Caractérisation et extraction de biclusters de valeurs similaires avec l'analyse de concepts triadiques

Mehdi Kaytoue*, Sergei O. Kuznetsov**,
Amedeo Napoli***, Juraj Macko****, et Wagner Meira Jr.*

*Universidade Federal de Minas Gerais – Belo Horizonte – Brésil
kaytoue@dcc.ufmg.br (contact principal)
**Higher School of Economics – Moscou – Russie
***INRIA Nancy Grand Est/LORIA – Nancy – France
***Palacky University – Olomouc – République Tchèque

Résumé. Le biclustering de données numériques est devenu depuis le début des années 2000 une tâche importante d'analyse de données, particulièrement pour l'étude de données biologiques d'expression de gènes. Un bicluster représente une association forte entre un ensemble d'objets et un ensemble d'attributs dans une table de données numériques. Les biclusters de valeurs similaires peuvent être vus comme des sous-tables maximales de valeurs proches. Seules quelques méthodes se sont penchées sur une extraction complète (i.e. non heuristique), exacte et non redondante de tels motifs, qui reste toujours un problème difficile, tandis qu'aucun cadre théorique fort ne permet leur caractérisation. Dans le présent article, nous introduisons des liens importants avec l'analyse formelle de concepts. Plus particulièrement, nous montrons de manière originale que l'analyse de concepts triadiques (TCA) propose un cadre mathématique intéressant et puissant pour le biclustering de données numériques. De cette manière, les algorithmes existants de la TCA, qui s'appliquent habituellement à des données binaires, peuvent être utilisés (directement ou après quelques modifications) après un prétraitement des données pour l'extraction désirée.

1 Introduction

Le biclustering de données numériques a connu un essor considérable depuis le début des années 2000, apportant des réponses aux nouveaux challenges soulevés par l'analyse de données biologiques, et plus particulièrement l'analyse de données d'expression de gènes. A partir d'une table de données objets/attributs, le but du biclustering est de caractériser des sous-tables d'intérêt selon les valeurs qu'elles comportent (voir par exemple l'état de l'art de Madeira et Oliveira (2004)). Ainsi, un bicluster correspond à une paire, composée d'un sous-ensemble d'objets et d'un sous ensemble d'attributs. Une telle paire se représente comme un rectangle dans la table de données, modulo des permutations de colonnes et de lignes. La table 1 est un exemple de telles données numériques : chaque ligne représente un objet ; chaque colonne un attribut ; chaque case la valeur prise par l'objet en ligne pour l'attribut en colonne. Cette table illustre également le bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ représenté par un rectangle grisé.

Il existe de nombreux types de biclusters selon la relation entre les valeurs de la sous-table correspondante. Un cas trivial consiste à définir un bicluster comme une sous-table de valeurs égales. Puisque le nombre de biclusters est exponentiel, on est intéressé par des sous-tables maximales, par exemple $(\{g_1, g_2, g_3\}, \{m_5\})$. De tels biclusters *à valeurs constantes* n'apparaissent que dans des situations idylliques : les données numériques sont généralement bruitées. Ainsi, divers travaux se sont penchés sur la caractérisation et l'extraction de biclusters à valeurs similaires (voir les travaux de Madeira et Oliveira (2004); Besson et al. (2007); Kaytoue et al. (2011a) par exemple). Le rectangle de la table 1 illustre un tel bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ où deux valeurs sont dites similaires si leur différence n'excède pas 1. De plus, ce bicluster est maximal : on ne peut pas ajouter d'objet ou d'attribut sans violer la condition de similarité.

TAB. 1: Exemple de données numériques et d'un bicluster à valeurs similaires avec $\theta = 1$.

	m_1	m_2	m_3	m_4	m_5
g_1	**1**	**2**	**2**	1	6
g_2	**2**	**1**	**1**	0	6
g_3	**2**	**2**	**1**	7	6
g_4	8	9	2	6	7

Seules quelques méthodes abordent une extraction complète, correcte et non redondante de tels motifs (Besson et al. (2007); Kaytoue et al. (2011a)), connue pour être un problème difficile, tandis qu'aucun cadre mathématique n'est précisément défini. Dans cet article, nous montrons que l'analyse de concepts formels (ACF) introduite par Ganter et Wille (1999) définit déjà, dans un cadre mathématique clair, tous les outils nécessaires pour une telle tâche. L'ACF est connue pour être la base de l'extraction de motifs fermés dans les tables binaires, ces motifs pouvant être vus comme des rectangles maximaux de 1. Il est évident que de telles notions se retrouvent dans les données numériques pour l'extraction de biclusters de valeurs similaires. Le problème est de comprendre comment les notions de maximalité de biclusters et les opérateurs de fermeture en AFC se distinguent.

Dans cet article, nous montrons que l'analyse de concepts triadiques (ACT) introduite par Lehmann et Wille (1995) – l'extension de l'ACF aux données binaires et relations ternaires – ainsi que l'échelonnage conceptuel interordinal – une discrétisation particulière sans perte d'information– sont deux outils suffisants pour traiter l'extraction souhaitée. Ainsi, cet article dresse de manière originale de nouveaux liens entre ACF et biclustering. Pour cela, nous détaillons deux méthodes basées sur l'ACT et l'échelonnage. Ces méthodes s'accompagnent d'une meilleure interprétation et calculabilité des biclusters comparées aux approches existantes. Pour une lecture plus aisée, nous détaillons les avantages de notre approche au cours de l'article et en discussion.

Le papier est organisé comme suit. Tout d'abord les notions de base de l'ACT sont présentées dans la section 2. Ensuite, la section 3 pose formellement le problème d'extraction. Suivent alors nos deux méthodes, respectivement en section 4 et 5. Le papier se termine alors sur une série d'expérimentations et une discussion autour des perspectives de recherche.

2 Analyse de concepts triadiques

On rappelle les notions de l'analyse de concepts formels introduite par Ganter et Wille (1999) et de l'analyse de concepts triadiques (ACT) introduite par Lehmann et Wille (1995).

Analyse de concepts formels. Soient G un ensemble d'objets, M un ensemble d'attributs et $I \subseteq G \times M$ une relation binaire. (G, M, I) est appelé *contexte formel* ou *contexte dyadique*

et $(g, m) \in I$ signifie que l'objet g "possède" l'attribut m. Un contexte se représente par une table binaire où une case contient une croix $\times$ si l'objet en ligne possède l'attribut en colonne. Les deux opérateurs de dérivation $(.)'$ définis par $A' = \{m \in M \mid \forall g \in A \subseteq G : (g, m) \in I\}$ et $B' = \{g \in G \mid \forall m \in B \subseteq M : (g, m) \in I\}$ définissent une *connexion de Galois* entre l'ensemble des parties de G et l'ensemble des parties de M. Un couple (A, B) avec $A' = B$ et $B' = A$ est un *concept formel* ou *concept dyadique*. A et B sont respectivement appelés l'*extension* et l'*intension* du concept. Un concept se voit comme un rectangle maximal de croix dans la table binaire.

Analyse de concepts triadiques. Cette extension su'utilise lorsqu'un objet possède un attribut dans certaines situations seulement, par exemple à certains temps d'une cinétique. Les données sont représentées alors par un contexte triadique noté (G, M, B, Y). G, M, et B sont respectivement les ensemble d'objets, d'attributs et de conditions, et $Y \subseteq G \times M \times B$. $(g, m, b) \in Y$ signifie que "L'objet g possède l'attribut m sous la condition b". Un concept triadique de (G, M, B, Y) est un triplet (A_1, A_2, A_3) avec $A_1 \subseteq G$, $A_2 \subseteq M$ et $A_3 \subseteq B$ satisfaisant : (i) $A_1 \times A_2 \times A_3 \subseteq Y$, et (ii) avec $X_1 \times X_2 \times X_3 \subseteq Y$, $A_1 \subseteq X_1$, $A_2 \subseteq X_2$ et $A_3 \subseteq X_3$ implique $A_1 = X_1$, $A_2 = X_2$ et $A_3 = X_3$. (G, M, B, Y) est représenté par une table à 3 dimensions, et (i) signifie qu'un tel concept est un cube rempli de croix, tandis que (ii) caractérise la maximalité. Pour un concept triadique (A_1, A_2, A_3), A_1 est l'extension, A_2 l'intension et A_3 le modus. Pour décrire les opérateurs de dérivations, il est plus aisé de représenter un contexte triadique par (K_1, K_2, K_3, Y). Alors, pour $\{i, j, k\} = \{1, 2, 3\}$, $j < k$, $X \subseteq K_i$ et $Z \subseteq K_j \times K_k$, les opérateurs de dérivation $(.)^{(i)}$ sont définis par :

$\Phi : X \to X^{(i)} : \{(a_j, a_k) \in K_j \times K_k \mid (a_i, a_j, a_k) \in Y \text{ pour tout } a_i \in X\}$

$\Phi^{'} : Z \to Z^{(i)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ pour tout } (a_j, a_k) \in Z\}$

Cette définition amène l'opérateur de dérivation $\mathbf{K}^{(3)}$ et le contexte dyadique $\mathbf{K}^{(3)} = \langle K_3, K_1 \times K_2, Y^{(3)} \rangle$. De plus, avec $\{i, j, k\} = \{1, 2, 3\}$, $X_i \subseteq K_i$, $X_j \subseteq K_j$ et $A_k \subseteq K_k$, les opérateurs de dérivation $(.)^{(i,j,A_k)}$ sont donnés par :

$\Psi : X_i \to X_i^{(i,j,A_k)} : \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_i, a_k) \in X_i \times A_k\}$

$\Psi^{'} : X_j \to X_j^{(i,j,A_k)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ pour tout } (a_j, a_k) \in X_j \times A_k\}$

Les opérateurs Φ et $\Phi^{'}$ sont qualifiés d'extérieurs, et la paire qu'ils forment d'opérateur de fermeture extérieur. De manière duale, Ψ and $\Psi^{'}$ sont qualifiés d'intérieurs. Les opérateurs de dérivations du contexte dyadique sont définis par $\mathbf{K}_{A_k}^{ij} = \langle K_i, K_j, Y_{A_k}^{ij} \rangle$, où $(a_i, a_j) \in Y_{A_k}^{ij}$ ssi a_i, a_j, a_k sont en relation Y for all $a_k \in A_k$.

3 Problème et notations

Données. En ACF, une table de données numériques se définit comme un *contexte multi-valué* (G, M, W, I) où G est un ensemble d'objets, M un ensemble d'attributs à valeurs numériques, W un ensemble de valeurs, et $I \subseteq G \times M \times W$ une relation ternaire telle que $(g, m, w) \in I$, aussi écrit $m(g) = w$, se lit : "L'objet g prend la valeur w pour l'attribut m". Dans la table 1, on a $G = \{g_1, g_2, g_3, g_4\}$, $M = \{m_1, m_2, m_3, m_4, m_5\}$, $W = \{0, 1, 2, 6, 7, 8, 9\}$ et $m_5(g_2) = 6$. Un bicluster est alors simplement une paire (A, B) avec $A \subseteq G$ et $B \subseteq M$.

Biclusters et similarité. Soient deux valeurs $w_1, w_2 \in W$ et un paramètre dit de similarité $\theta \in \mathbb{R}$, on dit que w_1 est similaire à w_2 ssi $|w_1 - w_2| \leq \theta$ et l'on note $w_1 \simeq_\theta w_2$. Alors, (A, B) est un bicluster de valeurs similaires (BVS) ssi $m(g) \simeq_\theta n(h)$ pour tout $g, h \in A$ et pour tout

$m, n \in B$. Un BVS est maximal (BMVS) ssi l'ajout soit d'un objet dans A, soit d'un attribut dans B ne résulte pas en un BVS. Dans l'exemple de la table 1, $(\{g_1, g_2\}, \{m_2\})$ est un BVS avec $\theta \geq 1$. Cependant, il n'est pas maximal. Avec $1 \leq \theta < 5$, $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ est maximal. Pour $\theta = 7$, $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3, m_4, m_5\})$ est maximal.
Problème. Le problème traité dans cet article est l'extraction de tous les BMVS d'un jeu de données numériques. Nous souhaitons une extraction complète, correcte et non-redondante par opposition aux méthodes heuristiques (Madeira et Oliveira (2004)). Ce travail fait réponse à l'article de Besson et al. (2007). Pour cela, nous proposons dans la prochaine section une première méthode ayant pour but d'extraire tous les BMVS pour n'importe quel paramètre θ. Cette méthode établit de nouveaux liens entre le biclustering et l'ACF en général, et l'ACT en particulier. La méthodologie est alors adaptée pour caractériser et extraire les biclusters maximaux pour un certain paramètre θ défini par l'utilisateur, comme cela se fait habituellement.

4 BMVS dans l'analyse de concepts triadiques

Nous considérons le problème de la génération de tous les BMVS pour n'importe quel paramètre θ. A partir d'un jeu de données (G, M, W, I), l'idée est de construire un contexte triadique (G, M, T, Y) où les deux premières dimensions restent respectivement les objets et les attributs du jeu de données, tandis W est échelonné (discrétisé) en une dimension T. Cette nouvelle dimension T est appelée la *dimension d'échelonnage* : intuitivement cet ensemble donne les différents espaces de valeurs que chaque bicluster peut prendre. Pour construire l'ensemble T, nous utilisons l'échelonnage interordinal introduit par Ganter et Wille (1999). Il permet d'encoder dans 2^T tous les intervalles de valeurs possibles de W. Cette échelle permet de dériver un contexte triadique dont les concepts correspondent exactement aux BMVS.

Echelonnage interordinal. Une échelle est une relation binaire $J \subseteq W \times T$ qui associe les éléments originaux de W à leur éléments dérivés de T. Pour l'échelonnage interordinal, on a $T = \{[min(W), w], \forall w \in W\} \cup \{[w, max(W)], \forall w \in W\}$. Ainsi, $(w, t) \in J$ ssi $w \in t$.

La table 2 donne la représentation tabulaire de l'échelle interordinale pour la table numérique 1. Intuitivement, chaque ligne décrit une valeur, tandis que les concepts formels – i.e. des rectangles maximaux de croix – représentent tous les intervalles de valeurs possibles de W. Un exemple de concept dyadique est $(\{6, 7, 8\}, \{t_6, t_7, t_8, t_9, t_{10}\})$, ré-écrit $(\{6, 7, 8\}, \{[6, 8]\})$ puisque $\{t_6, t_7, t_8, t_9, t_{10}\}$ représente l'intervalle $[0, 8] \cap [0, 9] \cap [1, 9] \cap [2, 9] \cap [6, 9] = [6, 8]$.

Contexte triadique dérivé. Considérons la relation ternaire $Y \subseteq G \times M \times T$. Alors $(g, m, t) \in Y$ ssi $(m(g), t) \in J$, ou simplement $m(g) \in t$. On appelle (G, M, T, Y) le contexte tradique dérivé du jeu de données numériques (G, M, W, I).

La paire (g_1, m_1), prenant la valeur $m_1(g_1) = 1$, permet de dériver les triplets $(g_1, m_1, t) \in Y$ où t est un intervalle dans $\{[0, 1], [0, 2], [0, 6], [0, 7], [0, 8], [0, 9], [1, 9]\}$. L'intersection des intervalles de cet ensemble est la valeur originale elle même, c-à-d $m_1(g_1) = 1$, une propriété basique de l'échelonnage interordinal. Finalement, la table 3 représente le contexte triadique dérivé dans son intégralité. La toute première croix de cette table (en haut à gauche) représente le tuple (g_2, m_4, t_1), ce qui signifie que $m_4(g_2) \in [0, 0]$.

J	$t_1 = [0,0]$	$t_2 = [0,1]$	$t_3 = [0,2]$	$t_4 = [0,6]$	$t_5 = [0,7]$	$t_6 = [0,8]$	$t_7 = [0,9]$	$t_8 = [1,9]$	$t_9 = [2,9]$	$t_{10} = [6,9]$	$t_{11} = [7,9]$	$t_{12} = [8,9]$	$t_{13} = [9,9]$
0	×	×	×	×	×	×	×						
1		×	×	×	×	×	×	×					
2			×	×	×	×	×	×	×				
6				×	×	×	×	×	×	×			
7					×	×	×	×	×	×	×		
8						×	×	×	×	×	×	×	
9							×	×	×	×	×	×	×

TAB. 2: Échelle interordinale de l'ensemble de valeurs W.

	$t_1 = [0,0]$					$t_2 = [0,1]$					$t_3 = [0,2]$					$t_4 = [0,6]$					$t_5 = [0,7]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1						×			×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_2				×			×	×	×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_3								×			×	×	×			×	×	×		×	×	×	×	×	×
g_4													×					×	×				×	×	×

	$t_6 = [0,8]$					$t_7 = [0,9]$					$t_8 = [1,9]$					$t_9 = [2,9]$					$t_{10} = [6,9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×		×					×
g_2	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×				×					×
g_3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×				×	×
g_4	×		×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×

	$t_{11} = [7,9]$					$t_{12} = [8,9]$					$t_{13} = [9,9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1															
g_2															
g_3				×											
g_4	×	×			×	×	×					×			

TAB. 3: Contexte triadique dérivé de la table 1 avec l'échelle interordinale.

Nous présentons maintenant notre premier résultat principal : il y a une bijection entre (i) l'ensemble de tous les BMVS de (G, M, W, I) pour n'importe quel paramètre de similarité θ et l'ensemble des concepts triadiques de (G, M, T, Y).

Proposition 1. (A, B, U), avec $A \subseteq G$, $B \subseteq G$ et $U \subseteq T$ est un concept triadique ssi (A, B) est un BMVS pour un $\theta \geq 0$ (et vice-versa).

Preuve. La preuve de cette proposition est donnée en annexe afin de faciliter la lecture.

Par exemple, $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \{t_3, t_4, t_5, t_6, t_7, t_8\})$ est un concept triadique du contexte représenté par la table 3. Il correspond au BMSV $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ avec $\theta = 1$. $\theta = 1$ car $\{t_3, t_4, t_5, t_6, t_7, t_8\}$ est maximal (c'est un modus) qui correspond à l'intervalle $[1, 2]$, et $2 - 1 = 1$ est la longueur de cet intervalle.

Nous avons montré que l'extraction de BMVS requiert d'échelonner les données numériques en un contexte triadique, puis d'en extraire les concepts. L'extraction des BMVS pour n'importe quel θ peut ne pas être efficace et générer un nombre non analysable de motifs (ce qui reste un problème majeur dans le domaine de l'extraction de motifs en général). Remar-

	label 1					label 2					label 3					label 4					label 5				
	[0, 1]					[1, 2]					[6, 7]					[7, 8]					[8, 9]				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1	×			×		×	×	×	×					×											
g_2		×	×	×		×	×	×						×											
g_3			×			×	×	×					×	×				×							
g_4								×					×	×		×				×	×	×			

TAB. 4: Contexte triadique dérivé à l'aide des blocs de tolérance sur W et $\theta = 1$.

quons que les concepts (A, B, U) avec les ensembles A, B et C les plus grands représentent de grands biclusters de valeurs proches : plus $|A|$ et $|B|$ sont grands, plus la couverture du bicluster est importante ; plus $|U|$ est grand, plus les valeurs du bicluster (A, B) sont proches par les propriétés de l'échelonnage interordinal. Ainsi, l'extraction des k-meilleurs BMVS peut être considérée avec des algorithmes existants comme DATA-PEELER (Cerf et al. (2009)).

5 Extraction de BMVS pour un θ donné

Nous considérons le problème d'extraction des BMVS avec l'ACT pour un θ donné. Intuitivement, considérons l'échelle appliquée à W du jeu de données numériques (G, M, W, I). Cette échelle permet de transformer W en l'ensemble T et les sous-ensembles de T caractérisent tous les intervalles de valeurs possibles de W. Pour obtenir les BMVS pour un θ donné seulement, nous n'avons pas besoin de considérer tous les intervalles de valeurs possibles, mais seulement des intervalles maximaux de valeurs similaires. Il s'agit là de *blocs de tolérance*. Une fois ces intervalles obtenus, le processus d'échelonnage reste le même et le contexte triadique résultant est traité avec un nouvel algorithme nommé TRIMAX.

Notons d'abord que $\simeq_\theta$ est une relation de tolérance, c-à-d réflexive, symétrique mais non transitive. Un bloc de tolérance de W est défini comme un ensemble maximal de valeurs similaires deux à deux. Dans la table 1, nous avons $W = \{0, 1, 2, 6, 7, 8, 9\}$, et avec $\theta = 2$, on obtient 3 blocs de tolérance, soient $\{0, 1, 2\}$, $\{6, 7, 8\}$ et $\{7, 8, 9\}$. Ces trois ensembles peuvent être renommés par leur enveloppe convexe sur $\mathbb{R}$: respectivement $[0, 2]$, $[6, 8]$ et $[7, 9]$. En effet, n'importe quel nombre compris entre les valeurs minimale et maximale d'un bloc de tolérance est similaire à toutes les autres valeurs du bloc.

Pour dériver le contexte triadique nous utilisons l'ensemble des blocs de tolérance renommés par leur enveloppe convexe, noté C et appliquons la même procédure que précédemment. Continuons avec notre exemple, nous avons : $C = \{[0, 1], [1, 2], [6, 7], [7, 8], [8, 9]\}$ avec $\theta = 1$. La table 4 illustre le contexte triadique dérivé avec cette nouvelle échelle et $\theta = 1$.

Contexte dyadique associé à un bloc de tolérance. Considérons un bloc de tolérance $c \in C$. Le contexte dyadique associé à ce bloc est donné par (G, M, Z) où $z \in Z$ dénote les $(g, m) \in G \times M$ tels que $m(g) \in c$.

Remarquons aussi que les blocs de tolérance sur W sont totalement ordonnés : soient $[v_1, v_2]$ et $[w_1, w_2]$ deux blocs, on a naturellement $[v_1, v_2] < [w_1, w_2]$ ssi $v_1 < w_1$. Aussi, les contextes dyadiques associés sont totalement ordonnés et nous utilisons dans la suite un ensemble d'indices pour les caractériser. Dans la table 4, les contextes pour les blocs $\langle [0, 1], [1, 2], [6, 7], [7, 8], [8, 9] \rangle$ ont pour labels respectifs $\langle 1, 2, 3, 4, 5 \rangle$.

Nous pouvons maintenant présenter notre deuxième résultat : le nouveau contexte triadique dérivé permet l'extraction des BMVS pour le seuil θ donné. Dans ce cas cependant, les algorithmes existants de l'ACT ne peuvent pas être appliqués directement. Par exemple, dans la table 4, le concept triadique $(\{g_3\}, \{m_4\}, \{3, 4\})$ correspond à un BVS qui n'est pas maximal. Nous détaillons alors le nouvel algorithme TRIMAX, ancré dans l'ACT et utilisant les opérateur de dérivation pour un calcul efficace des BMVS.

L'idée basique de TRIMAX repose sur les faits suivants. Tout d'abord, puisque chaque contexte dyadique est associé à un bloc de tolérance, nous n'avons pas besoin de considérer les intersections de contextes (ou des intervalles associés), comme c'est le cas avec l'ACT classique. De cette manière, chaque contexte dyadique est fouillé séparément. Cela assure que les concepts dyadiques résultant correspondent à des BVS, mais n'assure pas la maximalité (voir exemple précédent). Nous devons alors vérifier si un concept dyadique est toujours un concept dans d'autres contextes, c-à-d calculer son modus. La proposition suivante formalise ces faits.

Proposition 2. Soit (A, B, U) un concept triadique du contexte (G, M, C, Y), t.q. U est la fermeture extérieure du singleton $\{c\} \subseteq C$. Si $|U| = 1$, (A, B) est un BMVS. Sinon, (A, B) est un BMVS ssi $\nexists y \in [min(U); max(U)]$, $y < c$ t.q. $(A, B) \neq \Psi_y^{'}(\Psi_y((A, B)))$, où $\Psi_y^{'}(.)$ et $\Psi_y(.)$ correspondent aux opérateurs de dérivations internes associés au y^{eme} contexte dyadique.

Preuve. Si $|U| = 1$, (A, B) est un concept dyadique seulement dans un contexte correspondant à un bloc de tolérance. Par les propriétés de ces derniers, (A, B) est un BMVS. Si $|U| \neq 1$, (A, B) est un concept dyadique dans $|U|$ contextes dyadiques. Puisque les blocs de tolérance sont totalement ordonnés, cela implique directement que le modus U est un intervalle $[min(U); max(U)]$. Ainsi, si $\exists y \in [min(U); max(U)]$ t.q. $(A, B) = \Psi_y^{'}(\Psi_y((A, B)))$, cela signifie que (A, B) n'est pas un BMVS.

Description de l'algorithme TRIMAX**.** TRIMAX commence par échelonner les données numériques pour produire plusieurs contextes dyadiques (un par bloc de tolérance de W pour le θ donné). L'ensemble de ces contextes dyadiques forme naturellement un contexte triadique. Alors, chaque contexte dyadique est fouillé séparément avec n'importe quel algorithme classique d'AFC (ou algorithme d'extraction de motifs fermés), et tous les concepts dyadiques sont extraits. Pour chaque concept dyadique (A, B), on calcule son modus $\Phi^{'}((A, B))$, c-à-d pour obtenir l'ensemble des contextes dyadiques dans lequel le concept apparait. Si l'on obtient un singleton, (A, B) est un BMVS, n'a pas, et ne sera plus généré une autre fois. Sinon, (A, B) est un concept dyadique dans d'autres contextes, et peut alors avoir été généré plusieurs fois. De fait, (A, B) n'est considéré que si l'on est sûr qu'il est généré pour la dernière fois. Ce choix arbitraire est possible grâce à l'ordre total des blocs de tolérance. Finalement, nous devons encore vérifier que le concept dyadique correspond bien à un bicluster maximal, c-à-d qu'il n'existe pas un contexte dans son modus où (A, B) n'est pas un concept (n'est pas maximal).

Proposition 3. TRIMAX produit une collection (i) complète, (ii) correcte et (iii) non redondante des BMVS pour un jeu de données numériques et un seuil de similarité donnés.

Preuve. (i) et (ii) découlent directement de la proposition précédente. L'affirmation (iii) est assurée par la deuxième condition *if* de l'algorithme : un concept dyadique n'est considéré que s'il est extrait à partir du dernier contexte dyadique dans lequel il apparait.

Algorithme 1 : Pseudo-code de l'algorithme TRIMAX

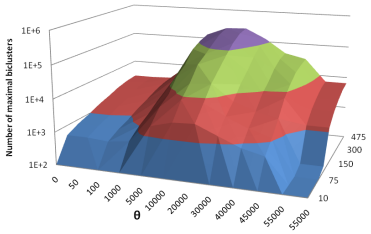
input : Données numériques (G, M, W, I), seuil de similarité θ
output : Tous les BMVS de (G, M, W, I)

Soit $C = \{[a_i, b_i]\}$ l'ensemble totalement ordonné des blocs de tolérance sur W pour un θ. Les i forment un ensemble d'indices.
forall $[a_i, b_i] \in C$ **do**
  Construire (G, M, Z_i) t.q. $(g, m) \in Z_i \Leftrightarrow m(g) \in [a_i, b_i]$
forall (G, M, Z_i) **do**
  Utiliser un algorithme d'AFC pour extraire ses concepts dyadiques (A, B)
  forall *concept dyadique* (A, B) *dans le contexte* (G, M, Z_i) *courant* **do**
    if $|\Phi^{'}((A, B))| = 1$ **then**
      Afficher (A, B)
    else if $max(\Phi^{'}((A, B)) = i$ **then**
      $x \leftarrow min(\Phi^{'}((A, B))$
      if $\nexists y \in [x, i[$ *s.t.* $(A, B) \neq \Psi^{'}_{y}(\Psi_y((A, B)))$ **then**
        Afficher (A, B)

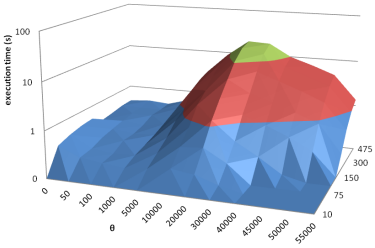
6 Expériences

Données. Nous explorons un jeu de données d'expression de gènes de l'espèce *Laccaria bicolor* disponible au NCBI [1]. Ces données mesurent l'expression de $11,930$ gènes dans 12 situations biologiques. Les valeurs d'expression varient entre 0 et 60000.
Implémentation de TRIMAX. TRIMAX est écrit en C++. Il utilise les structures de données de la librairie BOOST et l'implémentation originale de l'algorithme INCLOSE [2] pour l'extraction de concepts dyadiques. A chaque itération de la boucle principale, c-à-d pour chaque bloc de tolérance, le contexte dyadique courant est produit : nous ne générons pas le contexte triadique dans son intégralité par souci de consommation mémoire. De fait, le calcul du modus d'un concept dyadique nécessite de réaliser un échelonnage "à la volée" puisque l'on a pas accès aux autres contextes dérivés. Les expériences ont été réalisées avec un processeur Intel CPU 2.54 Ghz et 8 GO de mémoire vive sous Ubuntu 11.04.
Expériences. Le but des expériences n'est pas de donner une évaluation qualitative de l'approche présentée, mais plutôt de donner une évaluation quantitative en termes de temps d'exécution. En effet, ce travail vise à montrer comment un type de bicluster existant peut être considéré au sein de l'AFC. Pour une interprétation biologique, le lecteur peut se tourner vers les articles de Besson et al. (2007); Kaytoue et al. (2011b). Pour la plupart des expériences, le jeu de données utilisé comporte un nombre croissant d'objets et les 12 attributs. Les objets sont choisis aléatoirement une fois pour toutes afin de pouvoir comparer les différents résultats. Nous faisons également varier le paramètre θ de la même manière dans toutes les expériences. Nous nous intéressons aux aspects suivants : (i) le nombre de BMVS ; (ii) le temps d'exécution, (iii) le nombre de blocs de tolérance, (iv) la densité du contexte triadique correspondant

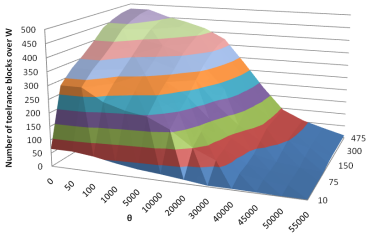
1. http ://www.ncbi.nlm.nih.gov/geo/ : série GSE9784 avec de plus amples détails
2. http ://sourceforge.net/projects/inclose



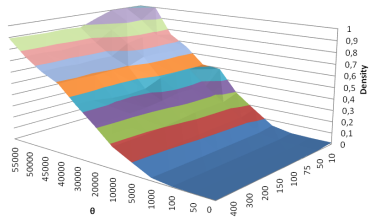
(i)Nombre de BMVS (axe Y) selon θ (axe X) et $|G|$ (axe Z)



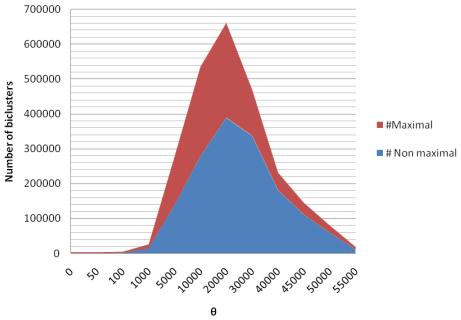
(ii) Temps d'exécution (sec) (axe Y) selon θ (axe X) and $|G|$ (axe Z)



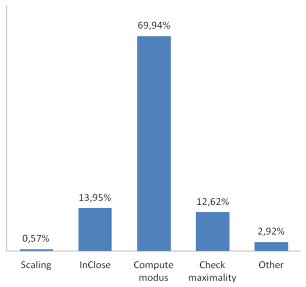
(iii) Nombre de blocs de tolérance (axe Y) selon θ (axe X) et $|G|$ (axe Z)



(iv) Densité du contexte triadique (axe Y) selon θ (axe X) et $|G|$ (axe Z)



(v) Comparaison du nombre de candidats par rapport au nombre de BMVS en variant θ avec $|G| = 500$



(vi) Répartition du temps d'exécution des procédures principales de TRIMAX avec $\theta = 33000$ et $|G| = 500$

FIG. 1: Expériences

$d(G, M, C, Y) = |Y|/(|G| \times |M| \times |C|)$ (un facteur important pour le calcul de concepts comme montré par Kuznetsov et Obiedkov (2002)), (v) le nombre de candidats (concepts dyadiques) comparé au nombre de BMVS, et finalement (vi) le temps d'exécution des diverses procédures principales de l'algorithme TRIMAX, à l'aide de l'outil GNU GPROF.

Résultats. La figure 1 donne les résultats de nos expériences. Une première observation du graphique (i) est la suivante : le nombre de biclusters est le plus important pour une valeur de θ autour de 30000. Une première explication est que 30000 est la médiane des valeurs de W qui contient presque tous les multiples de 100 de l'intervalle $[0; 60000]$. Les temps d'exécution du graphique (ii) montrent le même comportement. Ces résultats se comprennent à l'aide des graphiques (iii) et (iv). Dans (iii) le nombre de blocs de tolérance maximal est atteint quand $\theta = 0$, et au contraire, quand $\theta = max(W)$ alors $|C| = 1$. Maintenant observons le graphique (iv). La densité suit un comportement inverse : quand $\theta = 0$, la densité tend vers 0% ; quand $\theta = max(W)$, la densité vaut exactement 100%. Ces deux attitudes duales du nombre de blocs et de la densité expliquent le pire cas en terme de nombre de biclusters et de temps d'exécution, cette explication n'étant pas connue avec les approches précédentes d'extraction de BMVS. Une autre explication se découvre à l'aide du graphique (v) et est propre à notre méthode. Sont comparés sur cette figure le nombre de candidats et le nombre de BMVS. Là-aussi, le pire cas est atteint quand $\theta \simeq 30000$. Observons maintenant le graphique (vi) : la procédure la plus couteuse est le calcul du modus. L'explication est que les modus sont calculés avec un échelonnage à la volée. C'est là le point faible de l'algorithme. En pratique cependant, un analyste n'est pas intéressé par l'ensemble des biclusters, et définit des contraintes comme des tailles minimales ou maximales sur le nombre d'objets, d'attributs, etc. C'est là qu'est la force de TRIMAX : nombre de ces contraintes peuvent être évaluées sur les candidats avant de calculer leur modus. Ces expériences ne sont pas détaillés ici, mais les résultats sont prometteurs. Par exemple avec $\theta = 33000$, $|G| = 500$, et la contrainte $10 \leq A \leq 40$, TRIMAX ne produit que 5332 BMVS en 2 secondes comparés aux 104226 biclusters extraits en 16 secondes sans contraintes. Cet aspect modulaire est intéressant pour prendre en compte de nouvelles contraintes selon la nature de l'algorithme d'extraction de concepts utilisé au cœur de TRIMAX.

Comparaison avec les méthodes existantes. Deux méthodes existantes de la littérature considèrent le même problème : l'algorithme NBS-MINER de Besson et al. (2007) et l'algorithme IPS de Kaytoue et al. (2011a) basé sur les structures de patrons intervalles en AFC. Par manque de place, nous ne détaillons pas ces méthodes. NBS-MINER et IPS ont étés implémentés en C++ également comme décrits dans leur article respectif. Il s'avère que NBS-MINER ne passe pas à l'échelle comme le laissent entendre ses auteurs. IPS montre un meilleur passage à l'échelle, mais est largement devancé par TRIMAX quand le nombre d'objets croit, par exemple au dessus de 500 avec les données de ces expériences. Le problème de IPS est qu'il doit calculer des blocs de tolérance pour chaque motif candidat alors que TRIMAX ne fait cette tâche qu'une seule fois. La limite de la comparaison entre IPS et TRIMAX est le manque d'un algorithme efficace pour le calcul de blocs de tolérance sur un ensemble d'intervalles, et non sur un ensemble de nombres comme pour TRIMAX qui est une tâche bien plus aisée.

7 Conclusion

Nous nous sommes intéressés au biclustering de données numériques avec l'ACF, et avons montré comment les BMVS peuvent être caractérisés et extraits avec l'ACT qui se révèle être un cadre formel riche et associé à des outils existants et performants. Nous avons éclairé quelques liens entre maximalité des BMVS et opérateurs de dérivations en ACT. Après avoir "augmenté" les données d'une dimension, un bicluster est représenté par un concept triadique : son extension et son intension donnent sa couverture dans les données ; son modus donne son espace de valeurs. Ces trois ensembles sont maximaux. De plus, la notion très connue de concept fréquent prend un sémantique nouvelle au regard de la similarité. Par exemple, soit (A, B, C) un concept triadique et (A, B) son bicluster correspondant : plus $|C|$ est élevé, plus les valeurs du bicluster sont proches. La méthode peut aussi être adaptée aux jeux de données numériques n-aires : la fouille de BMVS à n dimensions peut être réalisée avec l'analyse de concepts $n + 1$-adiques, pour lequel il existe un algorithme (DATA-PEELER). Nous nous sommes aussi intéressés au problème plus classique de l'extraction de BMVS avec un θ donné, et TRIMAX se révèle être meilleur que ses concurrents. Cet algorithme est totalement modulaire : divers algorithmes d'extraction de concepts dyadiques peuvent être utilisés en son cœur. De nombreuses contraintes sur les biclusters peuvent être évaluées avant le calcul du modus à l'aide d'algorithmes d'extraction de motifs fermés existants. TRIMAX peut être distribué aisément sur plusieurs cœurs de calculs puisque que chacune de ses itérations est indépendante. Ce sont là nos perspectives de recherche : (i) le calcul des k-meilleurs BMVS, (ii) l'extension aux données multi-dimensionnelles, et enfin (iii) la parallélisation des calculs.

Références

Besson, J., C. Robardet, L. D. Raedt, et J.-F. Boulicaut (2007). Mining bi-sets in numerical data. In S. Dzeroski et J. Struyf (Eds.), *KDID*, Volume 4747 of *Lecture Notes in Computer Science*, pp. 11–23. Springer.

Cerf, L., J. Besson, C. Robardet, et J.-F. Boulicaut (2009). Closed patterns meet *n*-ary relations. *TKDD 3*(1).

Ganter, B. et R. Wille (1999). *Formal Concept Analysis*. Springer.

Kaytoue, M., S. O. Kuznetsov, et A. Napoli (2011a). Biclustering numerical data in formal concept analysis. In P. Valtchev et R. Jäschke (Eds.), *ICFCA*, Volume 6628 of *LNCS*, pp. 135–150. Springer.

Kaytoue, M., S. O. Kuznetsov, A. Napoli, et S. Duplessis (2011b). Mining gene expression data with pattern structures in formal concept analysis. *Inf. Sci. 181*(10), 1989–2001.

Kuznetsov, S. O. et S. A. Obiedkov (2002). Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell. 14*(2-3), 189–216.

Lehmann, F. et R. Wille (1995). A triadic approach to formal concept analysis. In *ICCS*, Volume 954 of *LNCS*, pp. 32–43. Springer.

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.

A Annexe : preuve de la proposition 1.

Nous devons tout d'abord introduire ce qui suit. Par soucis de simplicité, on considère W comme l'ensemble de tous les entiers compris dans le jeu de données numériques, c-à-d $W = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ avec notre exemple. Cela ne change en rien la preuve.

Valeur échelle et relation d'échelonnage. On appelle valeur d'échelle $s = q - r$ avec $r = min(W)$ et $q = max(W)$. La relation d'échelonnage est la relation binaire $J \subseteq W \times T$, où $T = \{t_1, \ldots, t_{2s+1}\}$ avec $r \leq w \leq q$ et $\langle w, t_i \rangle \in J$ ssi $i \in [w - r + 1, w - r + 1 + s]$. Il s'agit là d'une autre manière de définir l'échelonnage interordinal.

Base $E_{\theta w}$. On introduit $E_{\theta w} \subseteq T$ t.q. $E_{\theta w} = [t_{w+\theta-r+1}; t_{w-r+1+s}]$ pour un θ et $w \in W$ donné. Par exemple, $E_{12} = [t_{2+1-0+1}; t_{2-0+1+9}] = [t_4; t_{12}]$.

Proposition 1. $(w_b = m(g)) \simeq_\theta (n(h) = w_c)$ ssi $(\langle g, m \rangle \in Y^{12}_{E_{\theta b}}$ et $\langle h, n \rangle \in Y^{12}_{E_{\theta b}})$.

Preuve. Soient $E_b, E_c \subseteq T$ et $w_c \geq w_b$. Par définition $(g, m) \in Y^{12}_{E_{\theta b}}$ ssi m, g, t sont en relation Y pour tout $t \in E_{\theta b}$. On obtient alors $[t_{w_b-r+1}; t_{w_b-r+1+s}] = E_b \supseteq E_{\theta b} = [t_{w_b+\theta-r+1}; t_{w_b-r+1+s}]$ qui est direct. On doit alors montrer que $(h, n) \in Y^{12}_{E_{\theta b}}$ est vrai. On a alors $[t_{w_c-r+1}; t_{w_c-r+1+s}] = E_c \supseteq E_{\theta b} = [t_{w_b+\theta-r+1}; t_{w_b-r+1+s}]$ ssi $w_c - w_b \leq \theta$, qui est la définition de $\simeq_\theta$.

On peut alors déduire le corollaire suivant : $w_c - w_b \leq \theta$ ssi $E_b \cap E_c \supseteq E_{\theta b}$ et $w_c - w_b = \theta$ ssi $E_b \cap E_c = E_{\theta b}$. Nous pouvons alors prouver la proposition 1.

Reformulation de la Proposition 1. $\langle A_1, A_2, U \rangle$ avec $A_1 \subseteq G$, $A_2 \subseteq M$ et $U \subseteq T$ est un concept triadique ssi (A_1, A_2) est un BMVS pour un certain $\theta \geq 0$. De plus, $\theta = s - |U| + 1$.

Preuve. Soit $U = E_{\theta b}$ et le contexte dyadique $Y^{12}_U = Y^{12}_{E_{\theta b}}$ pour un certain w_b. A l'aide de l'opérateur de fermeture du contexte dyadique $\Psi'(\Psi((A_1))$ on obtient (A_1, A_2). De la définition d'un concept triadique, on sait que $A_1 \subseteq B_1$ implique $A_1 = B_1$ (cela est vrai pour A_2 également). De la définition d'un BMVS on sait que $\langle A_1, A_2 \rangle$ est maximal s'il n'existe pas $\langle B_1, B_2 \rangle$ t.q. $B_1 \supseteq A_1$ (cela tient aussi pour A_2). Il est alors clair que ces deux ensembles sont maximaux par définition et nous avons le même contexte dyadique $Y^{12}_U = Y^{12}_{E_{\theta b}}$. Intéressons nous au contexte dyadique $Y^{12}_U = Y^{12}_{E_{\theta b}}$. Avec $|U| = |E_{\theta b}| = |[t_{w_b+\theta-r+1}; t_{w_b-r+1+s}]|$ on observe que $|U| = s - \theta + 1$, qui donne $\theta = s - |U| + 1$. Finalement, U est maximal (c'est un modus) et $E_{\theta b}$ est maximal également puisque $w_c - w_b \leq \theta$ ssi $E_b \cap E_c \supseteq E_{\theta b}$. Les faits de cette preuve mènent à un bijection entre l'ensemble des BMVS et l'ensemble des concepts triadiques.

Summary

Biclustering numerical data became a popular data-mining task in the beginning of 2000's, especially for analysing gene expression data. A bicluster reflects a strong association between a subset of objects and a subset of attributes in a numerical object/attribute data-table. So called biclusters of similar values can be thought as maximal sub-tables with close values. Only few methods address a complete, correct and non redundant enumeration of such patterns, which is a well-known intractable problem, while no formal framework exists. In this paper, we introduce important links between biclustering and formal concept analysis. More specifically, we originally show that Triadic Concept Analysis (TCA), provides a nice mathematical framework for biclustering with a better algorithmic scalability over existing methods.