction and block clustering. *BMC Bioinformatics*, 8(Suppl 10) :S5.
Jain, A. K., Murty, M. N., et Flynn, P. J. (1999). Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323.
Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., et Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence*, pages 381–388. AAAI Press.
Keribin, C., Brault, V., Celeux, G., et Govaert, G. (2012). Model selection for the binary latent block model. In *Compstat*, pages 379–390.
Keribin, C., Brault, V., Celeux, G., et Govaert, G. (2013). Estimation and Selection for the Latent Block Model on Categorical Data. Rapport de recherche RR-8264, INRIA.
Keribin, C., Govaert, G., et Celeux, G. (2010). Estimation d'un modèle à blocs latents par l'algorithme SEM. In *42e Journées de Statistique, SFdS*, Marseille.
Kluger, Y., Basri, R., Chang, J. T., et Gerstein, M. (2003). Spectral biclustering of microarray data : coclustering genes and conditions. *Genome Research*, 13(4) :703–716.
Krzanowski, W. J. et Lai, Y. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, pages 23–34.
Lashkari, D. et Golland, P. (2009). Co-clustering with generative models. Technical report, MITCSAIL-TR-2009-054.
Lazzeroni, L. et Owen, A. (2000). Plaid models for gene expression data. *Statistica Sinica*, 12 :61–86.
Leredde, H. et Perin, P. (1980). Les plaques-boucles mérovingiennes. *Dossiers de l'Archéologie*, 42 :83–87.
Lerman, I. et Leredde, H. (1977). La méthode des pôles d'attraction. *Journées Analyse des Données et Informatique*.
Lomet, A., Govaert, G., Grandvalet, Y., et al. (2012a). An approximation of the integrated classification likelihood for the latent block model. *ICDM 2012 IEEE International Conference on Data Mining*.
Lomet, A., Govaert, G., Grandvalet, Y., et al. (2012b). Model selection in block clustering by the integrated classification likelihood. *Proceedings of Compstat 2012*, pages 519–530.
Lomet, A., Govaert, G., Grandvalet, Y., et al. (2012c). Un protocole de simulation de données pour la classification croisée. *44e Journées de Statistique de la SFdS*.
Long, B., Zhang, Z. M., et Yu, P. S. (2005). Co-clustering by block value decomposition. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 635–640. ACM.
Madeira, S. C. et Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis : a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(1) :24–45.
Mariadassou, M., Robin, S., et Vacher, C. (2010). Uncovering latent structure in valued graphs : a variational approach. *The Annals of Applied Statistics*, 4(2) :715–742.
Matias, C. et Robin, S. (2014). Modeling heterogeneity in random graphs : a selective review. *arXiv :1402.4296*.

 ISSN: 2102-6238

Maugis, C., Martin-Magniette, M.-L., Tamby, J.-P., Renou, J.-P., Lecharny, A., Aubourg, S., et Celeux, G. (2009). Sélection de variables pour la classification par mélanges gaussiens pour prédire la fonction des gènes orphelins. *La Revue de Modulad*, 40 :69–80.

McLachlan, G. J. (1982). The classification and mixture maximum likelihood approaches to cluster analysis. *Handbook of statistics*, 2 :199–208.

Meeds, E. et Roweis, S. (2007). Nonparametric Bayesian biclustering. Technical report, University of Toronto.

Milligan, G. W. et Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2) :159–179.

Murali, T. et Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data. In *Pacific Symposium on Biocomputing*, volume 8, pages 77–88.

Nowicki, K. et Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455) :1077–1087.

Oyanagi, S., Kubota, K., et Nakase, A. (2001). Application of matrix clustering to web log analysis and access prediction. In *WEBKDD 2001-Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, pages 13–21.

Podani, J. et Feoli, E. (1991). A general strategy for the simultaneous classification of variables and objects in ecological data tables. *Journal of Vegetation Science*, 2(4) :435–444.

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., et Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9) :1122.

Robert, C. (2006). *Le choix bayésien : Principes et pratique*. Springer Science & Business.

Rocci, R. et Vichi, M. (2008). Two-mode multi-partitioning. *Computational Statistics and Data Analysis*, 52(4) :1984–2003.

Rooth, M. (1995). Two-dimensional clusters in grammatical relations. In *AAAI Symposium on Representation and Acquisition of Lexical Knowledge*.

Rousseeuw, P. J. (1987). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20 :53–65.

Roy, D. M. et Teh, Y. W. (2008). The mondrian process. In *NIPS*, pages 1377–1384.

Schepers, J., Ceulemans, E., et Van Mechelen, I. (2008). Selecting among multi-mode partitioning models of different complexities : A comparison of four model selection criteria. *Journal of Classification*, 25(1) :67–85.

Seldin, Y. et Tishby, N. (2010). Pac-Bayesian analysis of co-clustering and beyond. *The Journal of Machine Learning Research*, 11 :3595–3646.

Seung, D. et Lee, L. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems 13*, pages 556–562.

Shafiei, M. et Milios, E. (2006). Model-based overlapping co-clustering. In *Proceeding of SIAM Conference on Data Mining*.

Shan, H. et Banerjee, A. (2008). Bayesian co-clustering. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, pages 530–539.

Tanay, A., Sharan, R., et Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. In *Proceedings of ISMB 2002*, pages 136–144.

Tishby, N., Pereira, F., et Bialek, W. (1999). The information bottleneck method. In *Invited paper to The 37th annual Allerton Conference on Communication, Control, and Computing*.

Van Dijk, B., Van Rosmalen, J., et Paap, R. (2009). A Bayesian approach to two-mode clustering. Technical Report 2009-06, Econometric Institute.

Wang, P., Laskey, K. B., Domeniconi, C., et Jordan, M. (2011). Nonparametric bayesian co-clustering ensembles. SIAM.

Wyse, J. et Friel, N. (2010). Block clustering with collapsed latent block models. *Statistics and Computing*, pages 1–14.

Yang, J., Wang, W., Wang, H., et Yu, P. (2002). δ-clusters : Capturing subspace correlation in a large data set. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 517–528. IEEE.

Yoo, J. et Choi, S. (2010). Orthogonal nonnegative matrix tri-factorization for co-clustering : Multiplicative updates on stiefel manifolds. *Information processing & management*, 46(5) :559–570.

 ISSN: 2102-6238