Le bi-partitionnement : Etat de l'art sur les approches et les algorithmes

Malika Charrad, Gilbert Saporta, Yves Lechevallier, Mohamed Ben Ahmed

Le bi-partitionnement : Etat de l'art sur les approches et les algorithmes

Malika Charrad*,* — Gilbert Saporta**— Yves Lechevallier***— Mohamed Ben Ahmed***

** Laboratoire RIADI, Ecole Nationale des Sciences de l'Informatique*
Université de la Manouba, Tunis

{malika.charrad, mohamed.benahmed}@riadi.rnu.tn

*** Laboratoire CEDRIC, Conservatoire National des Arts et Métiers*
292 rue Saint-Martin, 75141 Paris cedex 03

saporta@cnam.fr

**** INRIA-Rocquencourt, 78153 Lechesney cedex*

yves.lechevallier@inria.fr

RÉSUMÉ. *Les méthodes de bi-partitionnement cherchent simultanément des partitions sur l'ensemble des lignes et l'ensemble des colonnes. Ils trouvent aujourd'hui leur application dans plusieurs domaines tels que la bioinformatique, le text mining et le Web mining. Dans ce papier, nous proposons une étude sur les approches et les algorithmes proposés pour le bi-partitionnement ainsi qu'une classification de ces algorithmes suivant le type des biclasses et les approches utilisées.*

ABSTRACT. *Simultaneous clustering methods perform clustering in the two dimensions simultaneously. They seek to find sub-matrices, that is subgroups of rows and subgroups of columns. They have practical importance in a wide of variety of applications such as biology data analysis, text mining and web mining. In this paper, we introduce a large number of existing approaches of simultaneous clustering and classify them in accordance with the type of biclusters they can find, the methods used and the target applications.*

MOTS-CLÉS : *Algorithmes de classification simultanée, biclasses, approches de bi-partitionnement, classification directe, classification croisée.*

KEYWORDS: *Simultaneous clustering algorithms, biclusters, biclustering approaches, direct clustering, crossed clustering.*

1. Introduction

Les méthodes de classification automatique appliquées à des tableaux mettant en jeu deux ensembles de données agissent de façon dissymétrique et privilégient un des deux ensembles en ne faisant porter la structure recherchée que sur un seul ensemble. L'application d'une classification sur chaque ensemble est possible mais la détermination des liens entre les deux partitions est difficile. La recherche de structures de classes symétriques, plus précisément, la recherche simultanée de partitions sur les deux ensembles a donné naissance à des méthodes de classification simultanée ou bi-partitionnement. Ce type d'approche a suscité 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. Cependant, les travaux de synthèse sur les algorithmes de bi-partitionnement sont concentrés sur les algorithmes appliqués en bioinformatique tels que les travaux de Madeira et Oliveira (Madeira et Oliveira, 2004) et Tanay et al. (Tanay et al, 2004). Par ailleurs, les algorithmes de classification directe proposés par Hartigan (Hartigan, 1975) et les travaux de Govaert (Govaert, 1983) sur la classification croisée trouvent aujourd'hui leur application en web usage mining, information retrieval et text mining. D'autres algorithmes sont également proposés pour la classification simultanée tels que les algorithmes basés sur le modèle de mélange (Govaert et Nadif, 2005) et ceux basés sur la théorie de l'information (Dhillon et al., 2003) et (Robardet et al., 2002). Notre objectif dans ce papier est de présenter un état de l'art sur les différents algorithmes de classification simultanée et les classifier selon certains critères.

2. Principe général de la classification simultanée

Soit A une matrice des données à n lignes et m colonnes, définie par l'ensemble $X = \{X_1,...,X_n\}$ des lignes et l'ensemble $Y = \{Y_1,...,Y_m\}$ des colonnes. a_{ij}, avec $1 \leq i \leq n$ et $1 \leq j \leq m$, sont les éléments de la matrice A.

Tableau 1. ***Matrice des données***

	$\mathbf{Y_1}$	…	$\mathbf{Y_j}$	…	$\mathbf{Y_m}$
$\mathbf{X_1}$	a_{11}	…	a_{1j}	…	a_{1m}
…	…	…	…	…	…
$\mathbf{X_i}$	a_{i1}		a_{ij}		a_{im}
…	…	…	…	…	…
$\mathbf{X_n}$	a_{n1}	…	a_{nj}	…	a_{nm}

Les algorithmes de bi-partitionnement ou de classification simultanée ont pour objectif d'identifier un ensemble de biclasses $B_k = (I_k, J_k)$, I_k est une classe définie sur X et J_k est une classe définie sur Y, tel que chaque biclasse B_k satisfait certains critères d'homogénéité. Ces critères varient d'un algorithme à un autre.

3. Types des biclasses

Les algorithmes de bi-partitionnement permettent de découvrir quatre types de biclasses (Madeira et Oliveira, 2004).

– Biclasses à valeurs constantes

Une biclasse à valeurs constantes est une sous-matrice (I,J) tel que $\forall i \in I,\ \forall j \in J,\ a_{ij} = \mu$ Cependant, dans le cas réel, compte tenu du bruit dans les données, la valeur a_{ij} se présente généralement sous la forme $\eta_{ij} + \mu$, où η_{ij} est le bruit associé à la valeur réelle de a_{ij}.

– Biclasses à valeurs constantes sur les lignes ou les colonnes

Une biclasse à valeurs constantes sur les lignes est une sous-matrice (I,J) où toutes les valeurs a_{ij} de la biclasse sont obtenues en utilisant le modèle additif ou le modèle multiplicatif suivant.

Modèle additif : $a_{ij} = \mu + \alpha_i$

Modèle multiplicatif : $a_{ij} = \mu \times \alpha_i$

μ est une valeur caractéristique de la biclasse et α_i est l'ajustement sur la ligne $i \in I$. Une biclasse à valeurs constantes sur les colonnes est caractérisée par un ajustement β_j sur la colonne $j \in J$.

– Biclasses à valeurs cohérentes

Une biclasse à valeurs cohérentes est définie en utilisant le modèle additif ou le modèle multiplicatif.

Modèle additif : $a_{ij} = \mu + \alpha_i + \beta_j$

Modèle multiplicatif : $a_{ij} = \mu \times \alpha_i \times \beta_j$

où μ est une valeur caractéristique de la biclasse, α_i est l'ajustement sur la ligne $i \in I$ et β_j est l'ajustement sur la colonne $j \in J$.

– Biclasses à évolutions cohérentes

Les approches visant à identifier dans les données des biclasses à évolutions cohérentes considèrent que les éléments de la matrice sont des valeurs symboliques et essayent de découvrir des sous-ensembles de lignes et des sous-ensembles de colonnes dont l'évolution est cohérente par rapport à la valeur exacte de la matrice des données.

6. Approches de bi-partitionnement

(Madeira et Oliveira, 2004) proposent de classifier les algorithmes de bi-partitionnement suivant les approches utilisées pour leur construction. Ces approches sont classifiées en cinq catégories.

– IRCCC (Iterative Row and Column Clustering Combination)

Le principe de cette approche est d'appliquer un algorithme de partitionnement simple sur les lignes et les colonnes de la matrice séparément puis de combiner les résultats pour construire des biclasses.

– DC (Divide and Conquer)

L'approche DC présente l'avantage d'être rapide à identifier les biclasses dans les données. Cependant, elle a comme inconvénient de ne pas pouvoir identifier certaines bonnes biclasses avant leur découpage. Cette approche affecte initialement tous les éléments à la même biclasse et procède à un découpage itératif.

– GIS (Greedy Iterative Search)

L'approche GIS est basée sur l'idée de créer des biclasses en ajoutant ou supprimant des lignes ou des colonnes à ces biclasses afin d'optimiser un certain critère. Malgré sa rapidité, cette approche présente l'inconvénient de ne pas garder certaines bonnes biclasses en ajoutant ou supprimant des lignes ou des colonnes.

– EBE (Exhaustive Bicluster Enumeration)

L'approche EBE est basée sur l'idée que l'identification des meilleures biclasses nécessite une énumération exhaustive de toutes les biclasses possibles dans la matrice des données. L'avantage de cette approche est la certitude de trouver les meilleures biclasses mais son inconvénient majeur est sa complexité qu'il faut réduire en appliquant des restrictions sur la taille des biclasses.

– DPI (Distribution Parameter Identification)

L'approche DPI suppose la présence d'un modèle statistique dans les données et cherche à identifier la distribution des paramètres utilisés pour générer les données en minimisant certains critères à travers une approche itérative.

7. Classification des méthodes de bi-partitionnement

Nous proposons de classifier les méthodes de bi-partitionnement en quatre catégories :

– Méthodes basées sur des algorithmes de partitionnement simple

Ces méthodes consistent à utiliser un algorithme de classification simple tel que le K-means, le SOM ou la classification hiérarchique sur les lignes et les colonnes séparément. Les biclasses sont construites à partir des classes obtenues sur les lignes et les colonnes. Ces méthodes permettent d'extraire des classes de lignes et de colonnes mais pas directement des biclasses.

– Méthodes divisives

Ces méthodes procèdent par découpage itératif afin d'aboutir à des biclasses qui optimisent certains critères.

– Méthodes constructives

Les méthodes constructives consistent à construire des biclasses de différentes manières : par ajout et suppression des lignes et des colonnes (δ-biclusters), par permutation des lignes et des colonnes (OPSM), par estimation des paramètres des modèles (plaid models), à partir des vecteurs propres (algorithme spectral) ou à partir d'un graphe biparti (SAMBA).

– Méthodes probabilistes

Ce sont les méthodes basées sur le modèle de mélange et les méthodes basées sur la théorie de l'information.

5.1. *Méthodes basées sur des algorithmes de partitionnement simple*

5.1.1. *Algorithmes de classification croisée CROKI2, CROEUC et CROBIN*

Ces algorithmes sont proposés par Govaert (Govaert, 1983). L'algorithme CROKI2 (classification CROisée optimisant le Khi2 du tableau de contingence) appliqué aux tableaux de contingence a pour objectif de trouver une partition P de X en K classes et une partition Q de Y en L classes telle que le Khi2 de contingence du nouveau tableau construit en regroupant les lignes et les colonnes suivant les partitions P et Q soit maximum.

L'algorithme CROKI2 consiste à déterminer une série de couples de partitions (P^n,Q^n) optimisant le Khi2 du tableau de contingence en appliquant alternativement sur X et sur Y une variante de la méthode des nuées dynamiques.

Le principe de l'algorithme CROEUC (classifcation CROisée optimisant un critère basé sur la distance EUClidienne) proposé pour les tableaux de mesure est le même que celui de l'algorithme CROKI2. Il consiste à chercher alternativement la meilleure partition de X en fixant la partition de Y et inversement. Le critère à optimiser est l'information associée au tableau X(P,Q).

L'algorithme CROBIN (classification CROisée d'un tableau BINaire) appliqué aux tableaux binaires a pour objectif d'obtenir en réordonnant les lignes et les colonnes du tableau initial suivant les deux partitions P et Q, des blocs homogènes de 1 ou de 0. A chaque couple (k,l) de classes une valeur binaire idéale (1 ou 0) est associée. Le tableau binaire obtenu est appelé noyau. Le meilleur noyau N associé à un couple de partitions est formé des n_k^l définis comme étant l'élément 1 ou 0 majoritaire dans le couple (P_k,Q_l). L'objectif de l'algorithme consiste à minimiser l'écart entre le tableau initial structuré suivant les deux partitions P et Q et le tableau idéal N.

Le problème de ces algorithmes est qu'il faut fixer le nombre de classes en ligne et en colonnes.

5.1.2. *Algorithme CTWC « Coupled Two-Way Clustering »*

L'algorithme CTWC (Coupled two-way clustering) (Getz, 2000) consiste à appliquer un algorithme de classification hiérarchique, le SPC « SuperParamagnetic Clustering (SPC) » sur les colonnes en utilisant toutes les lignes et vice versa. Toutes les sous-matrices (I,J) tel que I est une classe en ligne et J une classe en colonne sont calculées. Seules les sous-matrices qui satisfont un certain critère comme la stabilité ou une taille minimale sont retenues. Ensuite le processus est réitéré : des classes de lignes et de colonnes sont extraites à partir de ces sous-matrices, etc.

5.1.3. *Algorithme ITWC « Interrelated Two-Way Clustering »*

L'algorithme ITWC (Tang, 2001) fait appel à un algorithme de partitionnement simple, tels que k-means ou SOM, appliqué séparément sur les deux dimensions de la matrice des données pour obtenir des biclasses. La première étape de l'algorithme consiste à appliquer l'algorithme de partitionnement simple sur les lignes, puis en se basant sur les classes obtenues à l'étape 1, le même algorithme est appliqué sur les colonnes. Le nombre de classes recherchées sur les colonnes est généralement k=2. L'étape suivante consiste à combiner les résultats de la première et la deuxième étape et trouver les biclasses hétérogènes. L'évaluation des biclasses est effectuée à l'aide de la « validation croisée ». La condition d'arrêt de l'algorithme est d'avoir une valeur du ratio « Occratio » qui atteint le seuil T, généralement égal à 0.9.

5.1.4. Algorithme DCC «Double Conjugated Clustering »

(Busygin et al., 2002) propose de partitionner l'ensemble des lignes et l'ensemble des colonnes à l'aide des cartes auto-organisatrices de Kohonen (SOM) et relier les deux partitions par l'intermédiaire d'une bijection associant à chaque nœud (i.e. vecteur représentant chaque classe) de l'un des deux espaces un nœud de l'autre espace appelé conjugué. Cette méthode présente l'avantage de convergence relativement rapide et aboutit à la construction de deux partitions, une dans l'espace des lignes et l'autre dans l'espace des colonnes. Chacune de ces partitions est le conjugué de l'autre.

5.2. *Méthodes divisives*

Les éléments à classifier ne sont plus ni les objets ni les variables mais les valeurs elles-mêmes du tableau de données. Ces méthodes sont nommées par Hartigan (Hartigan, 1975), méthodes de classification directe.

5.2.1. *Algorithme One-way splitting*

Le « one way splitting » est un algorithme divisif qui propose un découpage en blocs homogènes des objets. Il se concentre principalement sur la partition des objets, en essayant de construire des classes de telle manière que les variables aient une variance intra-classe inférieure à un certain seuil. L'idée de base de l'algorithme est de n'utiliser que les variables ayant une variance supérieure au seuil dans une classe donnée pour découper cette classe.

5.2.2. *Two-way splitting*

Lorsque les données sont directement comparables d'un attribut à un autre, Hartigan (Hartigan, 1975) propose un algorithme divisif, Two-way splitting, qui choisit à chaque étape entre une division de l'ensemble des instances et une division de l'ensemble des attributs. Ce choix est basé sur la réduction au maximum de l'hétérogénéité du groupe d'instances ou de variables à diviser. Afin de respecter les contraintes hiérarchiques imposées pour cet algorithme, les divisions effectuées à une étape ne sont jamais remises en cause aux étapes suivantes. L'avantage de cet algorithme est qu'il ne nécessite pas de savoir à l'avance le nombre de blocs à obtenir.

5.3. *Méthodes constructives*

5.3.1. *Algorithme δ-biclusters*

Le principe de l'algorithme δ-biclusters proposé par Cheng et Church (Cheng et Church, 2000) consiste à supprimer itérativement des lignes et des colonnes à partir de la matrice initiale jusqu'à ce que la mesure de distance soit inférieure à un certain seuil δ, puis ajouter des lignes et des colonnes itérativement sans entraîner une augmentation de cette mesure de distance. A chaque itération, une biclasse est générée puis remplacée dans la matrice initiale par des valeurs aléatoires.
Une limite de cette approche est que le nombre de biclasses à rechercher doit être fixé par l'utilisateur tout comme le seuil δ utilisé pour la mesure de la qualité. En plus, la qualité des biclasses diminue à chaque itération à cause des valeurs aléatoires ajoutées à chaque itération.
Pour pallier ce problème, Yang et al. (Yang, 2003) proposent l'algorithme FLOC (FLexible Overlapped biClustering) permettant de prendre en compte les valeurs manquantes qui ne sont plus considérées dans le calcul de la moyenne des lignes, la moyenne des colonnes et la moyenne de la biclasse utilisée pour le calcul de la distance. Cet algorithme est réalisé en deux phases. Dans la première phase, k biclasses initiales sont générées en affectant chaque ligne ou colonne à chacune des biclasses avec une probabilité p. Dans la seconde phase, un processus itératif permet d'améliorer la qualité des biclasses.

5.3.2. *Algorithme OPSM « Order-Preserving Sub-Matrix »*

Ben-Dor et al. (Ben-Dor, 2002) définissent une biclasse comme une sous-matrice préservatrice de l'ordre. Ils proposent l'algorithme OPSM dont l'objectif est de construire des larges biclasses. Une sous-matrice est préservatrice de l'ordre s'il existe une permutation des colonnes permettant d'avoir des valeurs strictement croissantes sur chaque ligne. (Ben-Dor, 2002) définit un modèle complet comme étant un couple (J,π) où J est un ensemble de colonnes et π une permutation. Une ligne de la matrice suit ce modèle si ses valeurs ordonnées suivant la permutation π sont strictement croissantes. Soit la matrice des données X×Y, le problème de détermination d'un OPSM de taille k×s est NP-complet. L'idée de l'algorithme est

de commencer par identifier des modèles partiels. Ces modèles sont agrandis pour aboutir à des modèles complets.

5.3.3. *Algorithme δ-patterns*

L'algorithme δ-patterns proposé par Califano et al. (Califano, 2000) a pour objectif d'identifier des biclasses maximales à valeurs constantes sur les lignes. Ils définissent une *ks*-biclasse δ-valide comme un sous-ensemble I de lignes, de taille k et un sous-ensemble J de colonnes, de taille s tels que la différence entre la valeur maximale et la valeur minimale de chaque ligne est inférieure à un certain seuil.
Une *ks*-biclasse δ-valide est dite maximale si elle ne peut pas être étendue en une *k's*-biclasse δ-valide en y ajoutant des lignes telle que k'> k ou en une *ks'*-biclasse δ-valide en y ajoutant des colonnes telle que s'> s.

5.3.4. *Algorithme SAMBA « Statistical Algorithmic Method for Bicluster Analysis »*

L'algorithme SAMBA proposé par Tanay et al. (Tanay, 2002) est basé sur la théorie des graphes et la modélisation probabiliste.
La matrice des données peut être représentée par un graphe G biparti pondéré où chaque nœud n_i correspond à une ligne et chaque nœud n_j correspond à une colonne. L'arête entre le nœud n_i et le nœud n_j a un poids a_{ij} correspondant à l'élément de la matrice se trouvant à l'intersection de la ligne i et la colonne j. Une biclasse correspond au sous-graphe $H = (I, J, E)$ de G et représente un sous-ensemble I d'objets (gènes) dont la valeur change significativement sous un ensemble de variables (conditions) J. L'objectif de l'algorithme SAMBA est de chercher dans les données des biclasses maximales. L'application de l'algorithme SAMBA est effectuée en deux étapes. Dans la première étape, les données sont normalisées et représentées par un graphe biparti. Dans la seconde étape, l'algorithme identifie les K bi-cliques maximales. Dans une phase ultérieure, SAMBA apporte des améliorations locales aux biclasses par ajout ou suppression des sommets, et sélectionne les biclasses similaires ayant un nombre important de sommets en commun.

5.3.5. *Algorithme de Lazzeroni et Owen*

L'algorithme proposé par Lazzeroni et Owen (Lazzeroni, 2000) pour la classification d'expression de gènes est connu sous le nom de « Plaid models ». L'idée de base est de représenter la matrice des données comme une superposition des biclasses. La matrice des données est représentée par une fonction linéaire de variables correspondant aux biclasses $f_{ij} = \sum_{k=0}^{K} \theta_{ijk} \rho_{ik} \psi_{jk}$

Avec K est le nombre de biclasses, ρ_{ik} vaut 1 si la ligne *i* appartient à la biclasse *k* et 0 sinon. Ψ_{jk} vaut 1 si la colonne *j* appartient à la biclasse *k* et 0 sinon. Les valeurs de θ_{ijk} représentent les modèles d'analyse de la variance (ANOVA) qui varient selon le type des biclasses recherchées.

Le problème de bi-partitionnement consiste à rechercher les valeurs des paramètres de manière à minimiser la distance euclidienne entre les valeurs observées et celles modélisées.

5.3.6. *Algorithme spectral*

Kluger et al. (Kluger, 2003) utilisent une approche spectrale pour le bi-partitionnement en supposant que la matrice des données comporte une structure d'échiquier après normalisation. L'algorithme est basé sur la recherche de vecteurs propres dans la matrice des données A. En effet, à partir d'une partition U des variables et de la matrice des données A, on estime une partition V des objets par $V=R^{-1}AU$ où R est une matrice de normalisation des lignes. De même, on peut estimer la partition sur les colonnes par $U= C^{-1}A^{T}V$ où C est une matrice de normalisation des colonnes. Ainsi, le vecteur de classification de colonnes recherché U est le vecteur propre de la matrice $C^{-1}A^{T}R^{-1}A$ et le vecteur de classification de lignes V est le vecteur propre de la matrice $R^{-1}AC^{-1}A^{T}$. Une décomposition en valeurs propres permet de résoudre ces deux problèmes. Kluger et al. proposent de normaliser conjointement les lignes et les colonnes de la matrice par un prétraitement itératif. Les vecteurs propres associés aux plus grandes valeurs propres sont partitionnés selon différents nombres de classes et différentes valeurs seuils.

5.4. *Méthodes probabilistes*

5.4.1. *Méthodes basées sur le modèle de mélange*

Pour traiter le problème de la classification croisée par l'approche des modèles de mélange, Govaert et Nadif proposent des extensions des algorithmes EM et CEM qui recherchent une double partition des individus et des variables. L'algorithme Bloc-CEM (Govaert et Nadif, 2003) maximise alternativement la log-vraisemblance classifiante conditionnelle à la partition en colonne et la log-vraisemblance classifiante conditionnelle à la partition en ligne. Ainsi, il utilise de façon alternée l'algorithme CEM sur les individus en bloquant la partition en colonne puis sur les variables en bloquant la partition en ligne. L'algorithme Bloc-EM consiste à maximiser alternativement l'espérance de la log-vraisemblance classifiante conditionnellement à la partition en colonne, puis conditionnellement à la partition en ligne. L'algorithme Bloc-EM est plus lent que l'algorithme Bloc-CEM mais il conduit à une estimation plus fiable des paramètres du modèle de mélange (Govaert et Nadif, 2005).

5.4.2. *Méthodes basées sur la théorie de l'information*

Deux variantes d'une même méthode de bi-partitionnement ont été développées d'une manière indépendante dans (Dhillon et al, 2003) et dans (Robardet et al., 2002). Cette méthode consiste à considérer les deux partitions cherchées comme des variables aléatoires à valeurs discrètes et à concevoir la recherche d'une bipartition comme un problème de maximisation de l'association entre ces deux variables. Les deux variantes produisent une partition par un processus d'optimisation locale. Dhillon et al. utilisent la mesure de divergence entre distribution de probabilités de Kullback et Leibler et proposent de fixer *a priori* le nombre de classes de chacune

des deux partitions puis optimisent localement la fonction en estimant itérativement une partition en fonction de l'autre jusqu'à la convergence. Robardet et al. ne fixent pas *a priori* le nombre de classes des deux partitions et utilisent un algorithme d'optimisation locale stochastique qui procède également par ajustement itératif d'une partition en fonction de l'autre.

Tableau 2. *Tableau comparatif des algorithmes de bi-partitionnement*

Algorithme	Fixer le nombre de (Bi)classes	Type de biclasses	Tableau de données	Nombre de biclasses	Application
One-way splitting	Non	Valeurs constantes	Tableau de mesures	Deux à la fois	Cadre général
Two-way splitting	Oui	Valeurs constantes	Tableau de mesures	Deux à la fois	Cadre général
CROEUC	Oui	Valeurs cohérentes	Tableau de mesures	Plusieurs à la fois	Cadre général
CROKI2	Oui	Valeurs cohérentes	Tableau de contingence	Plusieurs à la fois	Cadre général
CROBIN	Oui	Valeurs cohérentes	Tab. données binaires	Plusieurs à la fois	Cadre général
CTWC	Non	Constantes sur les colonnes	Tableau de mesures	Plusieurs à la fois	Bioinformatique
Plaid Models	Oui	Valeurs cohérentes	Tableau de mesures	Une	Bioinformatique
δ-biclusters	Oui	Valeurs cohérentes	Tableau de mesures	Une	Bioinformatique
δ-patterns	Non	Constantes sur les lignes	Tableau de mesures	Toutes à la fois	Bioinformatique
ITWC	Non	Valeurs cohérentes	Tableau de mesures	Plusieurs à la fois	Bioinformatique
DCC	Non	Valeurs constantes	Tableau de mesures	Toutes à la fois	Bioinformatique
OPSM	Non	Evolution cohérente	Tableau de mesures	Une	Bioinformatique
SAMBA	Non	Evolution cohérente	Tableau de mesures	Toutes à la fois	Bioinformatique
FLOC	Oui	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Bioinformatique
Spectral	Non	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Bioinformatique
(Dhillon et al, 2003)	Oui	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Classification des documents
(Robardet, 2002)	Non	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Cadre général
CEM	Oui	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Cadre général
BCEM	Oui	Valeurs cohérentes	Tableau de mesures	Toutes à la fois	Cadre général

6. Conclusion

Dans ce papier nous avons essayé de synthétiser les différents algorithmes de bi-partitionnement proposés et qui diffèrent selon la méthode de bi-partitionnement, le type et le nombre de biclasses obtenues et le cadre d'application. Certains de ces algorithmes présentent l'avantage de pouvoir identifier les biclasses sans avoir besoin de fixer *a priori* le nombre de biclasses ou le nombre de classes en ligne et en colonne. D'autres algorithmes, en particulier ceux utilisés pour l'analyse des biopuces, nécessitent des modifications pour pouvoir les appliquer sur des données différentes. Ainsi, ce travail constitue un premier pas sur le chemin d'étude des algorithmes de classification simultanée. La validation des résultats de bi-partitionnement et la mesure de la qualité des biclasses nécessitent également une étude approfondie afin de sélectionner les meilleurs algorithmes.

7. Bibliographie

Ben-Dor A., Chor B., Karp R. et Yakhini Z., "Discovering local structure in gene expression data: The order–preserving submatrix problem", *In Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02)*, 2002, p. 49–57.

Busygin S., Jacobsen G. et Kramer E., "Double conjugated clustering applied to leukemia microarray data", *In Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data*, 2002.

Califano A., Stolovitzky G., et Tu Y., «Analysis of gene expression microarrays for phenotype classification », *In Proceedings of the International Conference on Computational Molecular Biology*, 2000, p. 75–85.

Cheng Y. et George M. Church., "Biclustering of expression data". *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00),* 2000, p. 93-103.

Dhillon I.S., Mallela S. et Modha D.S. "Information-theoretic co-clustering", *In ACM SIGKDD, Washington, DC,USA, ACM,* 2003, p.89-98.

Getz G., Levine E. et Domany E., "Coupled two-way clustering analysis of gene microarray data", *Proc. Natl. Acad. Sci. USA*, 97(22):12079Œ84, 2000.

Govaert G., Classification croisée, Thèse de doctorat d'état, Paris, 1983.

Govaert G.et Nadif M., "Clustering with block mixture models", *Journal of the Pattern Recognition" 36,* 2003, p.463-473.

Govaert G. et Nadif M.,"An EM Algorithm for the Block Mixture Model",*IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol27, N°4*,April 2005, p.1-5.

Hartigan J., "Direct Splitting". Chap. 14, Dans Clustering Algorithms. John Wiley & Sons, New York, 1975, p.251-277.

Kluger Y.,Basri R., Joseph T. et Gerstein C.M., “Spectral biclustering of microarray data: coclustering genes and conditions”, In *Genome Research*, vol. 13, 2003, p. 703–716.

Lazzeroni L. et Owen A., “Plaid models for gene expression data”. Technical report, Stanford University, 2000.

Madeira S.C. et Oliveira A.L., “Biclustering Algorithms for Biological Data Analysis: A Survey”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:24-45, 2004.

Robardet C., Contribution à la classification non supervisée : proposition d'une méthode de bi-partitionnement, thèse de doctorat, Université Claude Bernard - Lyon 1.

Tanay A., Sharan R., et Shamir R., “Discovering statistically significant biclusters in gene expression data”, In *Bioinformatics*, vol. 18 (Suppl. 1), 2002, p. S136–S144.

Tanay A., Sharan R., et Shamir R, "Biclustering Algorithms: A Survey", In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman, 2004.

Tang C., Zhang L., Ramanathan M., Zhang A., “Interrelated Two-way Clustering: An Unsupervised Approach for Gene Expression Data Analysis”, *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, 2001, p.41.

Yang, J., Wang H., Wang W. et Yu P., “Enhanced biclustering on expression data”, In: *3rd IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2003), IEEE Computer Society,* Los Alamitos, CA, 2003, p. 321–327.