Une approche de co-classification automatique à base des cartes topologiques

Kais Allab, Khalid Benabdeslem, Alexandre Aussem

Une approche de co-classification automatique à base des cartes topologiques

Kais Allab *
Khalid Benabdeslem **, Alexandre Aussem **

*allab.kais@gmail.com,
**{kbenabde, aaussem}@univ-lyon1.fr
Université de Lyon, F69622-Lyon, France
Université de Lyon1 - Villeurbanne

Résumé. Nous présentons dans ce papier une nouvelle approche de co–classification automatique sur les tableaux de données continues. Cette approche est basée sur les cartes topologiques de Kohonen que nous appelons Bi-SOM (*Bi-clustering based on one Self-Organizing Map*). En outre de la question principale de la co-classification automatique liée au traitement simultané des lignes et des colonnes d'une matrice de données, nous proposons dans cette approche de répondre à plusieurs problématiques liées à cette tâche, à savoir: (1) la visualisation topologiques de bi-clusters avec une notion de voisinage, (2) l'optimisation de ces dits bi-clusters dans des macro-blocs et (3) la réduction de dimension par élimination itérative de blocs de "bruit". Enfin, nous présentons des résultats issus des expérimentations faites sur plusieurs bases de données réelles et d'autres synthétiques pour valider notre approche en comparaison avec d'autres méthodes de co-classification automatique.

1 Introduction

La classification non-supervisée est une tâche primordiale en apprentissage automatique et data mining. Elle consiste à représenter la meilleure structure induite de la distribution d'un ensemble de données non étiquetées. Dans ce cadre, son principe revient à regrouper les données en classes (clusters) en respectant deux propriétés intéressantes, à savoir la cohésion et la séparation. Ce regroupement est fait seulement sur les lignes (individus) en fonction de toutes les colonnes (variables) de la matrice de données. Or, il est clair que dans la partition finale, chaque classe d'individus est caractérisée par un sous ensemble de variables qui participent le plus à sa construction. Il est donc plus approprié de procéder par une recherche simultanée des relations spécifiques qui peuvent exister entre les individus et les variables. Cette démarche s'inscrit dans le cadre de la classification par blocs (block clustering, bi-clustering, co-classification ou classification croisée) qui consiste à réorganiser la matrice de données en blocs homogènes, suivant une certaine mesure de similarité (Govaert et Nadif, 2009).

Depuis la présentation de *Block Clustering* (Hartigan, 1972), le premier algorithme de co-classification qui remonte à plus de trente huit ans, plusieurs autres algorithmes ont été proposés dans divers domaines d'application, particulièrement en traitement d'images (Shi et Malik,

2000), text mining (Dhillon, 2001) et analyse des données biologiques (Madeira et Oliveira, 2004). Dans l'approche de classification directe (*Block Clustering*), la matrice de données est divisée en plusieurs sous matrices qui correspondent aux blocs. La division d'un bloc dépend ainsi de la variance de ses valeurs. En effet, plus la variance est faible plus le bloc est constant. La qualité de la partition est évaluée par la somme des variances des différents blocs ainsi trouvés (Hartigan, 1972).

En 1975, Le même auteur a proposé deux autres algorithmes de co-classification (Hartigan, 1975) : le premier algorithme (*One-Way Splitting*) se focalise principalement sur la partition des individus en essayant de construire une partition avec des variables ayant une variance intra-classe supérieure à un certain seuil pour découper la classe associée. Comme toute technique de classification directe, un seuil minimal induit un nombre important de classes de faible densité et vice-versa. Le deuxième algorithme de Hartigan (*Two-Way Splitting*) procède par division successive des lignes et des colonnes en calculant à chaque itération un grand nombre de variances, ce qui rend son utilisation inadaptée sur des données de grande taille. De plus, le choix du seuil reste hypothétique et nécessite soit une connaissance préalable de celui-ci, soit un test sur plusieurs valeurs. En 1983, Govaert a proposé trois algorithmes de co-classication, l'algorithme *Croeuc* pour les données continues, l'algorithme *Crobin* pour les données binaires et l'algorithme *Croki2* dédié aux tableaux de contingence (Govaert, 1983). Les trois algorithmes consistent à optimiser les partitions des lignes et des colonnes par à un algorithme itératif en se basant sur la somme des distances euclidiennes comme fonction objective à minimiser. De par leur simplicité et leur rapidité, ces algorithmes peuvent s'appliquer sur des données comparables de grande taille. Cependant, ils nécessitent la connaissance a priori du nombre de classes en lignes et en colonnes.

L'inconvénient majeur des méthodes divisives, soit en utilisant le partitionnement double (*Croeuc*, *Crobin* et *Croki2*), ou la classification direct (*Block Clustering*, *One-Way Splitting* et *Two-Way Splitting*) est qu'on ne peut pas remettre en question une division, une fois faite, ce qui peut engendrer la perte de bi-clusters potentiellement intéressants en les divisant avant de les identifier. Pour remédier à ce problème, Cheng et Church ont proposé en 2000, $\delta-clusters$, une méthode de recherche gloutonne itérative basée sur la création de bi-clusters en ajoutant des lignes (ou des colonnes) qui maximisent un gain local (Cheng et Church, 2000). Cette approche utilise le résidu quadratique moyen comme mesure de similarité. Elle a été améliorée en 2003 par Yang et al. en proposant la méthode *FLOC* (*Flexible Overlapped Clusters*), en introduisant une fonction supplémentaire liée au traitement des données manquantes et des chevauchements (Yang et al., 2003). En 2002, Tanay et al. ont proposé la méthode *Samba*, à base de graphes, qui consiste à énumérer exhaustivement toutes les cliques modélisant les bi-clusters possibles dans un graphe biparti représentant la matrice de données (Tanay et al., 2002). D'autres modèles statistiques de co-classification on été utilisés dans une approche appelée *Plaid Model* proposée par Lazzaroni et Owen (2002) et dans une méthode de *bi-clustering spectral* proposée par Klugar et al.(2003).

Par ailleurs, on peut citer des méthodes de co-classification générative proposées par Govaert et al. (2008,2009) basées sur les modèles de mélange. Les auteurs partent de l'hypothèse que l'ensemble de la population est représenté par une distribution probabiliste, qui est un mélange des lois de probabilités associées aux classes. Ils estiment simultanément les paramètres du mélange et de la partition. Pensa et al. ont proposé des approches de co-classification qui améliorent la pertinence des partitions en fonction des contraintes imposées par le domaine

d'application. Ces approches sont dédiées aux données binaires (Pensa et Boulicaut., 2008) (Pensa et al., 2010).

On peut citer aussi des méthodes de bi-clustering utilisant les cartes auto-organisatrices de Kohonen (SOM) à savoir DCC (*Double Conjugated Clustering*) (Busygin et al., 2002) et KDISJ (Cottrell et Letrémy, 2005). L'approche DCC souffre de plusieurs inconvénients liés tout d'abord à l'utilisation de deux cartes (une pour les individus et l'autre pour les variables), qui sont construites de manière indépendante avec les mêmes dimensions. Il suffit d'avoir un écart important entre le nombre d'individus et le nombre de variables pour construire une des cartes avec plusieurs classes vides. De plus la co-classification ici est implicite et est représentée par le lien entre chaque neurone dans l'une des deux cartes et son conjugué dans l'autre carte. Il est par ailleurs regrettable que les auteurs n'aient pas comparé l'approche avec d'autres méthodes de co-classification. Plusieurs autres méthodes de bi-clustering utilisant différentes techniques ont été proposées : bi-clustering hiérarchique (Eisen et al., 1998), bi-clustering basé sur des algorithmes évolutionnaires (Mitra et Banka, 2006), bi-clustering bayésien (Meeds et Roweis, 2007), bi-clustering utilisant le recueil simulé (Bryan et al., 2005) et bi-clustering utilisant le random walk (Angiulli et al., 2008).

En analysant les différentes approches de co-classification citées, on peut constater des inconvénients spécifiques à chaque approche et d'autres de nature commune. Le clustering itératif sur les lignes et les colonnes cherche à optimiser une fonction objective appliquée sur l'une des deux partitions (lignes ou colonnes) en fixant l'autre. L'inconvénient de cette approche est que dans certains cas, ce processus de modification alternée peut s'avérer inefficace lorsque les données sont de grande dimension. Par ailleurs, l'inconvénient des approches divisives et des approches de recherche gloutonne est que les décisions prises (division, ajout/suppression de lignes/colonnes) sont irréversibles et peuvent causer la perte de bons bi-clusters (Govaert, 1983) (Cheng et Church, 2000). En outre, la grande complexité des approches statistiques et celles par énumération exhaustive représente aussi un grand handicape lors de leurs application. Enfin, plusieurs méthodes nécessitent l'intervention de l'utilisateur pour fixer les valeurs de quelques paramètres (nombre de bi-clusters souhaité, seuils, etc.).

En contraste de toutes les approches citées ci-dessus, nous proposons une approche de co-classification topologique (Bi-SOM) basée sur les cartes auto-organisatrices (SOM), qui traite simultanément les individus et les variables dans une seule carte. Nous fournissons par ailleurs un outil de visualisation de blocs dans une structure topologique permettant ainsi de définir une métrique entre ces blocs pour un objectif d'optimisation. Enfin, nous définissons un processus itératif de réduction de dimension éliminant les blocs de variables non pertinentes dites de "bruit".

Le reste de cet article est organisé en quatre sections. La section suivante sera consacrée à la définition des cartes topologiques de Kohonen (SOM), ainsi qu'aux propriétés de l'algorithme associé pour la classification automatique. Dans la troisième section, nous présenterons l'approche de co-classification basée sur les cartes topologiques KDISJ (*Kohonen for Disjonctive Table*) (Cottrell et Letrémy, 2005). Ensuite, notre approche proposée (Bi-SOM) et ses différentes tâches seront décrites dans la quatrième section. Enfin, nous fournirons quelques résultats expérimentaux pour valider cette approche en comparaison avec d'autres méthodes de co-classification sur des données réelles et d'autres synthétiques.

2 Cartes auto-organisatrices de Kohonen

Les cartes auto-organisatrices (topologiques ou dites de Kohonen) sont utilisées dans plusieurs domaines d'application pour leur rôle polyvalent (quantification, classification, codage et visualisation) sur les données multidimensionnelles. En effet, outre leur capacité à produire des données similaires au moyen de prototypes comme en quantification vectorielle et/ou en classification, elles autorisent la conservation de la topologie, d'où sa capacité à reproduire des représentations ordonnées, qu'on appelle prototypes, ou vecteurs prototypes, sur un espace de faible dimension qu'on appelle carte.

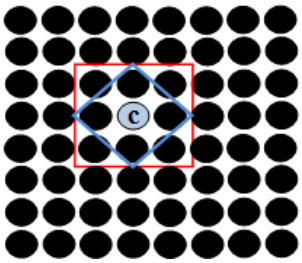


FIG. 1 – *Carte topologique à* 2 *dimensions avec un voisinage d'ordre* 1 *autour du neurone* c*. Voisinage carré (rouge) avec* 8 *voisins et hexagonal (bleu) avec* 4 *voisins.*

Cette carte est décrite par un graphe $C(V, E)$ où V est un ensemble de neurones interconnectés. A chacun de ces neurones, est associé un vecteur prototype w^c de l'espace des données. W est donc l'ensemble de tous les vecteurs prototypes : $W = w^1, w^2, ..., w^m$. L'apprentissage effectué par SOM introduit la conservation de la topologie et impose que deux neurones c, r voisins par rapport à la topologie discrète, soient associés à deux vecteurs w^c et w^r, proches par rapport à la distance choisie entre les données (Fig. 1).

L'algorithme des SOM est proposé en deux versions : stochastique (en ligne) ou batch (en différé, à la façon des nuées dynamiques). Nous optons dans ce papier pour le deuxième cas pour son déterminisme et sa rapidité bien qu'il faille surveiller l'auto-organisation de la carte. Des comparaisons théoriques entre les deux versions peuvent être trouvées dans (Fort et al., 2001). L'algorithme considère en entrée un ensemble de n observations $X = x_1, x_2, ..., x_n \in R^p$ et renvoie en sortie m neurones. A chaque neurone c est associé un ensemble d'observations et un vecteur prototype $w^c \in R^p$. De manière identique à l'algorithme des nuées dynamiques (Diday., 1971), la version batch des SOM procède en deux phases : affectation et adaptation. Lors de la phase affectation, on définit une fonction f de l'espace des données vers la carte C, qui associe à tout élément x_i le neurone dont le vecteur prototype est le plus proche de x_i au sens d'une distance généralisée notée d^T qui fait intervenir tous les neurones de la carte :

$$d^T(x_i, w^{f(x_i)}) = \sum_{r \in C} K^T(\delta(r, f(x_i))) \ \|x_i - w^c\|^2 \tag{1}$$

Ou $K^T(\delta(r, f(x_i)))$ est une fonction de voisinage autour du neurone gagnant issu de $f(x_i)$ et T est le rayon de voisinage qui décroît au cours du temps. $\delta(r, f(x_i))$ est la distance sur la carte entre les deux neurones r et celui issu de $f(x_i)$.

En pratique, On utilise souvent :

$$K^T(\delta(r,c)) = \ e^{-\frac{\delta_{rc}}{2T^2}} \tag{2}$$

La fonction d'affectation, notée f est définie comme suit :

$$f(x_i) = arg \ \ min_{r \in C} \ d^T(x_i, w^r) \tag{3}$$

f introduit donc une partition $P = P_c; c = 1, ..., m$ de l'ensemble des observations où chaque partie est définie par $P_c = x_i \in X$ tel que $f(x_i) = c$. Lors de la phase d'adaptation, l'algorithme met à jour les vecteurs prototypes de la carte par la formule suivante :

$$w^c = \frac{\sum_{r \in C} K_T \ (\delta_{cr}) X_r}{\sum_{r \in C} K_T \ (\delta_{cr}) n_r} \tag{4}$$

Où X_r représente la somme de toutes les observations qui ont été affectées au neurone r et n_r représente le nombre de ces observations. Après l'initialisation de la carte par un système de poids initial, l'algorithme procède en réitérant les deux phases d'affectation et représentation jusqu'à stabilisation.

3 Approche de co-classification basée sur SOM : KDISJ

KDISJ (*Kohonen for Disjonctive Table*) proposée par Cottrell et Letrémy (2005) est une approche de co-classification automatique basée sur les cartes topologiques. L'algorithme KDISJ est une extension de l'algorithme KORRESP (Cottrell et al., 2004) qui a été proposé pour analyser des tableaux de contingence à partir des données qualitatives. Il consiste à classer simultanément les individus et les modalités des variables qualitatives qui les décrivent. L'approche KDISJ sera à la base de l'alternative que nous proposons avec ses avantages et ses inconvénients.

Soit une matrice de données de dimension $(n \times p)$, on construit le tableau disjonctif complet D de dimension $(n \times m)$, tel que m est le nombre des modalités des p variables. Le tableau D est ensuite corrigé D^c par la formule suivante :

$$d^c_{ij} = \frac{d_{ij}}{\sqrt{d_{i.} d_{.j}}}, \ \ d_{i.} = \sum\nolimits_{j=1}^{m} d_{ij} \ , \ d_{.j} = \sum\nolimits_{i=1}^{n} d_{ij}. \tag{5}$$

Notons que $d_{i.}$ représente le nombre de variables p, et que $d_{.j}$ représente l'effectif de la modalité j, donc utiliser la distance euclidienne sur D^c équivaut à utiliser la distance de $\chi 2$ pondérée sur D.

Dans le réseau de Kohonen, on associe à chaque neurone u un vecteur prototype w^u formé de $(m + n)$ composantes, les m premières représentent une ligne de D^c et les n dernières représentent une colonne de D^c. Pour l'apprentissage on tire alternativement une ligne de D^c (un individu i), puis une colonne (une modalité j).

Quand on tire un individu i, on lui associe la modalité $j(i)$ qui maximise d^c_{ij} définie par :

$$j(i) = arg \, max_j \ (d^c_{ij}) = arg \, max_j \ (\frac{d_{ij}}{\sqrt{p d_{.j}}}) \tag{6}$$

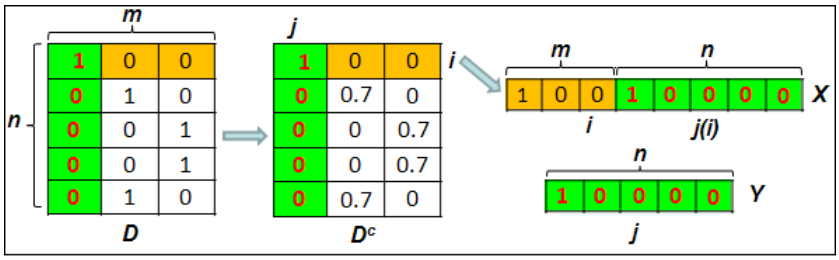


FIG. 2 – *La matrice de données D, la matrice corrigée D^c et les vecteurs X, Y.*

$j(i)$ représente la modalité la plus rare dans la population totale parmi les modalités qui correspondent l'individu i. Cette modalité est la plus caractéristique de cet individu. Dans le cas d'ex-aequos, on choisi aléatoirement une modalité parmi les modalités candidates.

Ensuite, on crée un vecteur individu étendu $X = (i, j(i))$, de dimension $(m + n)$ (Fig. 2). On cherche alors parmi les vecteurs prototypes celui qui est le plus proche au sens de la distance euclidienne restreintes sur les m premières composantes. Soit u^* le neurone gagnant associé à ce vecteur : $u^* = argmin_u \|X_{(1..m)} - w^u_{(1..m)}\|^2$, on adapte alors les m premières composantes de vecteur prototype de ce neurone et de ses voisins par rapport à celles du vecteur X :

$$w^u_{(1..m)}(t+1) = w^u_{(1..m)}(t) + \varepsilon\, \sigma(u, u^*)(X_{(1..m)} - w^u_{(1..m)}(t)) \tag{7}$$

Tel que ε est le paramètre d'adaptation qui est définie comme une fonction décroissante dans le temps t, qui dépend de n le nombre des neurones de la carte. σ est la fonction de voisinage, définie souvent par $\sigma(u, u^*) = 1$ si u et u^* sont voisins dans la carte de Kohonen, et $= 0$ sinon. Le paramètre d'adaptation et le rayon de voisinage changent dans le temps.

Quand on tire une modalité j, de dimension n (une colonne de D^c), on ne lui associe pas de vecteur. Notons par Y le vecteur correspondant à la modalité j (Fig. 2). On cherche alors parmi les vecteurs prototypes des neurones de la carte celui qui est le plus proche, au sens de la distance euclidienne restreinte sur les n dernières composantes.

Soit v^* le neurone gagnant : $v^* = arg\ min_u \|Y_{(m+1..m+n)} - w^u_{(m+1..m+n)}\|^2$, on adapte alors les n dernières composantes du vecteur prototype associé à v^* et de ses voisins par rapport à celles du vecteur Y sans modifier les m premières composantes en utilisant la formule :

$$w^u_{(m+1..m+n)}(t+1) = w^u_{(m+1..m+n)}(t) + \varepsilon\, \sigma(u, v^*)(Y_{(m+1..m+n)} - w^u_{(m+1..m+n)}(t)) \tag{8}$$

On applique ainsi l'algorithme de Kohonen sur les individus d'une part et sur les modalités d'autre part, tout en les maintenant associés. Il suffit en général de $(n + m)$ itérations pour obtenir la convergence (Cottrell et Letrémy, 2005). Après convergence, les individus et les modalités sont classés dans les classes de la carte. Des individus ou modalités proches sont classés dans la même classe ou dans des classes voisines.

Cette approche est dédiée aux données binaires et nécessite donc une discrétisation en cas de données continues, ce qui implique une perte d'informations parfois dramatique. Nous proposons de contourner ce problème par la définition d'un nouveau lien entre les lignes et les colonnes en respectant la nature des données continues.

4 Approche proposée : Bi-SOM

4.1 Principe

Bi-SOM est une approche de co-classification et de réduction de dimension sur des bases de données continues. Le principe de Bi-SOM sur un ensemble de n individus décrits par un ensemble de p variables, consiste à créer une partition constituée de blocs homogènes, contenant à la fois les individus et les variables qui les caractérisent. Nous utilisons cette notion de caractérisation comme le lien permettant d'associer une variable à un individu et de les classer simultanément dans la carte.

Pour la construction de la carte, Bi-SOM utilise une heuristique proposée par Kohonen (1995) basée sur l'Analyse en Composantes Principales (ACP) qui calcule automatiquement les dimensions des cartes et initialise ces vecteurs prototypes (D'autres types d'initialisation peuvent être trouvés dans (Dreyfus, 2002) (Thiria et al., 1997)).

Soit A la matrice de données de dimension $(n \times p)$. Dans la carte, on associe à chaque neurone u un vecteur prototype w^u composé de $(p + n)$ composantes, avec les p premières composantes évoluant dans l'espace des individus (représentées par les lignes de A) et les n dernières composantes évoluant dans l'espace des variables (représentées par les colonnes de A).

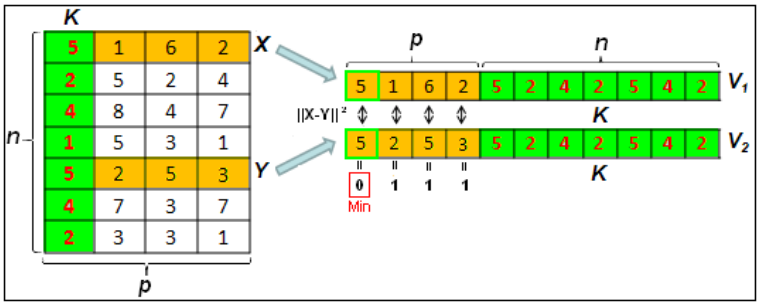


FIG. 3 – *La matrice de données, les deux vecteurs de dimension* $(p + n)$, $V_1 = [XK]$ *et* $V_2 = [YK]$ *créés quand l'élément tiré est une ligne.*

Pour chaque individu X, on cherche l'individu le plus proche Y dans A au sens de la distance euclidienne. Deux vecteurs V_1, V_2 de dimension $(p + n)$ sont ensuite construits en concaténant chacun des deux individus à la variable K qui les caractérise au mieux. La variable K correspond à la valeur minimale des distances entre les composantes de X et celles de Y (Fig. 3). En effet, s'il existe plusieurs minimums pour K, nous construisons plusieurs vecteurs pour la même paire d'individus X et Y avec des colonnes (variables) K différentes.

Pour chaque vecteur V_1 (V_2 respectivement), on cherche parmi les vecteurs prototypes des neurones de la carte, celui qui est le plus proche au sens de la distance euclidienne, (contrairement à KDISJ qui utilise la distance euclidienne restreinte sur les p premières composantes).

Soit u^* le neurone gagnant : $u^* = argmin_u \|V_1 - w^u\|^2$ (Fig. 4). On adapte alors le vecteur prototype de ce neurone u^* et de ses voisins par rapport au vecteur V_1 (V_2 respectivement).

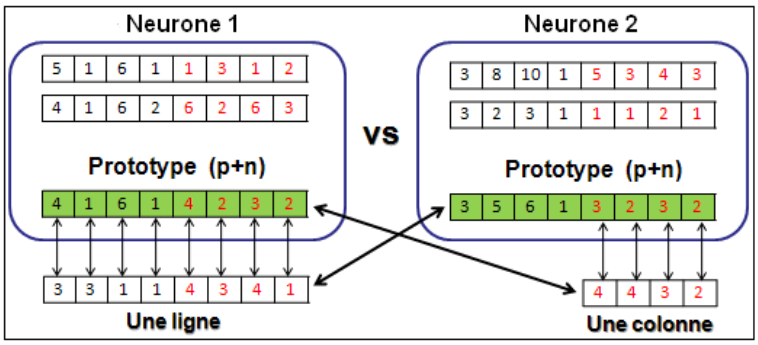


FIG. 4 – *la procédure de compétition entre deux neurones en calculant une distance globale si l'élément pris est une ligne, et une distance partielle sur les n dernières composantes si l'élément pris est une colonne.*

Nous précisons que si Y est l'individu le plus proche de X, X n'est pas forcément l'individu le plus proche de Y. Donc, l'individu Y va appartenir à plusieurs vecteurs qui vont être probablement classés dans des neurones différents. Par conséquent, l'individu Y va appartenir à plusieurs classes. Cette propriété induit le chevauchement des classes en lignes. De la même façon, si une colonne appartient à plusieurs vecteurs (V_i) qui ne sont pas classés dans la même classe, il y aura un chevauchement de classes en colonnes.

Pour chaque variable Z de dimension n (une colonne de A), on cherche parmi les vecteurs prototypes des neurones de la carte celui qui est le plus proche, au sens de la distance euclidienne restreinte sur les n dernières composantes. Soit v^* le neurone gagnant : $v^* = arg\ min_u \|Z_{(p+1..p+n)} - w^u_{(p+1..p+n)}\|^2$ (Fig. 4). On adapte alors les n dernières composantes du vecteur prototype associé à v^* et de ses voisins par rapport à celles de la variable Z, sans modifier les p premières composantes.

Dans la partition résultat, un bloc peut contenir des individus et des variables, il peut être vide comme il ne peut contenir que des variables. Dans ce dernier cas, ces variables ne caractérisent aucun individu. Bi-SOM les considère donc comme étant non pertinentes. Elles seront supprimées au fur et à mesure pour refaire la co-classification sur la nouvelle matrice de données. La procédure est ainsi répétée jusqu'à l'élimination de tous les blocs contenant des variables seules.

Bi-SOM se base sur la notion de voisinage entre les blocs offerte par l'algorithme de Kohonen, pour optimiser la partition résultat après la suppression de toutes les variables non pertinentes, en appliquant une Classification Ascendante Hiérarchique (CAH) sur les prototypes des neurones de la carte. Cette optimisation permet de regrouper les blocs proches en macros blocs. En général, l'indice interne Davies Bouldin (Davies et Bouldin, 1979) est utilisé pour trouver la meilleure troncature du déndrogramme.

4.2 Algorithme

L'algorithme Bi-SOM prend en entrée une matrice de données A de dimension $(n \times p)$, et retourne une partition Γ sur une seule carte topologique. Les vecteurs prototypes des neurones de la carte sont de dimension $(p + n)$ car les vecteurs soumis à cette carte sont le résultat de concaténation d'une ligne et d'une colonne.

Algorithm 1 : **Bi-SOM** (A) retourne Γ ;

Entrée : A (Matrice de données = $n \times p$) ;
Sortie : Γ : partition résultat (blocs) ;
Début
- Initialisation de la carte (elle contient r neurones de dimension $(p + n)$) ;
Répéter
  Répéter
    - Pour un élément x de A (ligne ou colonne)
    Si x est une ligne **Alors**
      - Chercher la ligne y la plus proche de x dans A ;
      Pour Tout K Tel que $K = argmin\|x - y\|^2$ **Faire**
        - Créer $V_1 = concatnation(x, K)$;
        - Créer $V_2 = concatnation(y, K)$;
        - Appliquer l'algorithme de Kohonen sur V_1 et V_2 de dimension $(p+n)$, en affectant chacun au neurone dont le vecteur prototype est le plus proche au sens de la distance euclidienne. (Fig. 4).
      Fin Pour
    Sinon
      - x est une colonne : appliquer l'algorithme de Kohonen sur x, en l'affectant au neurone dont le vecteur prototype est le plus proche au sens de la distance euclidienne restreinte sur les n dernières composantes (Fig. 4).
    Fin Si
  Jusqu'à traitement de toutes les lignes et les colonnes de A
  - $Bruit = \{variables\} \in C_i$ tel que $C_i \cap \{individus\} = \emptyset, i = 1..r$, C_i : le neurone i.
  - $A = A - Bruit$.
Jusqu'à $Bruit = \emptyset$.
- Γ (partition obtenue) = l'ensemble des r neurones obtenus.
- $\Gamma = CAH(\Gamma, DB)$: Optimisation de la carte en appliquant une (CAH) sur les r neurones, et en utilisant l'indice Davies-Bouldin (Davies et Bouldin, 1979) pour trouver le nombre optimal de classes.
- Réarrangement de la matrice de données (lignes et colonnes) selon Γ.
Fin

TAB. 1 – *Les bases de données utilisées*

Données	Type	n	p	# classes	Référence
Breast	Réelle	699	9	2	(Frank et Asuncion, 2010)
Heart	Réelle	303	13	2	(Frank et Asuncion, 2010)
Ovarian	Réelle	54	1536	2	(Schummer et al., 1999)
Leukemia	Réelle	72	1762	2	(Golub et al., 1999)
Prelic1	Synthétique	100	50	10	(Prelic et al., 2006)
Prelic2	Synthétique	100	100	10	(Prelic et al., 2006)
Prelic3	Synthétique	100	50	10	(Prelic et al., 2006)
Prelic4	Synthétique	100	100	10	(Prelic et al., 2006)

5 Expérimentations

5.1 Les données

Bi-SOM est implémentée en Matlab, en utilisant la Toolbox disponible dans `http://www.cis.hut.fi/projects/somtoolbox/`. Dans la première phase des expérimentations, nous avons appliqué l'approche sur des données réelles qui décrivent plusieurs maladies : le cancer du sein (*Breast cancer wisconsin*), une maladie cardiaque (*Heart*), le cancer des ovaires (*Ovarian*) et la maladie de la Leucémie (*Leukemia*) (Tab. 1).

Les données *Leukemia* ont été utilisées pour la première fois par Golub et al. (1999) et par d'autres auteurs par la suite (Klugar et al., 2003) (Madeira et Oliveira, 2004). Le jeu de données est constitué d'un ensemble de 72 individus, correspondant à deux types de Leucémie nommés *ALL (Acute Lymphocytic Leukemia)* et *AML (Acute Myelogenous Leukemia)*, avec 47 ALL et 25 AML. La base contient initialement 7929 variables (gènes d'expression), dans (Busygin et al., 2002) il a été proposé de supprimer les variables de contrôle "affymetrix" et les variables ayant au moins une valeur inférieure à 20 (biologiquement, les niveaux faibles d'expression sont difficile à interpréter), pour obtenir finalement 1762 variables.

Dans la deuxième phase des expérimentations, nous avons utilisé des jeux de données synthétiques utilisées par Prelic et al. (2006) pour la comparaison de sa méthode *RMSBE (Randomized Maximum Similarity Bi-cluster Resolution)* avec d'autres méthodes de co-classification *BiMax* (Prelic et al., 2006), *ISA* (Bergmann et al., 2004), *Samba* (Tanay et al., 2002), *CC* (Cheng et Church, 2000), *OPSM* (BenDor et al., 2003) and *xMotif* (Murali et Kasif, 2003) [1].

Le premier jeu de données *Prelic1* de dimension (100×50) illustre des bi-clusters constants, il contient une première version correcte des données, plus d'autres versions correspondant à des taux de bruitage de 5%, 10%, 15%, 20% et 25%. Le deuxième jeu de données *Prelic2* de dimension (100×100) illustre des bi-clusters additifs, il contient une première version correcte des données, plus d'autres versions correspondant à des taux de bruitage de 2%, 4%, 8% et 10%.

[1] CC : Cheng and Church's Algorithm, OPSM : Order Preserving Submatrix Algorithm, ISA : Iterative Signature Algorithm.

Nous précisons que cette notion de bruit (appliquée sur les données) est différente de la notion des variables non pertinentes détectées et filtrées par Bi-SOM (décrite dans la section 4.3). Le bruitage de données est effectué en ajoutant aux données initiales des valeurs aléatoires suivant une distribution selon la loi normale.

Deux autres jeux de données *Prelic3 et Prelic4* de dimension (100×100) sont utilisés. ils illustrent les deux types de bi-clusters, constants et additifs. Ces jeux de données ne sont pas bruités mais contiennent une première version correcte des données, plus d'autres versions contenant des bi-clusters chevauchés en lignes et en colonnes. Le taux de chevauchement varie de 0 à 8 éléments (lignes et colonnes).

5.2 Critère d'évaluation

Il est toujours délicat de comparer deux méthodes de co-classification à cause de leur caractère non supervisé. Néanmoins, il existe des indices internes et d'autres externes permettant d'évaluer la qualité des partitions obtenues. Nous présentons ainsi une liste d'indices qui est loin d'être exhaustive.

5.2.1 Indices internes.

Les indices internes mesurent la relation entre l'information intrinsèque des données et la partition résultat. Un indice interne est calculé à partir de deux matrices P et M. La matrice P contient les informations sur la proximité entre les éléments (lignes ou colonnes) tel que : $P_{ij} = P_{ji} = distance(x_i, x_j)$.

La matrice M est calculée de telle sorte que les grandes valeurs correspondent aux lignes (ou colonnes) qui ne sont pas ensemble dans les bi-clusters. Par exemple $M_{ij} = 1/(1+k)$ tel que k est le nombre de fois où i et j sont classés ensemble. M_{ij} est inclus entre 0 et 1, il est égale à 1 si i et j ne sont jamais classés dans la même classe et proche de 0 s'ils sont souvent classés dans la même classe. Ces deux matrices sont comparées en utilisant l'indice de Hubert (Prelic et al., 2006) :

$$H(P,M) = \frac{\frac{1}{h}\sum_{i=1}^{l-1}\sum_{j=i+1}^{l}(P_{ij}-\mu_P)(M_{ij}-\mu_M)}{\sigma_P \sigma_M} \tag{9}$$

Tel que l est le nombre d'objets de la matrice P et $h = l(l-1)/2$. μ_P, μ_M sont les moyennes des matrices P et M et σ_P, σ_M sont leurs variances, respectivement. On peut donc calculer l'indice H_L pour les individus (lignes) et l'indice H_C pour les variables (colonnes), puis combiner les deux résultats dans :

$$H_{LC} = \frac{nH_L + pH_C}{n+p} \tag{10}$$

Tel que n est le nombre d'individus et p est le nombre de variables dans la matrice de données. Une meilleure co-classification est celle qui minimise la valeur de H_{LC}.

5.2.2 Indices externes.

Les indices externes sont utilisés pour comparer la partition calculée avec la partition correcte des données, quand elle est disponible. Ces indices peuvent être utilisés pour comparer deux partitions différentes du même jeu de données.

Soient S et R deux partitions, et F une matrice de dimension $(s \times r)$ tel que s est le nombre de bi-clusters dans S et r est le nombre de bi-clusters dans R. Soit i un bi-cluster de S et j un bi-cluster de R, F_{ij} désigne la similarité entre i et j. On utilise la matrice F pour calculer l'indice *F-score* (Xiaowen et Wang, 2007).

- Soient g_i le nombre d'individus dans i et c_i le nombre de variables dans i, $n_i = g_i c_i$;
- Soient g_j le nombre d'individus dans j et c_j le nombre de variables dans j, $n_j = g_j c_j$;
- $g_{i \cap j}$ est le nombre d'individus appartenant aux deux bi-clusters i et j ;
- $c_{i \cap j}$est le nombre de variables appartenant aux deux bi-clusters i et j.

$$\textit{F-score(i,j)} = \frac{2(g_{i \cap j})(c_{i \cap j})}{n_i + n_j} \tag{11}$$

A partir de (11), on peut calculer la pertinence globale de la partition R par rapport a la partition S :

$$Pertinence(R, S) = \frac{1}{r} \sum_{i=1}^{r} max_{j=1}^{s}(\textit{F-score(i,j)}) \tag{12}$$

Pour évaluer Bi-SOM, On peut aussi calculer le degré de similarité entre la partition résultat et la partition correcte en utilisant le score d'appariement (Match Score) (Bandyopadhyay et al., 2007). Soit A la matrice de données de dimension $(n \times p)$, soit S la partition correcte et R la partition obtenue par Bi-SOM. G_1 le nombre de lignes dans S, G_2 le nombre de lignes dans R, C_1 le nombre de colonnes dans S et C_2 le nombre de colonnes dans R, le score d'appariement entre S et R est calculé comme suit :

$$Score(S, R) = \frac{1}{|S|} \sum_{(G_1, C_1) \in S} max_{(G_2, C_2) \in R} \frac{|G_1 \cap G_2| + |C_1 \cap C_2|}{|G_1 \cup G_2| + |C_1 \cup C_2|} \tag{13}$$

Enfin, trois autres indices externes sont utilisés à savoir l'indice de Jaccard (Jaccard, 1901), l'indice de Rand (IR) (Rand, 1971) et l'indice de Rand ajusté (IRA) (Bandyopadhyay et al., 2007). Notons par :

- a : le nombre de paires d'objets classés ensemble dans S et dans R.
- b : le nombre de paires d'objets classés ensemble dans S mais pas dans R.
- c : le nombre de paires d'objets classés ensemble dans R mais pas dans S.
- d : le nombre de paires d'objets qui ne sont classés ensemble ni dans S ni dans R.

$$Jaccard(S, R) = \frac{a}{a + b + c} \tag{14}$$

$$IR(S, R) = \frac{a + b}{a + b + c + d} \tag{15}$$

$$IRA(S, R) = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)} \tag{16}$$

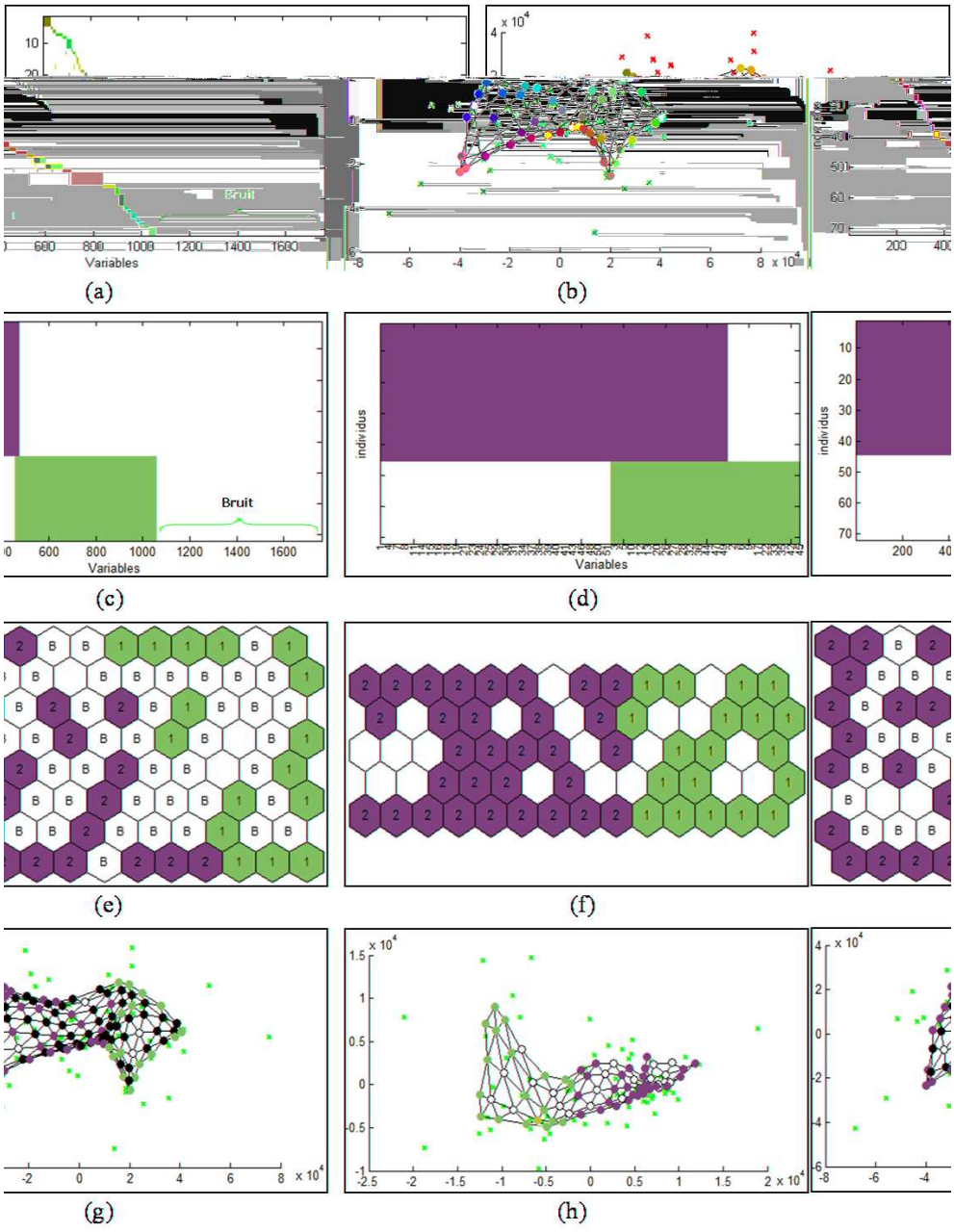


FIG. 5 – *Résultats de la co-classification sur la base Leukemia : (a) les blocs résultat (b) la carte projetée sur le même espace que les données (c) les macro-blocs résultat (d) les macro-blocs après la réduction de dimension (e) et (g) la carte optimisée avant la réduction de dimension (f) et (h) la carte optimisée après la réduction de dimension.*

5.3 Résultats

5.3.1 Résultats de la co-classsification.

Dans cette phase d'expérimentation, Bi-SOM a été appliquée sur la base *Leukemia* afin de montrer et visualiser les résultats de la co-classification et de la réduction de dimension. Après l'apprentissage de la carte de dimension (13×8), on a créé une matrice binaire D, tel que $D_{ij} = 1$ si la ligne i et la colonne j sont classés ensemble, $D_{ij} = 0$ sinon. Cette matrice est réarrangée pour visualiser les blocs résultat tel que chaque bloc dans Fig. 5(a) est représenté par le neurone ayant la même couleur dans Fig. 5(b). Ensuite, on a appliqué une Classification Ascendante Hiérarchique (CAH) sur les vecteurs prototypes des neurones issus de Bi-SOM pour les regrouper en macro-classes.

Les deux Figures Fig. 5(e) et Fig. 5(g) montrent les neurones de la carte répartis en deux classes (1 :AML et 2 : ALL) correspondant aux deux macro-blocs dans la Figure Fig. 5(c). Les deux Figures Fig. 5(f) et Fig. 5(h) montrent la répartition des neurones de la carte (avec des dimensions réduites et sans bruit) en deux classes correspondant aux deux macro-blocs dans la Figure Fig. 5(d) après la réduction de dimension. Les neurones ayant le label (B) dans la Figure Fig. 5(c) représentent les classes contenant des variables non pertinentes et qui correspondent aux neurones noirs dans Fig. 5(g). Les neurones blancs et sans label dans Fig. 5(g) et Fig. 5(h) représentent les neurones passifs, ce sont des neurones vides qui ne contiennent ni individu ni variable, leur présence représente la propriété importante du lissage liée à l'algorithme SOM.

Discussion. Après l'apprentissage, les neurones de la carte prennent des positions en fonction de la distribution des données (Fig. 5(b)). On peut remarquer qu'après le réarrangement de la matrice binaire D, chaque bi-cluster correspond à un neurone actif qui contient des individus et des variables (Fig. 5(a) et Fig. 5(b)). On remarque aussi que notre méthode tolère le chevauchement, en lignes et en colonnes, une étude plus détaillée sera donnée par la suite à ce sujet.

La remarque la plus importante, et que l'espace blanc dans les deux Figures Fig. 5(a) et Fig. 5(c), après la fin de la diagonale des bi-clusters, correspond à des valeurs nuls de D, donc à des variables qui ne sont pas classées dans des classes contenant des individus. Ces variables sont considérés comme étant non pertinentes vis-à-vis de la co-classification et sont aussi supprimées de la matrice de données initiale. Cette réduction de dimension va engendrer la réduction des dimensions de la carte utilisée pour la classification (Fig. 5(g) avant la sélection de variables et Fig. 5(h) après la sélection de variables).

Pour des fins de comparaison de notre approche avec d'autres approches de co-classification, nous avons utilisé l'outil **BicAT** *(Bi-clustering Analysis Toolbox)* disponible dans `http://www.tik.ee.ethz.ch/~sop/bicat/`. C'est un outil d'analyse de données biologiques qui inclut deux méthodes de classification classique (K-means et CAH) et cinq approches de co-classification : *BiMax* (Prelic et al., 2006), *ISA* (Bergmann et al., 2004), *Samba* (Tanay et al., 2002), *CC* (Cheng et Church, 2000) et *OPSM* (BenDor et al., 2003).

Nous avons appliqué toutes les méthodes sur les données réelles (Breast, Heart, Ovarian et Leukemia). Les performances des partitions obtenues sont présentées dans (Tab. 2, Tab. 3, Tab. 4 et Tab. 5). Notons que **BicAT** n'a pas détecté des bi-clusters en appliquant la méthode *ISA* sur la base *Breast* (Tab. 2).

Indice	K-means	CAH	ISA	CC	OPSM	BiMax	Bi-SOM
Jaccard	0,242	0,548	–	0,618	0,511	0,650	**0,884**
Rand	0,583	0,553	–	0,737	0,576	0,730	**0,934**
Rand Ajusté	0,219	0,016	–	0,492	0,107	0,448	**0,866**
Score	0,194	0,137	–	0,173	0,458	0,049	**0,928**
Pertinence	0,105	0,260	–	0,156	0,136	0,007	**0,963**

TAB. 2 – *Performances de Bi-SOM sur Breast* (699×9).

Indice	K-means	CAH	ISA	CC	OPSM	BiMax	Bi-SOM
Jaccard	0,118	0,490	0,500	0,403	0,502	0,376	**0,547**
Rand	0,513	0,502	0,500	0,530	0,502	0,522	**0,581**
Rand Ajusté	0,030	0,007	0,005	0,060	0,000	0,044	**0,128**
Score	0,125	0,114	0,428	0,114	0,372	0,227	**0,603**
Pertinence	0,061	0,264	0,267	0,114	0,124	0,292	**0,683**

TAB. 3 – *Performances de Bi-SOM sur Heart* (303×13).

Indice	K-means	CAH	ISA	CC	OPSM	BiMax	Bi-SOM
Jaccard	0,208	0,453	0,396	0,497	0,497	0,585	**0,701**
Rand	0,546	0,491	0,586	0,497	0,497	0,620	**0,747**
Rand Ajusté	0,088	0,013	0,171	0,000	0,000	0,474	**0,534**
Score	0,150	0,130	0,394	0,556	0,324	0,686	**0,898**
Pertinence	0,092	0,259	0,228	0,133	0,443	0,814	**0,927**

TAB. 4 – *Performances de Bi-SOM sur Ovarian* (54×1536).

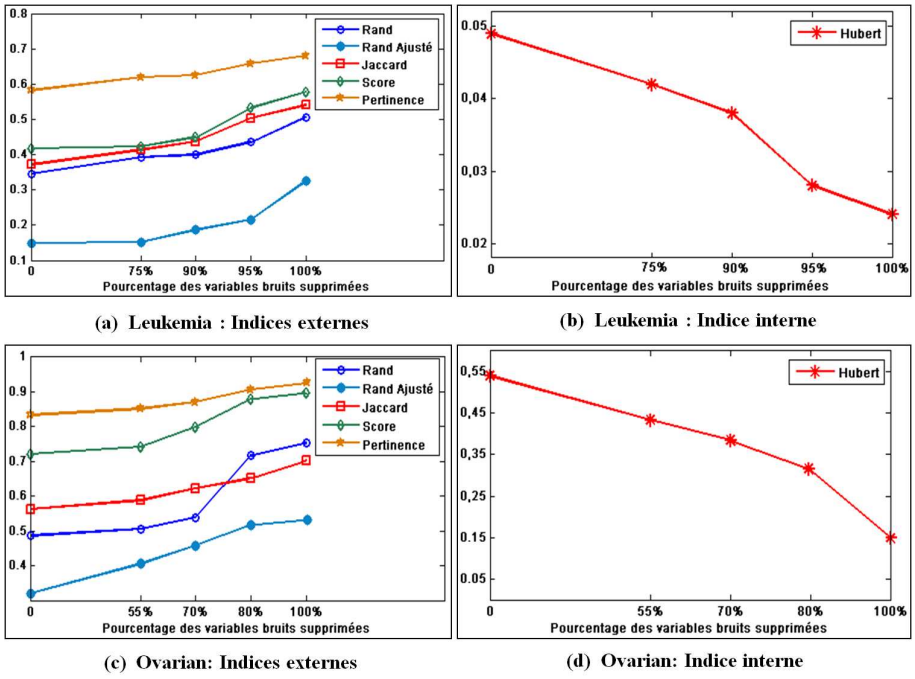
Indice	K-means	CAH	ISA	CC	OPSM	BiMax	Bi-SOM
Jaccard	0,204	0,529	0,515	0,393	0,520	0,564	**0,540**
Rand	0,549	0,559	0,489	0,516	0,520	0,588	**0,604**
Rand Ajusté	0,148	0,059	0,017	0,021	0,000	0,284	**0,325**
Score	0,169	0,162	0,435	0,099	0,241	0,662	**0,578**
Pertinence	0,118	0,255	0,163	0,106	0,145	0,702	**0,681**

TAB. 5 – *Performances de Bi-SOM sur Leukemia* (72×1762).

Discussion. En général, les résultats de Bi-SOM sont très encourageants sur tous les indices utilisés. On peut remarquer que Bi-SOM a fourni des partitions de meilleure qualité par rapport à toutes les autres méthodes, sauf dans certaines cas par rapport à la méthode BiMax. Par exemple, pour la base *Leukemia*, BiMax a produit des meilleurs résultats de l'indice de Jaccard, le score d'appariement et la pertinence.

L'inconvénient majeur de BiMax est sa complexité qui est de l'ordre de $(O(n\,p\beta min\,(n,p))$ pour une matrice de données de dimension $(n \times p)$. tel que β est le nombre des bi-clusters à inclusion maximale (calculé initialement par BiMax : $\beta = 2^{min\,(n,p)} - 1$) (Prelic et al., 2006). Par contre, la complexité de Bi-SOM est de l'ordre de $(O\,(r\,p\,(2n+1)))$. Au pire des cas, pour une matrice de données de dimension $(n \times p)$, chaque ligne sera concaténé à toutes les colonnes en construisant à chaque fois deux vecteurs (la ligne et sa plus proche voisine). Les colonnes sont prises telle qu'elle sont. Donc, le nombre des éléments à traiter par SOM est de $(2np)$ vecteurs (pour les lignes) plus (p) vecteurs (pour les colonnes). La complexité de l'algorithme SOM est $(O(rN))$, tel que N est le nombre des lignes dans la matrice de données et r est le nombre de neurones de la carte. Par conséquent, la complexité de l'algorithme Bi-SOM est de $(O\,(r\,p\,(2n+1)))$.

5.3.2 Résultats de la réduction de dimension.



(a) Leukemia : Indices externes **(b) Leukemia : Indice interne**

(c) Ovarian: Indices externes **(d) Ovarian: Indice interne**

FIG. 6 – *Évaluation des partitions obtenues par Bi-SOM sur Ovarian et Leukemia.*

Pour Bi-SOM, une variable est non pertinente si elle ne caractérise aucun individu, en autres termes, si elle est classée dans des neurones ne contenant pas des individus. Bi-SOM procède ainsi par une sélection itérative sur l'ensemble de variables. Elle les supprime au fur et à mesure, pour refaire la co-classification avec la nouvelle base obtenue. Cette opération est répétée jusqu'à ce qu'aucune nouvelle variable non pertinente n'est détectée. A chaque itération on calcule les performances de la partition obtenue.

Bi-SOM n'a pas détecté des variables non pertinentes dans les deux bases *Breast* et *Heart* qui contiennent (9) et (13) variables respectivement. Par contre, le taux de variables non pertinentes détectées est de (97%) pour la base *Leukemia* et de (65%) pour la base *Ovarian*. On peut remarquer dans les Figures (Fig. 6(a) et Fig. 6(c)) que les performances de Bi-SOM sont faibles quand elle est appliquée sur les bases initiales (sans sélection de variables), et qu'elle s'améliorent après chaque itération (suppression d'un paquet de variables non pertinentes). Les indices externes (à maximiser) commencent par des valeurs faibles pour arriver à chaque itération à des valeurs supérieures. Par contre, l'indice interne d'Hubert (à minimiser) ((Fig. 6(b) et Fig. 6(d))) indique la qualité de la partition obtenue par Bi-SOM, en prenant en compte les individus et les variables des bi-clusters. Les valeurs obtenues sont toujours proches de (0) et montrent que cet indice est influé par la suppression des variables non pertinentes. Il réalise sa meilleure performance avec la base totalement filtrée, (0.024) pour la base *Leukemia* et (0.150) pour la base *Ovarian*.

5.4 Autres comparaisons

Dans la deuxième phase d'expérimentations, nous avons appliqué Bi-SOM sur les jeux de données synthétiques fournis par (Prelic et al., 2006), contenant différents scénarios permettant d'étudier la sensibilité de l'approche face aux données bruités et au chevauchement, pour deux types de bi-clusters (constants et additifs).

Soit $A(I, J)$ est une matrice de données de dimension $(n \times p)$ et $i^* \in I$ la ligne référence. Un bi-cluster $A(I', J')$ tel que $I' \subseteq I$ et $J' \subseteq J$ est un bi-cluster constant par rapport à la ligne référence i^* si pour tout $i \in I'$ et tout $j \in J'$, $a_{ij} = a_i \times j$. Une sous-matrice $A(I', J')$ avec l'ensemble des lignes I' et l'ensembles des colonnes J' est un bi-cluster additif par rapport à la ligne référence i^* si pour tout $i \in I'$ et tout $j \in J'$, $a_{ij} - a_i \times j = c_i$, tel que c_i est constant pour toute ligne i (Fig. 7).

1.0	**2.0**	**3.0**	**4.0**
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

(a)

1.0	**2.0**	**5.0**	**0.0**
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

(b)

FIG. 7 – *Exemples des deux types de bi-clusters. les lignes en gras sont les lignes références (i^*). (a) Bi-cluster constant. (b) Bi-cluster additif.*

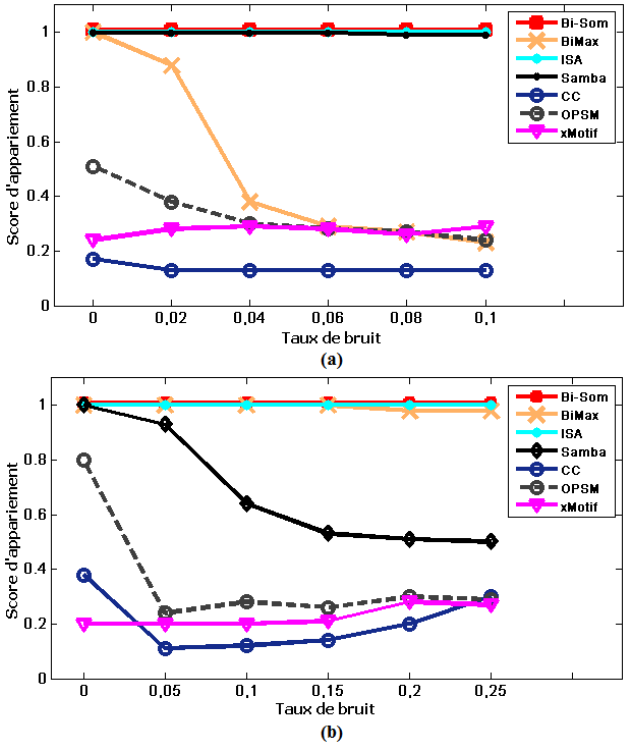


FIG. 8 – *Résultats de comparaison entre Bi-SOM et quelques méthodes de co-classification. (a) Bi-cluster constant (*Prelic1*). (b) Bi-cluster additif (*Prelic2*).*

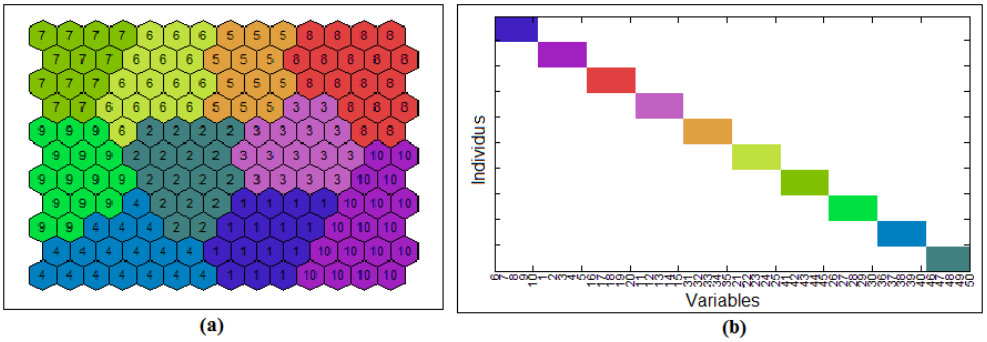


FIG. 9 – *Résultats de Bi-SOM appliquée sur* Prelic1 *Avec un taux de bruitage de* 5%. *(a) la carte segmentée. (b) Les blocs résultat.*

5.4.1 Influence du bruit.

Dans cette phase, nous avons appliqué Bi-SOM sur les jeux de données *Prelic1* (pour les bi-clusters constants) et *Prelic2* (pour les bi-clusters additifs). L'objectif est d'analyser sa sensibilité face à des données bruités, en utilisant le score d'appariement comme mesure de qualité.

La Figure (Fig. 8) montre la comparaison des résultats obtenus par Bi-SOM avec ceux des méthodes *(BiMax, ISA, Samba, CC, OPSM et xMotif)* obtenus par Prelic et al. (2006). En l'absence du bruit, Bi-SOM, *ISA, Samba* ont réalisé des scores supérieurs à 99%, aussi bien pour les bi-clusters constants que des bi-clusters additifs. *BiMax* a réalisé des scores instables et même faibles dans le cas des bi-clusters additifs. Les autres méthodes ont des résultats faibles pour les deux scénarios. Le plus important dans notre cas est qu'en présence du bruit, Bi-SOM réalise un excellent score d'appariement souvent égale à 100% (Fig. 8), ce qui signifie que la méthode est stable face à des données bruités, et surtout compétitive à d'autres méthodes telle que *ISA* et *Samba*.

La Figure (Fig. 9) montre les résultats obtenues par Bi-SOM appliquée sur la base de données *Prelic1* relative aux bi-clusters constants, avec un taux de bruitage de 5%. La Figure Fig. 9(b) montre les blocs (bi-clusters) résultats après la segmentation de la carte. Chaque bloc correspond à l'ensemble des neurones ayant la même couleur dans la Figure Fig. 9(a). On peut remarquer que Bi-SOM a obtenu le nombre optimal de dix (10) blocs, correspondant à la partition correcte (10 blocs de dimension (10×5)), non chevauchés). Les blocs sont visualisés sous forme d'une diagonale parfaite (Fig. 9(b)), selon une topologie illustrée dans la carte de la Figure Fig. 9(a).

5.4.2 Influence du chevauchement.

Dans cette phase, nous avons appliqué Bi-SOM sur les jeux de données *Prelic3* (pour les bi-clusters constants) et *Prelic4* (pour les bi-clusters additifs), permettant d'analyser son comportement face au chevauchement, en utilisant le score d'appariement comme mesure de qualité.

BiMax est la seule méthode qui est stable quelque soit le taux de chevauchement et quelque soit le type des bi-clusters. Bi-SOM, *Samba* et *ISA* réalisent des bons scores mais elles sont sensibles à l'augmentation du taux de chevauchement (*ISA* est la plus sensible). Bien que *OPSM* ne soit pas affectée par le degré de chevauchement dans le cas des bi-clusters additifs, elle reste inefficace face aux valeurs identiques des bi-clusters constants, ce qui explique l'absence de sa courbes dans la Figure Fig. 10(a).

La Figure (Fig. 11) montre les résultats de Bi-SOM appliquée sur le jeu de données *Prelic4* relatif aux bi-clusters additifs, avec un taux de chevauchement de 10% (en lignes et en colonnes). La Figure Fig. 11(b) montre les blocs (bi-clusters) résultats après la segmentation de la carte. Chaque bloc correspond à l'ensemble des neurones ayant la même couleur dans la Figure Fig. 11(a). Bi-SOM a obtenue dix blocs de dimension (10×10), chaque bloc est chevauché par une ligne et une colonne avec le bloc qui le précède dans la diagonale, ce qui correspond totalement à la partition correcte de ce jeu de données (le score d'appariement = 1).

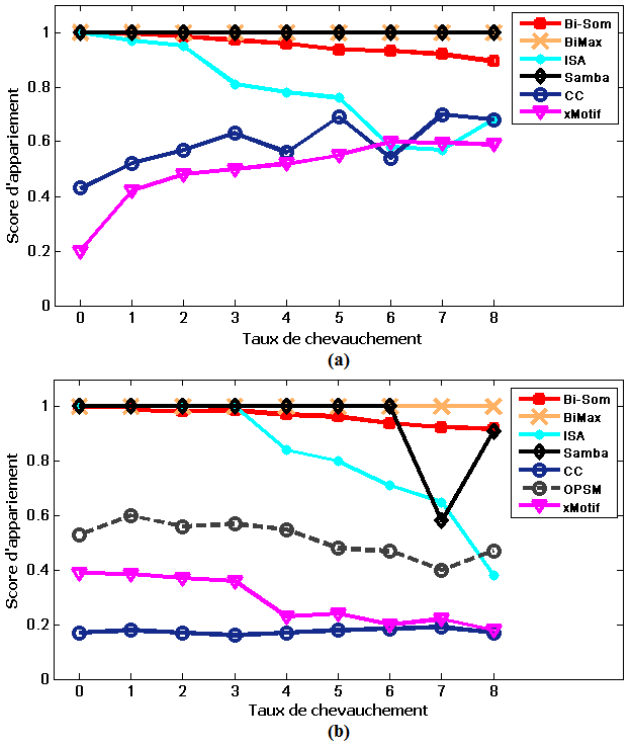


FIG. 10 – *Résultats de comparaison entre Bi-SOM et quelques méthodes de co-classification. (a) Bi-cluster constant (*Prelic3*). (b) Bi-cluster additif (*Prelic4*).*

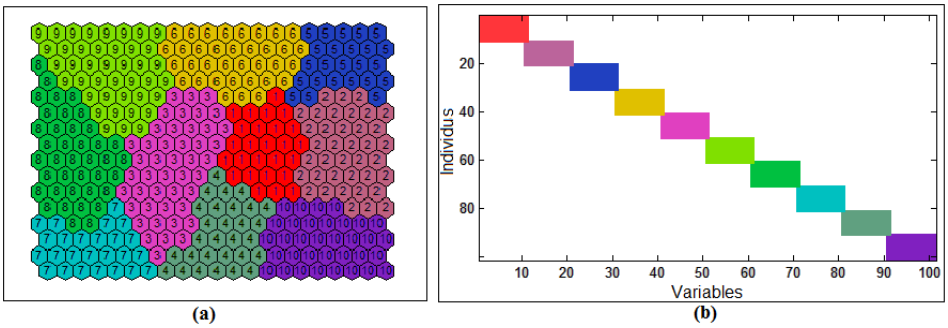


FIG. 11 – *Résultats de Bi-SOM appliquée sur (*Prelic4*). Avec un taux de chevauchement de* 10%. *(a) la carte segmentée. (b) Les blocs résultat.*

6 Conclusion

Dans cet article, nous avons proposé une nouvelle approche de co-classification automatique sur des données continues, basée sur les cartes topologiques de Kohonen que nous avons appelé Bi-SOM. La définition d'un lien associant à chaque individu la variable qui le caractérise nous a permis de classer simultanément les individus et les variables, et de maximiser ainsi l'homogénéité des blocs résultats. De plus nous avons bénéficié de la notion de voisinage offerte par SOM pour introduire une topologie de ces blocs. Nous avons ainsi exploité cette propriété de voisinage pour optimiser la partition obtenue en appliquant une classification hiérarchique sur les prototypes, avec l'application de l'indice de Davies-Bouldin pour déterminer le nombre optimal de blocs.

Bi-SOM nous a permis de réduire la dimension des données, en détectant et supprimant les variables non pertinentes afin d'améliorer le processus de co-classification et par conséquent, la qualité de la partition finale. Dans un premier temps, Bi-SOM a été appliquée sur des données réelles. L'évaluation de la partition obtenue en utilisant un indice interne (l'indice de Hubert) et des indices externes (Rand ajusté, Jaccard et le score d'appariement) par rapport à la partition correcte, confirme l'amélioration de la qualité de la partition après la suppression des variables non pertinentes. Dans une deuxième phase d'expérimentations, Bi-SOM a été appliquée sur des données synthétiques issues de la littérature. La comparaison de résultats obtenus avec les performances d'autres méthodes de co-classification a montré que notre approche est robuste au bruit et faiblement sensible aux chevauchements et reste compétitive avec des taux de performances très satisfaisants.

Références

Angiulli, F., E. Cesario, et C. Pizzuti (2008). Random walk biclustering for microarray data. *Information Sciences 178*, 1479–1497.

Bandyopadhyay, S., A. Mukhopadhyay, et U. Maulik (2007). An improved algorithm for clustering gene expression data. *Bioinformatics 21*, 2859–2865.

BenDor, A., B. Chor, R. Karp, et Z. Yakhini (2003). Discovering local structure in gene expression data: The order preserving sub matrix problem. *Journal of Computational Biology 10(3–4)*, 373–384.

Bergmann, S., J. Ihmels, et N. Barkai (2004). Defining transcription modules using large-scale gene expression. *Bioinformatics 1;20(13)*, 1993–2003.

Bryan, K., P. Cunningham, et N. Bolshakova (2005). Biclustering of expression data using simulated annealing. *CBMS 2005*, 383–388.

Busygin, S., G. Jacobsen, et E. Kramer (2002). Double conjugated clustering applied to leukemia microarray data.

Cheng, Y. et G. Church (2000). Biclustering of expression data. Volume 8, pp. 93–103.

Cottrell, M., S. Ibbou, et P. Letrémy (2004). Som-based algorithms for qualitative variables. *Neural Networks 17(8–9)*, 1149–1167.

Cottrell, M. et P. Letrémy (2005). How to use the kohonen algorithm to simultaneously analyze individuals and modalities in a survey. *Neurocomputing 63*, 193–207.

Davies, D. et D. Bouldin (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence 1*, 224–227.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 269–274.

Diday., E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes. la méthode des nuées dynamiques. *Statistiques Appliquées 2*, 19–33.

Dreyfus, G. (2002). *Réseaux de neurones-méthodologie et applications*. Paris : Eyrolls.

Eisen, M., P. Spellman, P. Brown, et D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. Volume 95(25), pp. 14863–14868.

Fort, J., M. Cottrel, et P. L. . (2001). Stochastic on-line algorithm versus batch algorithm for quantization and self-organizing maps. pp. 43–52.

Govaert, G. (1983). *Classification Croisée*. Thèse d'état, Université de Paris6.

Govaert, G. et M. Nadif (2009). Un modèle de mélange pour la classification croisée d'un tableau de données continue. *CAP 2009*, 81–106.

Hartigan, J. (1972). Direct clustering of data matrix. *Journal of the American Statistical Association, 67(337)*, 123–129.

Hartigan, J. (1975). Direct splitting. *Clustering Algorithms, John Wiley and Sons, New York Chap. 14*, 251–277.

Jaccard, P. (1901). Comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles 37*, 547–579.

Klugar, Y., R. Basri, J. Chang, et M. Gerstein (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research 13*, 703–716.

Madeira, S. et A. Oliveira (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1)*, 24–45.

Meeds, E. et S. Roweis (2007). Nonparametric bayesian bi-clustering. Technical report.

Mitra, S. et H. Banka (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition 39(12)*, 2464–2477.

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

Pensa, R. et J.-F. Boulicaut. (2008). Co-classification sous contraintes par la somme des résidus quadratiques. Volume RNTI-E-11, pp. 655–666.

Pensa, R., J.-F. Boulicaut, F. Cordero, et M. Atzori (2010). Co-clustering numerical data under user-defined constraints. *Statistical Analysis and Data Mining 3(1)*, 38–55.

Prelic, A., S. Bleuler, P. Zimmermann, A., Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, et E. Zitzler (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics 22(9)*, 1122–1131.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*, 846–850.

Schummer, M., W. Ng, R. Bumgarner, P. Nelson, B. Schummer, D. Bednarski, L. Hassell, R. Baldwin, B. Karlan, et L. Hood (1999). Comparative hybridization of an array of 21500 ovarian cdnas for the discovery of genes overexpressed in ovarian carcinomas. *Gene 238 (2)*, 375–385.

Shi, J. et J. Malik (2000). Normalized cuts and image segmentation. Technical report, University of California at Berkeley, Berkeley, CA, USA.

Tanay, A., R. Sharan, et R. Shamir (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics 18*, 36–44.

Thiria, S., Y. lechevalier, O. Gascuel, et S. Canu. (1997). *Statistiques et méthodes neuronales*. Paris : Dunod.

Xiaowen, L. et L. Wang (2007). Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics 23(1)*, 50–56.

Yang, J., W. Wang, H. Wang, et P. Yu (2003). Enhanced biclustering on expression data. BIBE '03, pp. 321–327.

Summary

In this paper, we present a new SOM based bi-clustering approach for continuous data matrices. This approach is called BiSOM (for Bi-clustering based on one Self-Organizing Map). Besides the main aim of bi-clustering which consists to simultaneously analyze the lines and columns of a given data matrix, we propose here to deal with some issues related to this task: (1) the topological visualization of bi-clusters with respect to their neighborhood relationship, (2) the optimization of these bi-clusters in macro-blocks and (3) the selection of relevant features by eliminating blocks of noises, iteratively. Finally, Experiments will be given over several databases for validating our approach in comparison with other bi-clustering approaches.