Découverte de biclusters avec présence ou absence de propriétés

Abdélilah Balamane*, Rokia Missaoui*
Léonard Kwuida**, Jean Vaillancourt***

*LARIM, Université du Québec en Outaouais, Québec, Canada
bala04@uqo.ca, rokia.missaoui@uqo.ca
**Haute École Spécialisée Bernoise (BFH), Suisse
leonard.kwuida@bfh.ch
***HEC Montréal, Québec, Canada
jean.vaillancourt@hec.ca

Résumé. La plupart des algorithmes de biclustering existants tiennent compte uniquement de la présence de propriétés que possède un ensemble d'objets. Cependant, il pourrait être fort utile dans plusieurs domaines d'application tels que le crime organisé, la génétique ou le marketing numérique d'identifier des groupes homogènes d'objets présentant des similarités tant au niveau de la présence que de l'absence d'attributs. Dans cet article, nous présentons une méthode générique de biclustering qui exploite une matrice binaire pour produire au moins trois types de biclusters : (i) ceux où toutes les valeurs sont égales à 1, (ii) ceux où toutes les valeurs sont égales à 0, et (iii) ceux indiquant la présence de certains attributs et/ou l'absence d'autres attributs sans nécessité de tenir compte du complémentaire du contexte (matrice) binaire de départ. L'implémentation et la validation de la méthode sur des collections de données permettent d'illustrer son potentiel de découverte de motifs pertinents.

1 Introduction

La fouille et la gestion de motifs font référence à un ensemble d'activités d'extraction, de stockage et de manipulation de motifs à partir des données. Un motif (*pattern*) est un résultat d'un processus de fouille de données exprimé sous forme de connaissance. En analyse formelle de concepts, le motif prend deux principales formes : a) des concepts formels décrivant des objets/individus avec leurs attributs communs et représentant des nœuds d'un treillis de concepts (Galois), et b) des règles d'association, y compris des implications, entre des groupes d'attributs.

Dans le présent article, nous proposons une nouvelle procédure de biclustering (classification croisée ou bipartitionnement) générique appelée BiP (***Bi****clustering* ***P****rocedure*) qui calcule des biclusters (blocs) sémantiquement significatifs à partir d'un graphe biparti issu d'une matrice d'adjacence à valeurs binaires. Différents types de biclusters peuvent alors être obtenus avec un contenu comportant : (i) des cellules remplies uniquement de 1 (type 1), (ii) des cellules remplies uniquement de 0 (type 2), et (iii) des colonnes complètement pleines de 1 et/ou

de 0 (type 3). Les biclusters des trois types correspondent à la notion de concepts en analyse formelle de concepts Ganter et Wille (1999) que nous rappelons ci-après. Ils sont stockés dans une structure d'arbre Patricia Morrison (1968) pour être retrouvés plus tard. L'utilisateur pourra ainsi extraire ceux qui contiennent un sous-ensemble d'objets et/ou d'attributs de la matrice d'adjacence. Notons aussi que la plupart des algorithmes de biclustering mettent l'accent sur uniquement la présence de propriétés que possèdent les objets ou les individus analysés. Cependant, il pourrait être fort utile dans plusieurs domaines d'application, tel que reconnu par plusieurs analystes depuis des décénnies, de montrer des groupes d'objets présentant des similarités tant au niveau de la présence que de l'absence (négation) d'attributs. Une étude récente Bala et al. (2017) montre l'intérêt de tenir compte de l'absence d'attributs en analyse de données. Une application à la détection de l'ambivalence au sein du cerveau est fournie afin de prédire la réponse d'une personne face à un choix. Mentionnons aussi une autre étude menée sur le cancer du sein Rodríguez-Jiménez et al. (2016) où les auteurs mettent en évidence le complément de connaissances que l'on peut obtenir lorsqu'on considère dans une analyse autant la présence d'attributs que leur absence. Certains résultats de cette étude ont révélé des associations entre des attributs qu'on n'aurait pas pu obtenir si on n'avait pas tenu compte de l'absence d'attributs. Aussi, cette étude a permis de mettre en évidence l'inutilité de certains examens médicaux que les oncologues avaient l'habitude de demander. Finalement, signalons qu'en marketing, si un client achète habituellement un ensemble de produits mais évite d'acquérir d'autres articles, ce motif peut être utilisé à des fins promotionnelles de produits.

1.1 Travaux connexes

En 1972, Hartigan Hartigan (1972) a proposé une nouvelle méthode de partitionnement appelée *biclustering* ou *coclustering*, qui regroupe simultanément des objets et des propriétés pour créer des sous-matrices appelées *biclusters*. Le principal avantage du biclustering est l'interprétation directe des blocs et la capacité d'identifier des corrélations entre des ensembles d'objets et des ensembles d'attributs principalement dans des matrices volumineuses et éparses. Il est à noter que le haut niveau de cohésion au sein des différents biclusters s'explique par le fait que seules les propriétés pertinentes à la création d'un ensemble d'objets sont utilisées plutôt que toutes les propriétés disponibles.

Plusieurs algorithmes de biclustering ont suivi la parution de l'article de Hartigan dans différentes disciplines et plus spécifiquement en bio-informatique Busygin et al. (2008); Charrad et al. (2008); Dhillon et al. (2003); Madeira et Oliveira (2004); Lewis et al. (2004), mais beaucoup moins en marketing, traitement d'image, ou analyse de réseaux sociaux. L'état de l'art présenté dans Madeira et Oliveira (2004) compare non seulement les caractéristiques du regroupement (*clustering*) avec celles du biclustering mais présente également un ensemble de méthodes de biclustering avec leurs particularités et le genre de biclusters qu'elles produisent.

Contrairement aux procédures de regroupement, les algorithmes de biclustering déterminent les groupes d'objets qui présentent des caractéristiques spécifiques pour un sous-ensemble donné de propriétés. Ainsi, ils sont considérés comme un bon choix pour l'analyse des données ayant des sous-ensembles d'objets (des gènes par exemple) dont les membres peuvent être décrits conjointement par un sous-ensemble d'attributs mais qui sont indépendants des autres Madeira et Oliveira (2004).

Les algorithmes de biclustering les plus connus utilisent une matrice binaire pour produire

des biclusters où toutes les valeurs sont égales à 1 Li et al. (2012). Chaque bloc généré couvre un ensemble d'objets et leurs attributs. L'un d'eux est Bimax et est connu pour être un algorithme dont la complexité temporelle est la plus basse Prelić et al. (2006). Bimax énumère tous les biclusters maximaux au sens de l'inclusion. En fait, les biclusters produits par Bimax ne sont rien d'autres que des concepts en analyse formelle de concepts Kaytoue et al. (2011). Nous croyons qu'un bicluster qui contient un arrangement approprié de 0 et 1 peut générer des motifs plus riches et plus pertinents qu'un bicluster composé seulement de 1. Ainsi, nous pouvons découvrir des objets ayant un ensemble de propriétés mais pas d'autres dans un bicluster donné, ce qui serait utile dans plusieurs domaines d'application. On peut par exemple identifier les espèces (humaines, animales ou végétales) qui ont un comportement spécifique face à un ensemble de conditions données et un comportement radicalement opposé lorsqu'elles sont exposées à d'autres conditions.

La plupart des méthodes de biclustering ont des paramètres d'entrée comme le nombre de biclusters à créer et/ou leur densité. La qualité des biclusters générés dépend alors de la valeur de ces paramètres. L'une de ces méthodes est la projection duale (*dual projection*) Everett et Borgatti (2013) qui produit des biclusters non chevauchants. L'approche est basée sur la projection d'un réseau à deux modes de données (graphe biparti) en deux réseaux à un mode de données. Les nœuds d'un même type sont connectés s'ils partagent des liens vers les mêmes nœuds du second type. Toutefois, le nombre de biclusters à produire doit être fixé à l'avance par l'utilisateur.

En analyse de réseaux sociaux, l'identification de groupes ayant des caractéristiques communes en utilisant le *blockmodeling* Beauguitte (2011), le biclustering Govaert et Nadif (2013) et les notions connexes telles que les équivalences structurelles ou régulières ont suscité moins d'intérêt de recherche que l'identification de groupes ou communautés cohésives. Tel que souligné par Knoke et Yang (2008), la connaissance de l'équivalence structurelle et des groupes concurrentiels conduit à la notion de substituabilité des acteurs d'un réseau social, ce qui signifie que si l'acteur *i* est structurellement équivalent à l'acteur *j*, alors le départ de l'un d'entre eux peut être compensé par la présence de l'autre sans altérer la structure du réseau d'origine.

Après un peu plus de quatre décennies de recherche sur les techniques de biclustering, nous pensons qu'il y a matière à amélioration par l'inclusion de la négation (absence) des propriétés des objets au sein des biclusters. C'est justement le cas des types 2, 3 que nous produisons à l'aide de la procédure BiP.

Au mieux de notre connaissance, il existe quelques méthodes qui produisent des biclusters combinant des valeurs 0 et 1 comme par exemple Everett et Borgatti (2013); Balamane (2017). Toutefois, la configuration de tels blocs ne vise pas nécessairement à faire ressortir tout aussi bien la présence que l'absence de propriétés.

Comme les trois instanciations de notre procédure générique permettent de calculer des concepts formels et utilisent un arbre Patricia pour les enregistrer, nous rappelons ci-après les principales notions de l'analyse formelle de concepts et celles concernant la structure de l'arbre Patricia et la gestion de ses éléments.

1.2 Analyse formelle de concepts

En analyse formelle de concepts Ganter et Wille (1999), un contexte formel est un triplet $\mathbb{K} := (G, M, I)$ où G, M et I représentent respectivement un ensemble d'objets, un ensemble

d'attributs et une relation binaire entre les éléments de G et M. Un concept formel est un couple (X, Y) tel que Y est l'ensemble de toutes les propriétés partagées par les objets de X, et X est l'ensemble de tous les objets qui ont les propriétés contenues dans Y. On note

$$X' := \{m \in M \mid (g, m) \in I \; \forall g \in X\} \quad \text{et} \quad Y' := \{g \in G \mid (g, m) \in I \; \forall m \in Y\}.$$

$(g, m) \in I$ signifie que l'objet g possède l'attribut m, et est aussi noté gIm. On dit que (X, Y) est un concept de $\mathbb{K}$ si et seulement si $X' = Y$ et $Y' = X$. L'extension du concept $c := (X, Y)$ est X (et notée ext(c)) et Y en est l'intention (et notée int(c)). Un sous-ensemble X est dit fermé si $X'' = X$. Le treillis de concepts (Galois) du contexte formel $\mathbb{K}$ est un treillis résultant de l'ordre partiel entre les concepts, défini par :

$$(U, V) \leq (X, Y) : \Longleftrightarrow \; U \subseteq X \quad (\text{ équivalent à } \; Y \subseteq V).$$

Si $(U, V) \leq (X, Y)$, on dit que le concept (U, V) est un *sous-concept* (ou prédécesseur) de (X, Y). Nous écrirons $g \not I \; m$ pour signifier que l'objet g ne possède pas l'attribut m. Nous désignons par $\bar{m}$ la négation de l'attribut m dans $\mathbb{K}$. Donc $gIm \iff g \not I \; \bar{m}$. On pose $\overline{M} := \{\bar{m} \mid m \in M\}$ et $\overline{I} := (G \times M) \setminus I$. Le *contexte complémentaire* de $\mathbb{K} := (G, M, I)$ est défini par $\overline{\mathbb{K}} := (G, \overline{M}, (G \times M) \setminus I)$, et est isomorphe à $(G, \overline{M}, I)$, où $gI\bar{m} : \Longleftrightarrow \; g \not I \; m$. L'apposition $\mathbb{K} := \mathbb{K}_1 | \mathbb{K}_2$ de deux contextes $\mathbb{K}_1 := (G, M_1, I_1)$ et $\mathbb{K}_2 := (G, M_2, I_2)$ est la juxtaposition verticale de ces deux contextes pour avoir $\mathbb{K} := (G, M_1 \cup M_2, I_1 \cup I_2)$.

1.3 Arbre Patricia

L'algorithme BiP présenté en section 2 utilise une structure de données nommée arbre Patricia pour stocker et gérer les biclusters. Cette structure de données est pratique pour le stockage et la recherche de l'information codée en alphanumérique. Elle a été introduite par Morrison (cf. Morrison (1968) pour plus de détails) et constitue une représentation compacte de l'arbre trie. L'arbre Patricia est utilisé pour stocker un ensemble de chaînes de caractères. Cependant, contrairement à un trie régulier, les arêtes de l'arbre Patricia sont identifiées par des séquences de caractères plutôt qu'avec des caractères simples. Il peut s'agir de chaînes de caractères, de chaînes de bits comme des nombres entiers, ou des séquences généralement arbitraires d'objets stockés dans un ordre lexicographique. Les arêtes de l'arbre ne sont donc plus étiquetés par des lettres mais par des mots. L'idée est de fusionner les branches en ne gardant que les nœuds internes avec au moins deux fils. Ainsi, chaque noeud de l'arbre possédant un fils unique sera fusionné avec son parent. En outre, aucun réaménagement de données ou d'index n'est nécessaire lorsque de nouvelles données doivent être ajoutées à l'arbre. Cette structure permet de récupérer des informations en réponse à des clés [1] fournies par l'utilisateur avec un temps de calcul borné linéairement par la longueur des clés et du nombre de leurs occurrences. Patricia nécessite un espace mémoire et des temps de calcul qui sont relativement faibles pour le problème défini. Notons que la recherche d'une chaîne s dans un arbre Patricia est semblable à celle de s dans une structure trie sauf que lorsqu'on traverse une arête de l'arbre, on vérifie l'étiquette de l'arête contre une sous-chaîne entière de s et pas seulement un seul caractère. L'opération de recherche d'une chaîne s prend un temps égal à $O(|s|)$. Quant à l'insertion d'une chaîne s, elle consomme un temps égal à $O(|s| + |\sum|)$ puisqu'elle implique

1. Une clé est une séquence de caractères allant de la racine vers un nœuds non terminal le plus bas.

une recherche de s suivie de la création d'au plus deux nouveaux nœuds, chacun de taille $O(|\sum|)$, avec $|\sum|$ représentant la cardinalité de l'alphabet utilisé Morin (2014). Dans le cadre de cet article, les mots sont une combinaison d'objets du contexte $\mathbb{K}$ tandis que les valeurs stockées dans les nœuds terminaux de l'arbre sont les attributs du contexte. Un exemple d'arbre Patricia est donné à la figure 1.

Après ce bref aperçu sur le biclustering, l'analyse formelle de concepts et la structure d'arbre Patricia, nous décrivons ci-après le fonctionnement de l'algorithme de biclustering *BiP*. Une illustration de l'exécution d'une telle procédure pour la production de biclusters de type 3 est fournie en section 3. On présente ensuite une analyse empirique préliminaire en section 4. Finalement, la section 5 conclut l'article et énumère les travaux futurs.

2 Algorithme BiP

Nous présentons dans cette section l'algorithme BiP qui permet de générer des biclusters issus d'un contexte binaire et dont le type peut être parmi l'un des suivants :

1. Toutes les valeurs du bicluster sont égales à 1
2. Toutes les valeurs du bicluster sont égales à 0
3. Les colonnes du bicluster sont pleines de 0 et/ou de 1.

Le type 1 donne lieu à des concepts formels (rectangles maximaux), alors que le type 2 va générer des rectangles maximaux comportant uniquement des 0, c.-à-d. des concepts formels présentant l'absence de propriétés pour une collection d'objets et donc des concepts formels du contexte $\overline{\mathbb{K}} := (G, \overline{M}, \overline{I})$.

Nous allons fournir dans ce qui suit la définition et les procédures de génération des biclusters (X, Y) de types 1 à 3 tout en mettant davantage l'accent sur la production de biclusters de type 3. Il est important de noter que l'ensemble des biclusters de type 3 est équivalent à l'ensemble de concepts formels qui peuvent être produits d'une manière naïve à partir de l'apposition (juxtaposition) du contexte initial $\mathbb{K}$ avec son complémentaire $\overline{\mathbb{K}}$. Toutefois, l'algorithme 2 génère d'une manière plus astucieuse et efficace cet ensemble qui représente en fait un sur-ensemble de la collection des biclusters de types 1 et 2 puisqu'il inclut aussi ceux qui contiennent simultanément des colonnes remplies de 1 et des colonnes remplies de 0.

2.1 Définition

Un bicluster de type 3 de $\mathbb{K} := (G, M, I)$ est un couple (X, Y) avec $X \subseteq G, Y \subseteq M \cup \overline{M}$, $|X| \geq 1, |Y| > 1$ vérifiant les deux conditions suivantes avec I représentant la relation binaire entre G et $M \cup \overline{M}$ (cf. sous-section 1.2) :

1. $\forall m_t \in Y, \exists m_l \neq m_t \in Y$ tel que $\forall o \in X \; oIm_t \wedge oIm_l$
2. (X, Y) est maximal pour l'inclusion. Cela signifie que :
   $X \times Y \subset I$ et $\forall X_1 \supseteq X, \forall Y_1 \supseteq Y$, ($X_1 \times Y_1 \subset I$, et (X_1, Y_1) vérifie la condition 1 $\Rightarrow X = X_1, Y = Y_1$).

La première condition signifie que deux colonnes (attributs positifs ou négatifs de $M \cup \overline{M}$) quelconques se trouvent dans un même bicluster de type 3 si les attributs correspondants sont

distincts et ne représentent pas un attribut et sa négation. En outre, elles ont uniquement des 1, uniquement des 0, ou l'une est une colonne totalement remplie de 1 alors que la seconde est totalement remplie de 0 ou inversement.

2.2 Algorithmes

Partant du contexte (matrice) $\mathbb{K}$, l'algorithme 1 permet d'obtenir des biclusters de type 1 ou 2 suivant la valeur du paramètre p. Pour le type 1, la valeur de p est égale à 1 alors qu'elle est de 0 pour le type 2. Pour obtenir les biclusters de type 3, l'algorithme 2 est exécuté sans aucun paramètre et sa première étape consiste à créer l'ensemble $\mathbb{C}$ lequel est l'union de deux ensembles $\mathbb{C}_0$ et $\mathbb{C}_1$ qui représentent des couples $(X, \{m\})$ où m est un attribut respectivement de M et de $\overline{M}$. L'ensemble $\mathbb{C}$ est donc composé de couples $(X, \{m\})$ où m est un attribut $\in M \cup \overline{M}$ dont les valeurs pour l'ensemble des objets de X sont égales à 1 ou à 0. Ce traitement est fait à travers les lignes 5 à 20 de l'algorithme. Ensuite, la procédure "Extraire" est lancée pour construire progressivement l'ensemble des biclusters à partir de $\mathbb{C}$. Les cas considérés dans cette procédure permettent d'assurer la condition 2 de maximalité et de générer tout nouveau bloc. Dans la suite de cet article, $\mathbb{T}$ représente un arbre Patricia et $\mathbb{T}[X]$ retourne les attributs associés aux objets contenus dans X à l'intérieur de l'arbre $\mathbb{T}$.

Algorithme 1 : BiP - Type 1 & 2

Entrées : $\mathbb{K} := (G, M, I)$ avec $M := \{m_1, \ldots, m_m\}$, $G := \{g_1, \ldots, g_n\}$ et p
Sorties : $\mathbb{T}$: Arbre Patricia contenant l'ensemble de biclusters

```
1  m ← |M|
2  n ← |G|
3  ℂ ← ∅
4  pour i ← 1 à m faire
5  |  Y ← {m_i}
6  |  X ← ∅
7  |  pour j ← 1 à n faire
8  |  |  si 𝕂[j, i] = p alors
9  |  |  |  X ← X ∪ {g_j}
10 |  fin
11 |  si X ≠ ∅ alors
12 |  |  ℂ ← ℂ ∪ {(X, Y)}
13 fin
14 𝕋 ← Extraire (ℂ)
15 Retourner 𝕋
```

Algorithme 2 : BiP - Type 3

Entrées : $\mathbb{K} := (G, M, I)$ avec $M := \{m_1, m_2, \ldots, m_m\}$ et $G := \{g_1, g_2, \ldots, g_n\}$
Sorties : $\mathbb{T}$: Arbre Patricia contenant l'ensemble des biclusters

$m \leftarrow |M|$
$n \leftarrow |G|$
$\mathbb{C}_0 \leftarrow \emptyset$
$\mathbb{C}_1 \leftarrow \emptyset$
pour $i \leftarrow 1$ ***à*** m **faire**
  $X_0 \leftarrow \emptyset$
  $X_1 \leftarrow \emptyset$
  pour $j \leftarrow 1$ ***à*** n **faire**
    si $\mathbb{K}[j, i] = 0$ **alors**
      $X_0 \leftarrow X_0 \cup \{g_j\}$
      $Y \leftarrow \{\overline{m_i}\}$
    sinon
      $X_1 \leftarrow X_1 \cup \{g_j\}$
      $Y \leftarrow \{m_i\}$
  fin
  si $X_0 \neq \emptyset$ **alors**
    $\mathbb{C}_0 \leftarrow \mathbb{C}_0 \cup \{(X_0, Y)\}$
  si $X_1 \neq \emptyset$ **alors**
    $\mathbb{C}_1 \leftarrow \mathbb{C}_1 \cup \{(X_1, Y)\}$
fin
$\mathbb{C} \leftarrow \mathbb{C}_0 \cup \mathbb{C}_1$
$\mathbb{T} \leftarrow Extraire\,(\mathbb{C})$
Retourner $\mathbb{T}$

Algorithme 3 : Procédure Extraire

Entrées : $\mathbb{C}$
Sorties : $\mathbb{T}$: arbre Patricia contenant l'ensemble des biclusters

$Z \leftarrow UnElement(\mathbb{C})$; $X_b \leftarrow$ ext(Z) ; $Y_b \leftarrow$ int(Z) ; $\mathbb{T}[X_b] \leftarrow Y_b$
$\mathbb{C} \leftarrow \mathbb{C} \setminus Z$
pour ***Chaque*** $(X_a, Y_a) \in \mathbb{C}$ **faire**
  pour ***Chaque*** $(X_b, Y_b) \in \mathbb{T}$ **faire**
    CAS :
    $X_b \subset X_a$: $\mathbb{T}[X_b] \leftarrow Y_b \cup Y_a$; $\mathbb{T}[X_a] \leftarrow Y_a$
    $X_a \subseteq X_b$: $\mathbb{T}[X_a] \leftarrow Y_b \cup Y_a$
    $(X_a \cap X_b \neq \emptyset)$ et $(X_b \not\subset X_a)$ et $(X_a \not\subseteq X_b)$: $\mathbb{T}[X_a \cap X_b] \leftarrow Y_b \cup Y_a$; $\mathbb{T}[X_a] \leftarrow Y_a$
    DÉFAUT : $\mathbb{T}[X_a] \leftarrow Y_a$
    FIN CAS :
  fin
fin
Retourner $\mathbb{T}$

2.3 Analyse théorique de complexité

Nous allons calculer la complexité temporelle de BiP et la comparer à celle de Bimax (Prelić et al., 2006). Notons que Bimax ne produit que des biclusters de type 1 (avec des cellules complètement remplies de 1) et a une complexité $O\left(n.m.\beta.min\left(n,m\right)\right)$ où β est le nombre de biclusters générés. Cette complexité est égale à $O\left(n^3.\beta\right)$ au pire cas.

Si nous supposons que le contexte $\mathbb{K}$ a plus de lignes que de colonnes $(n > m)$, alors Bimax a une complexité maximale de $O\left(n.m^2.\beta\right)$. Dans le cas de notre procédure BiP, la complexité de l'algorithme 1, produisant des coclusters pleins de 1 comme Bimax, peut être calculée comme suit. Le temps nécessaire à la création de l'ensemble $\mathbb{C}$ est $O(m.n)$. L'exécution de la procédure *Extraire* est $O(m.\beta)$. Les opérations d'intersection, de comparaison et d'union d'ensembles de l'algorithme 3 (procédure "Extraire") peuvent être exécutées en temps linéaire. En plus, l'ajout et le remplacement des éléments dans l'arbre $\mathbb{T}$ peuvent être réalisés en moyenne dans un temps $O\left(1\right)$ et au pire cas $O\left(n\right)$. Nous en déduisons que l'algorithme 2 produit les biclusters en un temps $O\left(n(m+2\beta)\right)$ dans le pire cas. En conclusion, la complexité de BiP est inférieure à celle de Bimax.

3 Illustration de l'algorithme

Pour illustrer les étapes de l'algorithme 2 et mettre en évidence la richesse sémantique des biclusters obtenus (composés d'attributs positifs et négatifs), on a utilisé un contexte $\mathbb{K}$ (ci-dessous) ayant 5 objets et 5 attributs.

$\mathbb{K}$	R_1	R_2	R_3	R_4	R_5
C_1	1	1	1	1	1
C_2	1	1	1	1	1
C_3	0	1	0	0	0
C_4	0	1	0	1	1
C_5	0	1	0	0	1

TAB. 1 – *Un contexte binaire* $\mathbb{K}$.

Pour simplifier les notations, nous omettrons les accolades de la notation ensembliste et écrirons par exemple C_1C_2 au lieu de $\{C_1, C_2\}$. On commence par former les ensembles $\mathbb{C}_1$ et $\mathbb{C}_0$ constitués de couples $(X, \{m\})$ où $m \in M$ respectivement $m \in \overline{M}$.

$$\mathbb{C}_1 = \{(C_1C_2, R_1);\ (C_1C_2C_3C4C_5, R_2); \ldots; (C_1C_2C_4C_5, R_5)\}$$
$$\mathbb{C}_0 = \left\{(C_3C_4C_5, \overline{R}_1);\ (C_3C_4C_5, \overline{R}_3), (C_3C_5, \overline{R}_4);\ (C_3, \overline{R}_5)\right\}.$$

L'étape suivante consiste à stocker dans l'arbre Patricia $\mathbb{T}$ des couples d'objets attributs. Une clé composée d'une liste d'objets permet de retrouver les attributs partagés par l'ensemble des objets formant cette clé. Comme indiqué auparavant, le choix d'une telle structure de données se justifie par son efficacité de stockage des données et de recherche par clé des éléments. On commence par placer le premier couple $(C_3C_4C_5, \overline{R}_1)$ dans $\mathbb{T}$. On a $\mathbb{T}[C_3C_4C_5] = \overline{R}_1$. À l'étape suivante, on compare le couple $(X_a, Y_a) = (C_1C_2, R_1)$ avec tous les éléments de

X_a	$\mathbb{T}$	Ligne ou opération	X_b
$C_3C_4C_5$	$\mathbb{T}[C_3C_4C_5] = \overline{R}_1$	1	$C_3C_4C_5$
C_1C_2	$\mathbb{T}[C_1C_2] = R_1$	9	C_1C_2
			$C_3C_4C_5$
$C_1C_2C_3C_4C_5$	$\mathbb{T}[C_3C_4C_5] = \overline{R}_1R_2$	$\supset$	$C_3C_4C_5$
	$\mathbb{T}[C_1C_2] = R_1R_2$	$\supset$	C_1C_2
	$\mathbb{T}[C_1C_2C_3C_4C_5] = R_2$		$C_1C_2C_3C_4C_5$
$C_3C_4C_5$	$\mathbb{T}[C_3C_4C_5] = \overline{R}_1R_2\overline{R}_3$	=	$C_3C_4C_5$
	$\mathbb{T}[C_1C_2] = R_1R_2$	9	C_1C_2
	$\mathbb{T}[C_1C_2C_3C_4C_5] = R_2$	$\subseteq$	$C_1C_2C_3C_4C_5$
C_1C_2	$\mathbb{T}[C_3C_4C_5] = \overline{R}_1R_2\overline{R}_3$		$C_3C_4C_5$
	$\mathbb{T}[C_1C_2] = R_1R_2R_3$	$\ldots$	C_1C_2
	$\mathbb{T}[C_1C_2C_3C_4C_5] = R_2$		$C_1C_2C_3C_4C_5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
C_3	$\mathbb{T}[C_1C_2C_3C_4C_5] = R_2$		$C_1C_2C_3C_4C_5$
	$\mathbb{T}[C_1C_2C_4C_5] = R_2R_5$		$C_1C_2C_4C_5$
	$\mathbb{T}[C_1C_2C_4] = R_2R_4R_5$		$C_1C_2C_4$
	$\mathbb{T}[C_1C_2] = R_1R_2R_3R_4R_5$		C_1C_2
	$\mathbb{T}[C_3C_4C_5] = R_2\overline{R}_3\overline{R}_1$		$C_3C_4C_5$
	$\mathbb{T}[C_4C_5] = R_2R_5\overline{R}_3\overline{R}_1$	$\ldots$	C_4C_5
	$\mathbb{T}[C_3C_5] = R_2\overline{R}_3\overline{R}_1\overline{R}_4$		C_3C_5
	$\mathbb{T}[C_5] = R_2\overline{R}_3\overline{R}_1\overline{R}_4R_5$		C_5
	$\mathbb{T}[C_3] = R_2\overline{R}_3\overline{R}_1\overline{R}_4\overline{R}_5$		C_3
	$\mathbb{T}[C_4] = \overline{R}_1R_2\overline{R}_3R_4R_5$		C_3

TAB. 2 – *Différents états de* $\mathbb{T}$.

$\mathbb{T}$. Mais, l'arbre $\mathbb{T}$ contient actuellement juste le couple $(X_b, Y_b) = (C_3C_4C_5, \overline{R}_1)$. Comme on est dans le cas par défaut (cf. ligne 9 de l'algorithme 3), on ajoute le couple $(X_a, Y_a) = (C_1C_2, R_1)$ à l'arbre $\mathbb{T}$. Puis, on compare le couple $(X_a, Y_a) = (C_1C_2C_3C_4C_5, R_2)$ avec les éléments de $\mathbb{T}$. Pour les deux éléments actuels dans $\mathbb{T}$, on applique les instructions de la ligne 6 de l'algorithme 3. Ainsi, $\mathbb{B}[C_3C_4C_5] = \overline{R}_1R_2$, $\mathbb{T}[C_1C_2] = R_1R_2$ et $\mathbb{T}[C_1C_2C_3C_4C_5] = R_2$.

La figure 1 illustre la structure de l'arbre Patricia où une clé est représentée par la séquence d'objets apparaissant dans les nœuds non terminaux et fait état des objets du bicluster alors que ses propriétés (intention) se trouvent dans une feuille qui suit la clé.

Comme nous pouvons le voir, différents types de biclusters peuvent être extraits. Certains avec juste des 1, d'autres avec uniquement des 0 et finalement des biclusters contenant à la fois des colonnes de 1 et des colonnes de 0. Considérons les deux biclusters de la table 3 extraits de la structure de l'arbre. Le premier bicluster est $(C_1C_2, R_1R_2R_3R_4R_5)$ et est obtenu par un parcours de la première branche de gauche de l'arbre, et le second est $(C_4C_5, \overline{R}_1R_2\overline{R}_3R_5)$ et est extrait à partir de la quatrième branche en comptant depuis la droite de l'arbre.

Comme indiqué précédemment, un nœud de l'arbre Patricia avec un seul fils est automatiquement fusionné avec le nœud parent. La branche de l'arbre $\mathbb{T}$ partant de la racine et comportant les objets 1-2 et 3-4-5 en est une illustration.

Tandis que le bicluster B_1 montre que les deux objets C_1 et C_2 possèdent tous les attributs du contexte $\mathbb{K}$, le bicluster B_2 nous informe non seulement sur les attributs que possèdent les objets C_4 et C_5 mais aussi sur ceux qu'ils ne possèdent pas. Une telle connaissance est

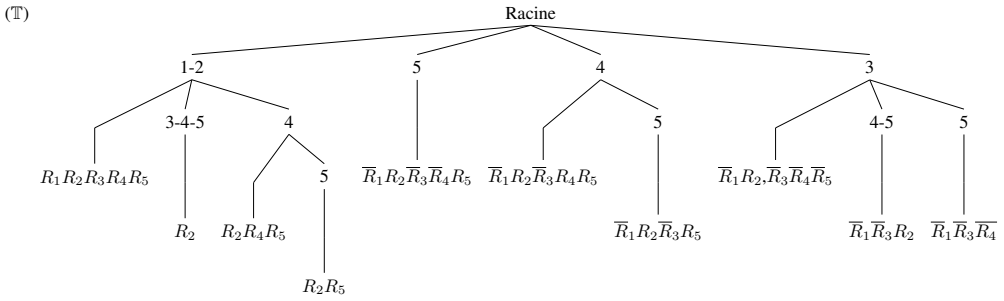

FIG. 1 – *États de l'arbre $\mathbb{T}$ suite à l'insertion des concepts formels.*

B_1	R_1	R_2	R_3	R_4	R_5
C_1	1	1	1	1	1
C_2	1	1	1	1	1

B_2	R_1	R_2	R_3	R_5
C_4	0	1	0	1
C_5	0	1	0	1

TAB. 3 – *Deux biclusters de type 3 extraits du contexte binaire $\mathbb{K}$.*

extraite d'un seul bicluster sans recourir à d'autres biclusters. Notons que les biclusters B_1 et surtout B_2 ne sont rien d'autres que des concepts formels qu'on aurait pu obtenir en faisant l'apposition du contexte $\mathbb{K}$ avec son complémentaire. Toutefois, notre algorithme BiP permet de les générer sans recours à cette apposition.

Tel qu'on a pu le constater sur l'exemple des biclusters de type 3, la connaissance extraite est plus riche et plus variée. Toutefois, un type de bicluster n'est pas systématiquement meilleur qu'un autre, cela dépend de l'usage qu'on veut faire des résultats obtenus et de la nature des informations et des connaissances qu'on cherche à extraire.

4 Expérimentation

Dans cette section, nous avons choisi de mener une série de tests sur un jeu de 20 contextes binaires générés aléatoirement par le système Coron (Kaytoue et al., 2010) en vue d'estimer la performance de notre algorithme à produire des concepts formels combinant la présence d'attributs et l'absence d'autres attributs (biclusters de type 3). Ensuite, nous avons pris un jeu de données réel du domaine médical (Zwitter et Soklic, 1988) afin de mettre en évidence l'utilité et la pertinence des biclusters de type 3. Les expérimentations ont été menées sur une machine Intel 3.2 GHz avec une mémoire RAM de 16 GO et le système d'exploitation Windows 10.

4.1 Données synthétiques

Les 20 contextes générés synthétiquement sont répartis en deux groupes. Le premier groupe contient des matrices binaires de 50 colonnes chacune mais avec un nombre de lignes (objets) variant de 25 à 1500. Le second groupe a des matrices de 25 à 375 lignes mais un nombre fixe

de 50 colonnes. La densité des matrices du premier groupe est fixée à 25% alors que celle du deuxième groupe est de 40%.

Le tableau suivant nous donne pour chacun des contextes le nombre de concepts générés, le temps moyen pour produire un concept ainsi que le temps total moyen requis par l'algorithme pour produire la totalité des concepts. Les résultats expérimentaux montrent que notre

Nombre de lignes	Densité %	Nb. concepts	Temps par concept en **millisecondes**	Temps total en secondes
25	25	421	0.143	0.060
50	25	1 645	0.133	0.219
150	25	12 971	0.143	1.867
250	25	33 774	0.184	6.214
500	25	111 099	0.236	26.284
750	25	218 300	0.298	65.113
1000	25	368 263	0.359	132.292
1250	25	504 454	0.417	210.495
1500	25	675 228	0.464	313.741
25	40	2 541	0.175	0.446
50	40	14 024	0.156	2.196
100	40	98 717	0.172	17.002
150	40	267 523	0.180	48.208
200	40	567 934	0.202	114.990
250	40	887 355	0.217	192.600
275	40	1 083 192	0.221	240.153
300	40	1 463 826	0.223	327.233
325	40	1 690 886	0.281	475.681
350	40	2 142 236	0.322	691.248
375	40	2 635 170	0.323	851.16

TAB. 4 – *Temps d'exécution en fonction de la densité du contexte et du nombre d'objets.*

algorithme prend, à titre d'exemple, un temps moyen de production d'un concept de l'ordre de 4.6 millisecondes pour une matrice de 1500 lignes (objets), 50 colonnes (attributs) et une densité de 25% et que globalement, 313 secondes sont nécessaires pour produire tous les 675 228 biclusters. Avec un nombre plus réduit d'objets mais une densité plus grande (souvent rare dans les cas réels), les résultats montrent qu'un contexte de 375 objets, 50 attributs et une densité de 40% a permis de générer 2 635 170 biclusters avec un temps d'exécution moyen par concept de l'ordre de 0.32 millisecondes et un temps global de 851 secondes.

Remarquons qu'en faisant varier le nombre de lignes (objets) d'un contexte dont la densité est de 25% entre 0 et $1\ 500$ le temps nécessaire pour la production des concepts varie pratiquement de façon linéaire tel qu'illustré par la figure 2. Cependant, dès que la densité atteint 40%, ce temps devient quadratique à cause de l'explosion du nombre de concepts produits.

Des tests de performance antérieurs ont permis de confirmer la supériorité de BiP par rapport à Bimax (Balamane, 2017) dans la production de biclusters de type 1. Pour la production des biclusters de type 3, la comparaison de BiP avec Bimax aurait nécessité un prétraitement

découverte de biclusters

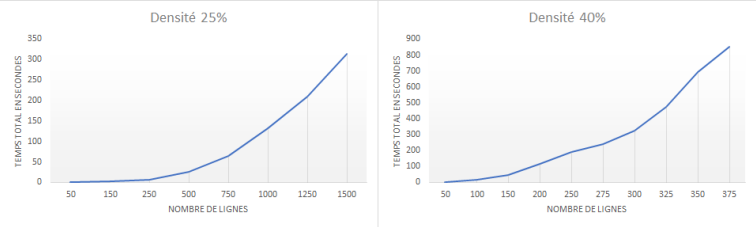

FIG. 2 – *Temps total en secondes pour le calcul des concepts issus de contextes ayant des densités de* 25% *et* 40%.

des données d'entrée de Bimax pour effectuer l'apposition du contexte initial avec son complémentaire, l'exécution de Bimax puis un post-traitement des résultats pour faire ressortir l'absence (négation) d'attributs et afficher les blocs d'une manière compacte.

4.2 Données réelles

Le jeu de données utilisé dans cette analyse comprend les dossiers médicaux de 286 patientes souffrant du cancer du sein. L'information recueillie sur chacune de ces patientes a été enregistrée au niveau de 10 attributs définis comme suit :

1. Récurrence du cancer (oui, non)
2. Âge (20-29, 30-39, 40-49, 50-59, 60-69, 70-79)
3. Ménopause (avant 40 ans, après 40 ans, pré-menopause)
4. Taille tumeur : (0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59)
5. Nœuds envahis : (0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39)
6. Ganglions lymphatiques atteints (oui, non)
7. degré de malignité (1, 2, 3)
8. Sein (gauche, droit)
9. Cadrant du sein (gauche-sup, gauche-bas, droit-sup, droit-bas, central)
10. Radiation (oui, non)

Dans un premier temps, nous avons converti les données en une matrice binaire de 286 objets (patientes) et 38 attributs. Comme nous l'avons mentionné ci-dessus, une étude préalable Rodríguez-Jiménez et al. (2016) sur ce jeu de données a montré l'utilité de considérer dans une analyse médicale autant la présence que l'absence d'attributs. Celle-ci a permis d'aboutir à des résultats très pertinents qu'on n'aurait pas pu obtenir sans avoir pris en considération autant les attributs positifs (présence) que négatifs (absence).

Un premier résultat intéressant est que la valeur positive de l'attribut de récurrence est toujours apparue être reliée aux valeurs négatives des attributs âge (20-29) ou (70-79) lorsque la patiente est ménopausée avant l'âge de 40 ans. Cela signifie qu'aucun cas de récurrence

chez les patientes suivies n'a été constaté lorsque l'âge de la patiente est entre (20-29) ou (70-79) et que la ménopause est survenue chez cette patiente avant l'âge de 40 ans. Un autre résultat intéressant est que l'atteinte des ganglions lymphatiques indique un degré de malignité élevé de la tumeur cancéreuse (2 ou 3) lorsque la patiente est âgée entre 30 et 69 ans et a été ménopausée avant l'âge de 40 ans.

Un autre résultat important a pu être obtenu à travers les biclusters de type 3 ne contenant que la présence de propriétés. Il concerne les patientes dont la tranche d'âge est 30-39 ans. Plusieurs biclusters mettent en évidence le niveau de gravité de la tumeur cancéreuse chez les femmes de cette tranche d'âge lorsqu'elles sont pré-ménopausées. Le degré de malignité de leur tumeur est supérieur ou égal à 2, sachant que le niveau 3 est le plus élevé. Pour l'ensemble de ces patientes, la tumeur est localisée dans la partie inférieure droite du sein. Nous avons noté aussi que le nombre de nœuds impliqués dans une éventuelle migration de la tumeur est corrélé positivement avec le degré de malignité de la tumeur.

5 Conclusion et discussions

Nous avons présenté une nouvelle méthode de biclustering pouvant prendre en entrée une matrice d'adjacence de type binaire à partir de laquelle on peut générer des biclusters d'au moins trois types Balamane (2017). À travers les types 2 (absence d'attributs) et 3 (présence et/ou absence d'attributs), nous désirons introduire la notion d'absence de propriétés au sein des biclusters. Dans cet article, nous avons mis l'accent sur le type 3 de bicluster et illustré le fait que des objets sont similaires non seulement par rapport à la présence mais également à l'absence de propriétés. Ce type peut être particulièrement utile dans des domaines d'application comme la bioinformatique, le marketing et l'analyse des réseaux sociaux pour mieux révéler des groupes d'entités par rapport à l'absence et/ou la présence de certains attributs. À titre d'exemple, le fait qu'un groupe de clients achète certains produits mais jamais d'autres produits peut mener vers un marketing ciblé de ce groupe et des offres avantageuses pour les produits non sollicités.

Comme le nombre de biclusters de type 3 peut être très grand, nos travaux futurs visent à définir un ensemble d'opérations d'interrogation des biclusters stockés dans la structure d'arbre Patricia dans le but de répondre à des besoins et requêtes spécifiques de l'utilisateur. À titre d'exemple, l'utilisateur peut demander à trouver un bicluster dont l'extension comporte au moins (ou plus, ou exactement) un ensemble donné d'objets afin de déterminer les propriétés qui le définit. La structure des données actuellement utilisée pourra être combinée avec des techniques d'indexation et de hachage pour une meilleure recherche des motifs. En outre, l'évolution des données dans le temps nous amène à finaliser une procédure incrémentale de construction de l'arbre Patricia des biclusters afin de mettre à jour un ensemble de biclusters lorsque de nouveaux objets sont ajoutés.

Au niveau des analyses empiriques, nous envisageons d'utiliser des ensembles de données de diverses tailles et densités pour comparer les performances de BiP dans la génération des trois types de biclusters avec des algorithmes connus de calcul de l'ensemble de concepts formels tels que l'algorithme *Next Closure* Ganter et Wille (1999) et l'algorithme de Nourine Nourine et Raynaud (1999).

Références

Bala, H., E. Labonté-LeMoyne, et P.-M. Léger (2017). Neural correlates of technological ambivalence : A research proposal. In *Information Systems and Neuroscience*, pp. 83–89. Springer.

Balamane, A. (2017). *Découverte et gestion de motifs en analyse formelle de concepts*. Thèse de doctorat, Université du Québec en Outaouais.

Beauguitte, L. (2011). Blockmodeling et équivalences. Rapport <halshs-00566474>, CNRS, groupe fmr (flux, matrices, réseaux).

Busygin, S., O. Prokopyev, et P. M. Pardalos (2008). Biclustering in data mining. *Computers & Operations Research 35*(9), 2964–2987.

Charrad, M., Y. Lechevallier, G. Saporta, et M. Ben Ahmed (2008). Le bi-partitionnement : état de l'art sur les approches et les algorithmes. Ecol'IA 2008.

Dhillon, I. S., S. Mallela, et D. S. Modha (2003). Information-theoretic co-clustering. In *Proceedings of the ninth ACM KDD*, pp. 89–98. ACM.

Everett, M. G. et S. P. Borgatti (2013). The dual-projection approach for two-mode networks. *Social Networks 35*(2), 204–210.

Ganter, B. et R. Wille (1999). *Formal concept analysis : mathematical foundations*. Springer Science & Business Media.

Govaert, G. et M. Nadif (2013). *Co-Clustering* (1st ed.). Wiley-IEEE Press.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the american statistical association 67*(337), 123–129.

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

Kaytoue, M., F. Marcuola, A. Napoli, L. Szathmary, et J. Villerd (2010). The coron system. In *ICFCA Supplementary Proceedings*, pp. 55–58.

Knoke, D. et S. Yang (2008). *Social Network Analysis* (Second Edition ed.). Sage Publications, Inc.

Lewis, D. D., Y. Yang, T. G. Rose, et F. Li (2004). Rcv1 : A new benchmark collection for text categorization research. *The Journal of Machine Learning Research 5*, 361–397.

Li, L., Y. Guo, W. Wu, Y. Shi, J. Cheng, et S. Tao (2012). A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data. *BioData mining 5*(1), 1–10.

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

Morin, P. (2014). *Advanced Data Structures - Course Notes*. Carleton University. http ://cglab.ca/ morin/teaching/5408/notes/strings.pdf.

Morrison, D. R. (1968). PATRICIA - practical algorithm to retrieve information coded in alphanumeric. *J. ACM 15*(4), 514–534.

Nourine, L. et O. Raynaud (1999). A fast algorithm for building lattices. *Information Processing Letters 71*(5), 199 – 204.

Prelić, 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), 1122–1129.

Rodríguez-Jiménez, J. M., P. Cordero, M. Enciso, et A. Mora (2016). Data mining algorithms to compute mixed concepts with negative attributes : an application to breast cancer data analysis. *Mathematical Methods in the Applied Sciences 39*(16), 4829–4845.

Zwitter, M. et M. Soklic (1988). Breast cancer data. Institute of Oncology, University Medical Center, Ljubljana, Yugoslavia. "`https://archive.ics.uci.edu/ml/datasets/Breast+Cancer`".

Summary

Most of the existing biclustering algorithms take into account the properties that hold for a set of objects. However, it could be very useful in several application domains such as organized crime, genetics or digital marketing to identify homogeneous groups of similar objects in terms of both the presence and the absence of attributes. In this paper, we present a generic method of biclustering that exploits a binary matrix to produce at least three types of biclusters: (i) those where all values are equal to 1, (ii) those where all values are equal to 0, and (iii) those indicating the presence of certain attributes and/or the absence of other attributes without the need to take into account the complementary of the initial binary context (matrix). The implementation and validation of the method on data sets illustrate its potential in the discovery of relevant patterns.