N° d'ordre : 2006-ISAL-0078 Année 2006

THÈSE

présentée devant

L'Institut National des Sciences Appliquées de Lyon

pour obtenir

Le Grade de Docteur

Spécialité

Informatique

Ecole Doctorale : Informatique et Information pour la Société

par

Ruggero Gaetano Pensa

Un cadre générique pour la co-classification sous contraintes : application à l'analyse du transcriptome

Soutenue publiquement le 20/11/2006 devant le jury :

Jean-François Boulicaut, Professeur, INSA de Lyon	Co-directeur
Bruno Crémilleux, Professeur, Université de Caen	Rapporteur
Luc De Raedt, Professeur, Université de Freiburg (Allemagne)	Examinateur
Céline Robardet, Maître de Conférences, INSA de Lyon	Co-directeur
Céline Rouveirol, Professeur, Université Paris-Nord	Rapporteur
Marc Sebban, Professeur, Université de Saint-Etienne	Examinateur

Je tiens tout d'abord à exprimer mes plus vifs remerciements à Jean-François Boulicaut et Céline Robardet : il y a trois ans, ils ont cru dans mes possibilités, m'ont suivi de manière constante, et m'ont transmis l'amour pour la recherche et le travail d'équipe.

Je tiens à remercier très chaleureusement mes "coturnes" Jérémy Besson, maître de ski dont la présence a toujours été indispensable, et Ieva Mitasiunaite, avec sa présence parfois un peu bruyante, mais qui a su me supporter pendant deux ans.

Merci aussi à tous les membres de l'équipe "Data Mining et Bases de Données Inductives" de l'INSA et en particulier à Claire Leschi pour ses précieux conseils et son soutien moral, et Christophe Rigotti, pour des échanges brillants et motivants.

Je souhaite aussi remercier les membres de mon jury pour leur travail et les conseils qu'ils m'ont prodigués.

La préparation d'une thèse est constellée de moments de joie et satisfaction, mais aussi de quelques moments de détresse. Je tiens donc à remercier toutes les personnes qui ont partagé avec moi ces instants : mes parents et ma sœur Sara, qui n'ont jamais cessé de m'aider et de me conseiller, mes grands-parents, et en particulier, Nonna Sina, qui n'a jamais cessé de prier pour moi, mes amis turinois et ex-turinois (Alfonso, Elena, Gianvito "Spezza", Giovanni "il Fighetto", Giovanni "Satiro", Matteo), mes ex-camarades Clara et Luigi "Comio", mon amis depuis toujours Paolo "lu Dottore". Merci également aux "ritals" (Ciccio, Rosario et Sandro), mes copains des ces années en France.

Merci, enfin, à Chiara, ses pêches, ses pommes et ses veaux.

Ce mémoire est dédié à Francesca et Fabiana.

La preparazione di un dottorato è costellata da momenti di gioia e soddisfazione, ma anche da qualche momento di sconforto. Un ringraziamento va quindi a tutte le persone che hanno condiviso con me questi istanti : i miei genitori e mia sorella Sara, che non hanno mai smesso di aiutarmi e consigliarmi, i miei nonni, ed in particolare Nonna Sina, che non ha mai smesso di pregare per me, i miei amici torinesi ed ex-torinesi (Alfonso, Elena, Gianvito "Spezza", Giovanni "il Fighetto", Giovanni "Satiro", Matteo), i miei ex-compagni di corso Clara e Luigi "Comio", il mio amico di sempre Paolo "lu Dottore". Grazie anche ai "ritals" (Ciccio, Rosario e Sandro), miei compagni di questi anni in Francia.

Grazie, infine, a Chiara, alle sue pesche, le sue mele e i suoi vitelli.

Questa tesi è dedicata a Francesca e Fabiana.

Résumé

La recherche de groupements intéressants dans les données booléennes (ensembles d'objets décrits par un ensemble de propriétés) a motivé la conception de méthodes d'extractions de motifs globaux (partitions) et de motifs locaux (ensembles fréquents, règles d'association et concepts formels). Cette thèse concerne la co-classification c'est-à-dire le calcul de bi-partitions (couplage de partitions sur les deux dimensions). Les algorithmes de co-classification disponibles ne permettent aux analystes d'exploiter leur connaissance du domaine qu'à travers un nombre réduit de paramètres. D'autre part, les techniques d'extraction de motifs locaux produisent d'énormes collections qui sont difficilement exploitables et interprétables. Nous avons développé une nouvelle méthode de co-classification qui calcule des bi-partitions à partir de motifs capturant des associations localement fortes (e.g., des concepts formels, une forme de motif tolérant aux exceptions appelé δ-bi-ensemble). Le principe consiste à exploiter l'information contenue dans la collection des motifs locaux en la propageant au niveau global pour faciliter l'optimisation de la fonction objectif. Il devient alors possible de propager un certain nombre de contraintes depuis l'extraction des motifs locaux jusqu'à la construction de la bi-partition (e.g., pour imposer des formes particulières aux groupes calculés). Il s'agit donc d'une contribution au domaine très récent de la classification sous contraintes. Une approche duale consiste à utiliser des motifs locaux pour faciliter l'interprétation de bi-partitions déjà calculées. Pour ce faire, nous proposons une méthode de caractérisation des bi-clusters au moyen de motifs locaux auxquels sont associés des mesures d'intérêt. L'application de nos méthodes à l'analyse de données d'expression de gènes a montré la pertinence de nos propositions pour expliciter des hypothèses biologiques plausibles.

Mots clés Extraction de connaissances, co-classification, classification conceptuelle, motifs locaux ensemblistes, concepts formels, motifs tolérants au bruit, bi-partitions sous contraintes, caractérisation de classes, transcriptome.

Abstract

The search for interesting groups in boolean data (sets of objects described by sets of properties) has motivated the design of methods for computing global patterns (e.g.., partitions), and extracting local patterns s(e.g., frequent itemsets, association rules, formal concepts. This thesis concerns co-clustering, i.e., computing bi-partitions (coupled partitions on both dimensions). When using available co-clustering algorithms, the user can hardly exploit his/her domain knowledge since he/she has limited possibilities for setting just a few parameters. On the other hand, classical local pattern mining techniques usually provide huge collections of patterns that are hard to evaluate and interpret. We have designed a new co-clustering framework which computes a bi-partition by starting from collections of patterns that capture locally strong associations (e.g., formal concepts, δ-bi-set that are a form of fault-tolerant patterns). The idea is that the available information about the local patterns can be exploited to build a relevant global pattern. It becomes possible to consider the declarative specification of constraints on the bi-partitions (e.g., user-defined requirements about the shape of clusters) and to use such constraints at the local pattern mining step and then during the co-clustering phase. As such, our proposal is a contribution to the recent domain of constraint-based clustering. A dual approach consists in using local patterns to interpret bi-partitions. We propose a method for bi-cluster characterization by means of local patterns and their associated interestingness measures. The application of our methods to a gene expression data analysis scenario has illustrated the added-value of our proposal to give rise to plausible biological hypothesis.

Keywords

Knowledge discovery, co-clustering, conceptual clustering, local set patterns, formal concepts, noise tolerant patterns, bi-partition under constraints, cluster characterization, transcriptome.

Table des matières

Table des figures

Liste des tableaux

Introduction

Ce mémoire présente nos travaux de recherche sur la co-classification sous contraintes. Le domaine d'application privilégié est l'analyse du transcriptome, c'est-à-dire l'étude des mécanismes d'expression des gènes. La principale contribution est d'odre méthodologique puisque nous proposons un cadre générique pour l'utilisation combinée de motifs locaux (e.g., des itemsets fréquents, des concepts formels) et globaux (e.g., des bi-partitions).

Nous avons travaillé au sein de l'équipe "Data mining et bases de données inductives" de l'UMR LIRIS tout en ayant une étroite collaboration avec l'équipe "Bases moléculaires de l'auto-renouvellement et ses altérations" de l'UMR CGMC. C'est ce contexte inter-disciplinaire qui nous a permis d'expérimenter de nouvelles pistes pour l'analyse de données biologiques, notamment en vue de la compréhension des mécanismes de régulation des gènes. Il est désormais clair que l'utilisation d'une technique de fouille de données particulière ("Data Mining") n'est généralement pas suffisante pour étudier et comprendre des systèmes aussi complexes que les systèmes de régulation génique. La théorie des bases de données inductives suggère qu'un processus d'extraction de connaissances puisse être considéré comme une séquence de requêtes portant à la fois sur les données et sur différents types de motifs ou modèles dans ces données. Notre travail s'inscrit dans cette vision et, plus précisément, ses développements dans le cadre des projets européens IST FET cInQ (IST-2000-26469, 2001-2004) et IQ (FP6-516169, 2005-2008). Ce manuscrit montrera aux lecteurs, du moins nous l'espérons, l'intérêt scientifique qui a motivé ces projets, et l'enthousiasme qui a animé notre équipe et les personnes qui participent ou ont participé à ces projets.

Contexte

L'équipe "Data Mining et bases de données inductives" concentre une partie de ses efforts de recherche sur l'extraction de motifs comme des bi-ensembles dans des données booléennes (également appelées données transactionnelles ou données 0/1). Dans de telles données disponibles sous la forme de grandes matrices, une valeur

	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

"1" entre une ligne l et une colonne c indique que l et c sont liées par une certaine relation. Différentes relations peuvent être représentées. Par exemple, la table suivante peut indiquer si un gène (en colonne) est sur-exprimé ou non dans une condition expérimentale (en ligne), ou bien si un mot (en colonne) est présent ou non dans le texte d'un document (en ligne), ou encore si un produit (en colonne) à été acheté ou non lors d'une transaction commerciale (en ligne). Nous parlerons souvent d'objets pour les lignes et d'attributs pour les colonnes. Dans de telles matrices, on peut s'intéresser à des bi-ensembles de "1" (valeur vraie), i.e., des sous-ensembles de lignes associés des sous-ensembles de colonnes et qui ne contiennent que des valeurs "1". Ainsi, dans notre exemple, $(\{g_3, g_4\}, \{t_1, t_3, t_4\})$ est un tel bi-ensemble. Ce motif peut indiquer que les gènes g_3 et g_4 sont simultanément sur-exprimés dans les conditions biologiques t_1, t_2 et t_4, ou bien que les mot g_3 et g_4 sont présents dans les documents t_1, t_3 et t_4, ou enfin que les produits g_3 et g_4 ont été achetés ensembles lors des transactions t_1, t_3 et t_4. En étudiant ce motif, on s'aperçoit qu'il n'est pas possible de rajouter une colonne ou une ligne sans introduire des exceptions (c'est-à-dire des "0") dans le bi-ensemble. Un tel bi-ensemble sera dit maximal et correspond à des objets mathématiques bien étudiés : les "concepts formels" [Wil82]. En regardant cette table, on peut également remarquer que la propriété g_1 est presque toujours vraie quand les propriétés g_3 et g_4 le sont. On peut lire cette association de la manière suivante : "lorsque g_3 et g_4 sont vraies, alors g_1 est souvent vraie". Il s'agit de ce que l'on appelle une règle d'association (notée $g_3, g_4 \Rightarrow g_1$), un type de motif très étudié depuis une dizaine d'années [AIS93]. La popularité du problème de calcul des ensembles fréquents vient justement de son application à la découverte de règles d'association.

La caractéristique commune à ces types de motifs est celle d'être des motifs "locaux", c'est-à-dire, selon la définition de D. Hand [Han02], "un vecteur de données qui sert à décrire une densité anormalement haute de points de données". Une des spécialités de l'équipe du LIRIS est de développer des algorithmes (appelés solveurs) qui permettent d'extraire des collections correctes et complètes de motifs locaux, c'est-à-dire de tous (et rien d'autre que) les motifs satisfaisant les contraintes spécifiées. Par exemple, si l'on a spécifié une contraintes de fréquence minimale (resp. maximale), il s'agit de calculer très exactement la collection de tous les motifs fréquents (resp. infréquents). La correction et la complétude des algorithmes d'ex-

traction des motifs locaux peuvent être assurés par l'exploitation des propriétés des contraintes utilisées, notamment les propriétés de monotonicité qui permettent un élagage sûr dans les espaces de recherche.

Une fois ces collections de motifs extraites, leurs analyses peuvent se révéler très compliquées, notamment à cause de leurs tailles. En effet, il est courant de produire des collections de centaines de milliers de motifs locaux, voir même quelques millions. Un analyste se trouve alors devant des collections peu exploitables. D'autre part, il est souvent difficile de trouver quelles sont les contraintes qui permettent à la fois d'améliorer la pertinence des motifs tout en diminuant la taille des solutions.

Nous pouvons également nous intéresser à des collections de bi-ensembles particulières appelées "bi-partitions", c'est-à-dire des motifs où les ensembles d'objets déterminent une partition sur la totalité de l'ensemble d'objets, et les ensembles de propriétés déterminent une partition sur la totalité de l'ensemble des propriétés. Un exemple de bi-partition pour notre exemple est $\{\{\{t_1, t_3, t_4\},\{g_1, g_3, g_4\}\}, \{\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}\}\}$. Une interprétation courante d'une telle bi-partition sera de dire que la classe (ou "cluster") d'objets $\{t_1, t_3, t_4\}$ est caractérisée par le fait que les propriétés de $\{g_1, g_3, g_4\}$ sont presque toujours satisfaites. Dans une application d'analyse de données d'expression, une bi-partition permettra d'identifier des groupes de gènes qui sont spécifiques de certains groupes de conditions expérimentales. On pourra dire, par exemple, que les gènes g_1, g_3, g_4 sont majoritairement sur-exprimés dans les conditions expérimentales t_1, t_3 et t_4 plutôt que dans les autres conditions. Ce "plutôt que" met bien en évidence la différence d'un tel bi-ensemble (appelé aussi bi-cluster) par rapport à un motif local comme un concept formel. En effet, un "cluster" est défini comme un groupe d'objets ayant d'une part une forte similarité entre eux et d'autre part une forte dissimilarité avec les objets qui ne font pas partie du cluster. La définition d'une bi-partition dépend de la définition des similarités et dissimilarités, c'est-à-dire de la fonction objectif à optimiser. Ainsi, un algorithme de (co-)classification cherchera à maximiser la similarité intra-classe et la dissimilarité inter-classes. Pour trouver l'optimum de cette fonction, un algorithme correct et complet devrait être capable d'explorer l'espace de toutes les bi-partitions possibles. Il est clair qu'en pratique, une telle opération n'est pas faisable au regard du nombre astronomique de partitions possibles. Il existe donc des heuristiques pour se rapprocher de la solution optimale et permettre d'identifier un optimum local. Ainsi, les méthodes présentées dans [RF01b, DMM03] optimisent des fonctions objectifs qui prennent en compte à la fois les objets et les propriétés.

En fait, les applications qui nous intéressent, e.g., l'analyse de données d'expression de gènes, peuvent s'appuyer à la fois sur la découverte de motifs locaux et sur celle de motifs globaux comme des bi-partitions. D'autre part, d'un point de vue méthodologique, nous nous intéressons aux primitives qui permettent de spécifier de nouvelles propriétés ou contraintes sur les motifs à extraire. Un enjeu algorithmique majeur est alors de pouvoir évaluer les requêtes inductives construites sur

ces primitives, notamment dans le cadre de la co-classification sous contraintes. En effet, ces primitives ne sauraient se limiter à la seule déclaration de la fonction objectif à optimiser (où l'heuristique à utiliser), mais il faut permettre l'utilisation d'autres types de contraintes comme, par exemple, forcer des éléments à être (ou non) dans une même classe. Combiner l'optimisation de la qualité des groupements avec la satisfaction des autres contraintes pose des problèmes délicats. Des solutions ont été proposées pour des contraintes simples et en se limitant à la classification mono-dimensionnelle [WCRS01, BBM02, KKM02, BBM04a, DR05b, DR05a]. Dans le cadre de la co-classification, nous ne connaissons pas de méthodes exploitant des contraintes. En effet, la plupart des méthodes sous contraintes qui ont été développées (méthodes de classification semi-supervisée) cherchent à améliorer les performances de prédiction de la variable de classe lorsque peu d'instances sont étiquetées. Nous sortons clairement de ce contexte puisque, pour nous, les contraintes sont un outil permettant à l'analyste de spécifier la pertinence des groupements. Ainsi, si les objets correspondent à des situations ordonnées dans le temps, nous pourrions vouloir rechercher des bi-partitions qui préservent des successions d'objets contigus dans chacune des classes calculées.

On peut se demander maintenant, quel genre d'information est porté par un motif global tel qu'une bi-partition. En se basant sur des mesures qui prennent en compte la totalité des données, on peut considérer que cette information reflette des structures intrinsèques dans les données. Cela veut aussi dire que, en dehors des extractions dans des contextes totalement exploratoires, les motifs globaux ne sont généralement pas surprenants. De plus, le fait d'analyser de grandes classes de propriétés et/ou d'objets, rend difficile la tâche d'interprétation. Nous avons finalement un problème dual de celui que l'on trouve dans l'analyse des motifs locaux. Il est possible de travailler avec de très grandes collections de motifs locaux individuellement faciles à interpréter mais avec peu de possibilités d'interpréter les collections complètes. Il est également possible d'avoir des motifs globaux facilement interprétables grâce à cette globalité (e.g., une bi-partition) mais dont les éléments constitutifs peuvent être relativement gros (e.g., des classes de plusieurs milliers d'objets et/ou de propriétés) et donc difficilement analysables individuellement. Il apparaît clairement qu'il existe une certaine complémentarité entre les motifs locaux et les motifs globaux. En particulier, un motif global comme une bi-partition pourrait servir de point de départ pour aider l'analyste à sélectionner un certain nombre de motifs locaux qui sont, par exemple, caractéristiques de cette bi-partition ou, au contraire, qui apparaissent comme contradictoires avec le modèle que représente la bi-partition. Une autre idée consiste à vouloir exploiter l'information portée par les motifs locaux lors de la construction, par exemple, de meilleures bi-partitions au regard d'un problème particulier.

Le *leit motiv* de cette thèse est d'exploiter cette complémentarité entre motifs locaux et motifs locaux. En nous situant dans un cadre non supervisé, cette complémentarité se traduit pour nous dans la construction de bi-partitions sous contraintes et l'aide à leurs interprétations. Dans une optique d'analyse du trans-

criptome, cela signifie que l'utilisation conjointe de motifs globaux et locaux pourrait faciliter la découverte de nouvelles associations de gènes et faire émerger de nouvelles directions de travail en biologie moléculaire. Tel est l'état d'esprit qui nous a accompagné tout au long des nos travaux : il ne s'agit pas de vouloir remplacer un outil d'analyse mais d'utiliser au mieux les complémentarités entre des méthodes de fouille de données existantes. Ce faisant, nous avons également un terrain d'application remarquable du cadre des bases de données inductives. Les processus d'extraction de connaissance qui nous intéressent peuvent en effet être modélisés par des séquences de requêtes portant à la fois sur les données et sur divers types de motifs ou modèles.

Nous introduisons maintenant les principaux résultats obtenus dans le cadre de cette thèse.

Caractérisation de bi-partitions

Nous avons d'abord travaillé à l'interprétation des résultats d'une classification non supervisée, et en particulier d'une co-classification. Lorsque l'on cherche à analyser le contenu d'un bi-cluster, e.g., en analyse du transcriptome, il est courant d'avoir à considérer plusieurs centaines de propriétés. La caractérisation portée par une bi-partition est déjà intéressante mais elle ne permet pas de comprendre des associations ou interactions plus locales. Par exemple, le fait que deux propriétés soient souvent vraies à l'intérieur d'un bi-cluster, ne veut pas dire que ces deux propriétés sont souvent vraies dans les mêmes objets. Dans le cas de données médicales, cela pourrait indiquer deux symptômes qui ne se vérifient jamais ensemble même s'ils sont caractéristiques de patients malades. Dans [PB05b] puis son extension significative [PRB06b], nous avons présenté une méthode de caractérisation basée sur des bi-ensembles. Dans ce travail, nous partons d'une bi-partition calculée sur une table binaire et d'une collection de bi-ensembles extraites depuis cette même table, par exemple, des collections de concepts formels. Nous avons alors proposé d'associer chacun des bi-ensembles à l'un des bi-clusters de la bi-partition grâce à une mesure de similarité (prise en compte de l'intersection entre le bi-ensemble et le bi-cluster considéré comme un bi-ensemble). Dans une seconde phase, il a été possible de donner des mesures de la qualité de telles caractérisations. Ces mesures sont basées sur le nombre d'exceptions, c'est-à-dire le nombre d'éléments contenus dans le bi-ensemble mais qui ne sont pas dans le bi-cluster. Nous avons aussi relié ces caractérisations par bi-ensembles à des caractérisations par règles d'association avec des critères de fréquence et de confiance. Il devient alors possible de formuler de requêtes pour sélectionner tous les motifs qui caractérisent un certains bi-cluster avec des seuils de fréquence et de confiance donnés. Cette approche permet d'obtenir des résultats similaires à ceux décrits dans [RCB02]. Cependant, dans [RCB02], la caractérisation était obtenue au moyen d'un algorithme interactif et heuristique pour éliminer certaines règles candidates à la caractérisations. Dans notre approche, la sélection se fait

en utilisant de simples contraintes et se prête donc mieux aux approches de types "requêtes inductives".

Nous avons également analysé la caractérisation au moyen d'un autre type de bi-ensemble : les δ-bi-ensembles [PB05c]. Cette contribution s'inscrit dans les multiples tentatives de l'équipe pour identifier des extensions des concepts formels vers la tolérance aux exceptions [BRB04b, BRB05, BRB06, BPRB06]. En effet, dans des applications comme l'analyse des données d'expression, les données sont souvent bruitées et le nombre de concepts formels dans de telles données explose. Notre idée pour travailler sur les δ-bi-ensembles a été de remplacer l'opérateur de fermeture utilisé pour l'extraction de concepts formels par un opérateur admettant un nombre borné d'exceptions par colonne (opérateur de δ-fermeture). Cette notion avait été développée dans le contexte des représentations condensée d'ensembles fréquents par des motifs δ-libres [BBR03]. Un motif δ-libre est tel si il n'est pas inclus dans la δ-fermeture de ses sous-ensembles stricts. En associant à chaque ensemble δ-libre, sa δ-fermeture et son support (c'est-à-dire l'ensemble des objects pour lesquels les propriétés de l'ensemble δ-libre sont vérifiées), on obtient un nouveau type de bi-ensemble nommé δ-bi-ensemble. Par construction, le nombre d'exceptions sur de tels bi-ensembles est borné par colonne (valeur δ). Cette définition ne fixe pas de borne sur les exceptions par lignes, ne permet pas de préserver l'existence de fonctions et a fortiori celle d'une connection de Galois. Cependant, elle permet un calcul efficace de tous les δ-bi-ensembles par une adaptation simple de l'algorithme présenté dans [BBR03]. Nous avons également étudié certaines propriétés des δ-bi-ensembles [PRB06b] et nous avons participé à l'étude comparative [BPRB06] des différents types de bi-ensembles tolérants au bruit développés au LIRIS (e.g., les $\alpha\beta$-concepts [BRB04b] et les DR-bi-ensembles [BRB05, BRB06]).

Notre approche de la caractérisation a été évaluée empiriquement sur divers jeux de données (données "benchmark", données médicales et données d'expression [PRB06b]). La pertinence de notre méthode et le gain apporté par l'utilisation de bi-ensembles tolérant au bruit ont été mis en évidence. Nous l'avons également évalué lorsque l'algorithme de co-classification utilisé (e.g., Cocluster [DMM03]) donne des résultats très instables (i.e., des bi-partitions différentes lors de multiples exécutions). Ces résultats ouvrent de nouvelles pistes de recherche pour ce qui concerne l'évaluation d'une bi-partition via des motifs locaux mais aussi l'utilisation de notre technique de caractérisation pour la construction de descripteurs en classification supervisée. Ces directions de recherche se poursuivent dans le cadre du projet Européen IQ IST-FET FP6-516169.

Construction de bi-partitions

En parallèle et en complémentarité avec notre démarche de caractérisation, nous avons travaillé à la construction d'une bi-partition à partir d'une collection de motifs

locaux. Les méthodes classiques de construction de bi-partitions travaillent essentiellement au niveau d'instances. Cela veut dire que chaque objet et/ou propriété est traité individuellement. Par exemple, dans [DMM03, RF01b], on cherche à optimiser une partition alors que l'autre reste fixée et en considérant une fonction objectif commune aux deux partitions. Une fois la première partition réarrangée, on la fixe pour pouvoir optimiser l'autre (en considérant toujours la même fonction objectif). L'opération est répétée jusqu'à convergence. Dans certains cas, ce processus de modification alternée, peut se révéler inefficace. Ce sera notamment le cas lorsque la cardinalité des ensembles d'objets et propriétés est élevée, et qu'il est plus probable d'identifier un optimum local très distant du global. Nous avons donc mis au point une méthode de co-classification qui est basée sur le post-traitement de motifs locaux et plus particulièrement des bi-ensembles [PRB05]. L'idée appelée L2G (Local-To-Global), est que, l'information portée par des bi-ensembles (e.g., des concepts formels) peut être propagée pour la construction d'une bi-partition. Intuitivement, on peut imaginer que par fusions successives, les concepts formels peuvent conduire à des bi-clusters et donc à une bi-partition. L'un des concepts clés est celui de distance entre un bi-ensemble et un bi-cluster. Cette distance ressemble à celle calculée pour l'association de bi-ensembles lors d'une caractérisation mais elle a été simplifiée pour optimiser les calculs. Chaque bi-cluster est représenté par un centroïde dont les valeurs des coefficients dépendent du nombre de bi-ensembles affectés au bi-cluster. Ensuite, chaque objet/propriété est affecté à un bi-cluster selon sa valeur dans les centroïdes. Ceci illustre des différences fondamentales entre notre approche et les méthodes existantes. Tout d'abord, le nombre de bi-ensembles (éventuellement satisfaisant des contraintes) est central dans la phase de construction des bi-clusters. En effet ce que l'on traite, ce sont les interactions (sous forme de bi-ensembles) et non pas les éléments pris individuellement. Ensuite, il est facile d'admettre un certain niveau de chevauchement des bi-clusters (tout en gardant le recouvrement de la matrice), en jouant sur des seuils agissant sur les valeurs des coefficients.

Notre cadre est générique. Nous en avons décrit une instance particulière qui est une adaptation de l'algorithme K-Means. Notre algorithme CDK-Means, part d'une collection de bi-ensembles et du nombre K de bi-clusters désirés. Il calcule une bi-partition, éventuellement avec chevauchement. L'étape d'affinement des centroïdes se fait avec une approche de type *K-Means*, i.e., à partir d'une bi-partition tirée aléatoirement. Nous avons testé CDK-Means sur plusieurs jeux de données "benchmark". Il s'avère compétitif par rapport aux approches classiques de co-classification mais aussi vis-à-vis de certaines approches en classification mono-dimensionnelle. Les résultats où notre approche paraît plus performante concernent les contextes difficiles, i.e., des matrices relativement peu denses mais avec une forte dimensionalité. Nous avons aussi testé CDK-Means à partir de collections de δ-bi-ensembles. Notons d'ailleurs que l'utilisation de ces motifs tolérants au bruits a permis, dans certains cas, d'améliorer la qualité des bi-partitions tout en utilisant moins de ressources de calcul. Des données d'expression de gènes ont également été traitées. Nous avons ainsi pu

améliorer significativement la pertinence des bi-partitions obtenues avec COCLUSTER. En effet, nous avons retrouvé des bi-clusters qui mettaient en évidence des interactions validées par les connaissances biologique disponibles. COCLUSTER échouait alors que notre traitement des associations locales entre des gènes et les conditions où ces gènes sont sur-exprimés, préservait ces informations jusqu'au niveau global.

Notre méthode a ses propres limites. En particulier, l'utilisation d'une approche de type *K-Means* exige la déclaration préalable du nombre de bi-clusters. Dans certaines applications, ce nombre peut être difficile à identifier *a priori*. Cette limite est également valable pour à la méthode COCLUSTER. Une autre limite concerne la complexité de calcul lorsque le nombre de bi-ensembles considérés est important. Dans le cadre de l'une des applications, nous avons montré qu'en utilisant des contraintes de taille minimale sur les bi-ensembles, il était possible d'améliorer à la fois la vitesse de calcul et la qualité de la bi-partition. Cependant, ces contraintes ne sont pas toujours faciles à établir. De plus, lorsque l'on impose des contraintes de taille minimale, certains objets et/ou propriétés qui ne sont pas assez fréquents, peuvent donc disparaître de la bi-partition finale. La construction de modèles comme des bi-partitions à partir de motifs locaux se poursuit dans le cadre de l'ACI BINGO MD 46 et du consortium Européen IQ FP6-516169.

Bi-partitions sous contraintes

'A la recherche de primitives pour la co-classification, nous nous sommes intéressés aux contraintes qu'il faudrait savoir traiter pour que les analystes puissent spécifier déclarativement les propriétés des bi-partitions à calculer. Aller au delà de la seule contrainte d'optimisation d'une fonction objectif est une étape importante pour imaginer ce que pourraient être des requêtes inductives renvoyant des bi-partitions. Les requêtes inductives et donc les types de contraintes à utiliser lors des extractions de motifs locaux comme des ensembles ou des séquences ont été trés étudiées. Il n'en est pas de même pour des motifs globaux comme des bi-partitions ou des classifieurs. Nous avons effectué des travaux préliminaires pour la spécification et le calcul de bi-partitions sous contraintes [PRB06a, PRB06c]. Nous avons étendu les contraintes "must-link" et "cannot-link" introduites en classification mono-dimesnionnelle. Dans des données transcriptomiques, nous pouvons, par exemple, nous intéresser à une bi-partition où le gène g et la condition expérimentale c sont classé (ou non) dans le même bi-cluster. Nous avons aussi proposé un nouveau type de contrainte utile lorsque l'on possède des information ordonnées (ordre spatial ou temporel) sur au moins l'une des dimensions. Dans un tel cas, on peut s'intéresser à des bi-clusters où l'ensemble sur la dimension ordonnée est un intervalle bien formé (succession d'objets ou de propriétés "contigus"). Ainsi, dans des données d'expression cinétiques où les échantillons biologiques sont pris à des instants successifs, chercher des interactions de gènes qui prennent en compte cet aspect temporel améliore la pertinence

des bi-clusters et, par conséquent, des bi-ensembles caractérisant la bi-partition. On peut aussi considérer le problème inverse, c'est-à-dire le cas où des algorithme de bi-partitionnement classiques, ne retrouvent que des associations qui dépendent du temps. Nous proposons alors d'utiliser une contrainte qui force à trouver des bi-clusters indépendant de l'ordre.

Une fois que l'on a identifié des contraintes pertinentes dans le sens où elles permettent de capturer l'intérêt subjectif de l'analyste, il faut aussi pouvoir les exploiter au cours des calculs. Notre hypothèse de travail est que, des contraintes définies sur des bi-ensembles locaux peuvent se propager jusqu'au niveau global (la bi-partition), quitte à ce que cela nécessite une stratégie de propagation. Nous avons donc présenté un modèle basé sur notre cadre L2G, où, pour une contrainte globale donnée, on définit d'abord sa version locale (éventuellement relaxée ou renforcée) pour ensuite rechercher à la propager pendant l'exécution d'un algorithme CDK-Means. Notre but est bien sûr que la bi-partition finale satisfasse au mieux toutes les contraintes. Notons d'ailleurs que la contrainte d'optimisation de la fonction objectif qui est implicite à toute co-classification n'est jamais elle-même garantie. A ce jour, nous avons obtenu de bons résultats pour les contraintes d'intervalle sans propagation. En particulier nous avons pu constater que, lorsque les résultats sont satisfaisant sans l'utilisation de contraintes, leurs définitions permettent de n'utiliser que les bi-ensembles qui sont pertinent par rapport à la contrainte : on réduit donc les calculs tout en préservant les résultats. Lorsque les résultats sans contraintes ne sont pas satisfaisants, nous avons constaté que la définition des contraintes d'intervalle améliorait la qualité des bi-partitions obtenues. Pour ce qui concerne les autres contraintes nous n'avons que des résultats très préliminaires. Ce travail est donc clairement incomplet mais il a ouvert des pistes de recherche très intéressantes, une fois encore dans le cadre dans le cadre de l'ACI BINGO MD 46 et du consortium Européen IQ FP6-516169.

Scénarios d'extraction de connaissances

Les contributions méthodologiques et algorithmiques citées jusqu'ici ont été motivées par nos collaborations avec des biologistes ayant des besoins en analyse du transcriptome. La mise en oeuvre d'algorithmes de fouille de données booléennes à partir des données brutes issues des méthodes SAGE ou des Puces ADN (cf. [Bes05]) n'est pas triviale : il faut d'abord encoder des propriétés booléennes concernant l'expression des gènes (e.g., la sur-expression) dans les conditions expérimentales. Pour aller au delà des propositions simples décrites dans [BBJ+02], nous avons développé des méthodes de pré-traitement de données d'expression afin de déterminer précisément la méthode de discrétisation à employer [PLBB04, PB05a]. Notre idée est de sélectionner parmi des méthodes de discrétisation paramétrées celle qui conserve au mieux les structures globales présentes dans les données numériques. Plus précisément, nous appliquons un clustering hiérarchique et sur les données brutes (données numériques

avant discrétisation) et sur les différents jeux de données discrétisées. La discrétisation retenue est celle dont le dendrogramme (représentation sous forme d'arbre du résultat du clustering hiérarchique) se rapproche le plus du dendrogramme obtenu à partir des données réelles.

Toujours dans le cadre de scénarios réels basés sur de grandes collections de motifs locaux, il faut assister les experts qui doivent explorer ces grands volumes de motifs. Pour celà, nous avons développé une technique de manipulation et de visualisation de collections de concepts formels [Pen03, RPBB04]. Cet outil réalise un clustering hiérarchique non pas sur les gènes ou les conditions expérimentales mais sur les concepts formels. C'est un outil typique de post-traitement pour faciliter le travail d'interprétation du biologiste. Cet outil exploite d'ailleurs la grande familiarité de ces biologistes avec les outils de clustering hiérarchique. La pertinence de notre outil a été validée dans le cadre d'une coopération avec le laboratoire CGMC concernant l'analyse de données SAGE humaines [BPB$^+$06].

Les développements des outils de pré-traitement, d'extraction de motifs et de post-traitement dédiés aux données d'expression de gènes ont demandé des validations expérimentales plus ou moins poussées. D'autre part, nous avons aussi contribué à des applications réelles comme [BPB$^+$06]. Il était intéressant de capitaliser ce savoir-faire dans la formalisation de scénarios prototypiques d'extraction de connaissances pour l'analyse du transcriptome [PBB04, PBRB05]. Les scénarios proposés abordent les problèmes de l'encodage de propriétés, de l'enrichissement de données (utilisation d'autres sources d'information), et de l'utilisation des contraintes pour répondre à des questions biologiques précises. Nous avons aussi étudié les possibilités d'enrichissement à chaque itération d'un processus d'extraction afin de dynamiquement focaliser l'analyse [PBB04]. Ces scénarios prototypiques sont réellement des abstractions des différentes analyses anxquelles nous avons participé. Dans ce mémoire nous présentons un nouveau scénario qui met en œuvre le cadre L2G sur un jeu de données Puces à ADN concernant Plasmodium Falciparum. Cette application met en évidence les avantages de l'approche de la co-classification sous contraintes associé à une caractérisation par des motifs locaux. Il ne s'agit pas d'une application de découverte biologique mais plutôt de redécouverte : elle nous permet de valider notre approche en retrouvant des résultats de la littérature non triviaux. En particulier, grâce à la définition d'une contrainte d'intervalle nous avons retrouvé les trois étapes du développement du plasmodium, et l'application de la méthode de caractérisation au bi-cluster correspondent à la phase "anneau", nous à permis d'identifier un certain nombre de gènes connus pour leur activité dans cette phase du cycle de vie (appartenance au groupe nommé *cytoplasmic translation machinery*).

Notons enfin que nous avons contribué au développement du logiciel d'extraction de connaissances Bio++ [BPB$^+$04]. Ce logiciel intègre les extracteurs et les méthodes présentés précédemment et a été l'un des livrables de fin du contrat Européen cInQ (IST-2000-26469).

Nous résumons maintenant la structure du reste du mémoire. Il est structuré en trois parties. La première (Chapitre 1) présente le cadre de la recherche de groupements de qualité, puis les méthodes de classification et de co-classification. En identifiant certains problèmes restés ouverts, ce chapitre motive donc notre démarche de création du cadre L2G. La seconde partie est entièrement consacrée à nos contributions méthodologiques et algorithmiques, c'est-à-dire, à la caractérisation de bi-partitions (Chapitre 2), à la construction de bi-partitions à partir de motifs locaux (Chapitre 3), et enfin à la construction de bi-partitions sous contraintes (Chapitre 4). Dans la troisième partie, nous présentons nos développements dédiés à l'anayse du transcriptome. Tout d'abord nous présentons nos contributions à l'évaluation des méthodes de discrétisation et à la visualisation de motifs locaux par post-traitement (Chapitre 5). Ensuite, nous formalisons un scénarios d'extraction de connaissances à partir de données Puces ADN (Chapitre 6). Enfin, nous donnons une brève conclusion avant de considérer quelques perspectives de cette recherche.

Première partie

État de l'art

Chapitre 1

État de l'art en co-classification

1.1 Le problème de la recherche des groupements de qualité

Les méthodes d'extraction de connaissances et d'apprentissage automatique ont pour objectif de structurer un ensemble d'objets (ou individus), décrits par des caractéristiques (attributs), sous une forme induite par la méthode utilisée. Pour les méthodes dites supervisées, une ou plusieurs étiquettes sont associées aux objets, et le but est alors de constuire un modèle (classifieur) à partir des caractéristiques des objets. Ce modèle peut ensuite être utilisé pour prédire la valeurs des étiquettes pour de nouveaux objets.

Pour les méthodes dites non supervisées, l'objectif est de mettre en évidence des relations entre des objets en les regroupant en entités homogènes (classes) suivant une certaine mesure de similarité évaluée à partir des valeurs des attributs pour ces objets.

Un troisième groupe de méthodes plus récente, est celui des approches dites semi-supervisées, dont le principe consiste à combiner les avantages des approches supervisées et non supervisées. Deux approches existent, selon que l'on cherche à réaliser une tâche de prédiction ou de description de données :

- Dans la première approche, des méthodes de type supervisé sont adaptées de telle sorte à pouvoir utiliser des objets non étiquetés lors de la construction du modèle prédictif.
- Dans la seconde approche, les objets sont regroupés en utilisant une méthode non supervisée biaisée de telle sorte à produire une partition aussi pure que possible par rapport à la distribution de la variable de classe sur les objets étiquetés. La structuration produite a ici un caractère descriptif.

Dans ce chapitre nous allons présenter essentiellement les méthodes non su-

pervisées ainsi que certaines méthodes semi-supervisées basées sur l'utilisation de contraintes. La classification non supervisée est un domaine de recherche qui date déjà de plusieurs décennies, mais qui est en constante évolution, notamment pour répondre aux besoins en termes de passage à l'échelle, et pour le traitement des données complexes et structurées. Nous parlons de classification unidimensionnelle lorsqu'il s'agit de partitionner l'ensemble des objets en fonction de la similarité de leurs caractéristiques, et de co-classification lorsqu'en plus de partitionner les objets, la méthode partitionne simultanément les valeurs des attributs. Les méthodes de co-classification sont relativement récentes[1]. La co-classification peut être considérée comme une approche de classification conceptuelle, qui rassemble toutes les méthodes qui intègrent un processus de construction d'une interprétation au processus de construction de la partition. Parmi ces méthodes, l'analyse des concepts formels, connue déjà depuis deux décennies, peut être considérée comme l'ancêtre de la co-classification.

Il nous faut encore distinguer les méthode non supervisées qui produisent une bi-partition de l'ensemble des données (ou du moins qui structurent celui-ci de manière globale) des méthodes qui calculent des motifs locaux, sous la forme de couples (bi-ensembles) composés d'un ensemble d'objets et d'un ensemble de valeurs d'attributs fortement associés localement dans les données. En réalité cette distinction n'est pas aussi nette dans la littérature et le mot anglais "bi-clustering", en particulier, a été utilisé pour désigner les deux approches.

L'élément central de cette discussion est donc la classe. La définition de classe "naturelle" est un problème fondamental en classification. Différentes façons de considérer une classe peuvent être envisagées. B. Everitt dans [JD88] en propose trois :

- "Une classe contient des éléments semblables, et les éléments appartenant à différentes classes sont différents"
- "Une classe est une agrégation d'éléments dans l'espace de description, telle que la distance entre tout couple d'éléments soit inférieure à la distance entre tout élément de la classe et tout élément hors de la classe"
- "Une classe peut être décrite comme une région connexe d'un espace multi-dimensionnel contenant une densité d'éléments relativement haute, et séparée d'autres régions du même type par des régions de densité relativement basse"

Non seulement la notion de classe peut recouvrir des concepts différents, mais de plus leur traduction sous forme de critères mathématiques opérationnels est également non unique. Cela est d'autant plus vrai dans le cadre de la co-classification. La définition plus générique que l'on peut donner à la notion de bi-cluster est celle d'un ensemble d'objets et d'attributs qui satisfait une caractéristique spécifique d'ho-

[1]La première méthode de ce type a été produite au milieu des années 70 par Govaert, mais il faut attendre le début des années 2000 pour que cette problématique connaisse un regain d'intérêt, se traduisant à la fois par la production de différentes méthodes, et également par leur utilisation, notamment pour le traitement de données post-génomiques.

mogénéité [MO04]. La définition de classe et de bi-cluster dépend non seulement du type d'association que l'on cherche à capturer, mais aussi, d'une manière plutôt forte du type de données que l'on traite.

Dans la suite nous allons présenter des méthodes permettant de partitionner un ensemble de m éléments, appelés objets, que l'on notera $\mathcal{T} = \{t_1, \ldots, t_m\}$. Chacun de ces objets est décrit par n attributs notés $\mathcal{A} = \{a_1, \ldots, a_n\}$. Chaque attribut définit une application :

$$\begin{array}{rcl} a_j & : & \mathcal{T} \rightarrow dom_j \\ & & t_i \rightarrow a_j\,(t_i) \end{array}$$

où dom_j est appelé domaine d'observation de l'attribut a_j et est :

- un intervalle de $\mathbb{R}$ si a_j est un attribut numérique
- un ensemble discret ordonné de valeurs si a_j est un attribut ordinal
- un ensemble discret non ordonné de valeurs appelées modalités si a_j est un attribut nominal

L'ensemble de ces objets définit un tableau $\mathcal{T} \times \mathcal{A}$, où chaque ligne représente un objet, et chaque colonne un attribut qui décrit ces objets (voir tableau 1.1).

$\mathcal{T}\vert\mathcal{A}$	a_1	$\ldots$	a_j	$\ldots$	a_n
t_1	$a_1\,(t_1)$	$\ldots$	$a_j\,(t_1)$	$\ldots$	$a_n\,(t_1)$
$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
t_i	$a_1\,(t_i)$	$\ldots$	$a_j\,(t_i)$	$\ldots$	$a_n\,(t_i)$
$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
t_m	$a_1\,(t_m)$	$\ldots$	$a_j\,(t_m)$	$\ldots$	$a_n\,(t_m)$

TAB. 1.1 – Un tableau de données objets×attributs

Un exemple de tableau de données est donné en Figure 1.1a. Dans ce chapitre, même si nous allons traiter les différentes méthodes de classification de façon générique, nous nous intéresserons en particulier aux attributs de type booléen, c'est-à-dire aux attributs nominaux à deux valeurs "0" (faux) ou "1" (vrai). Même si cela peut paraître une limitation, chaque attribut peut être discrétisé en un ou plusieurs attributs booléens. Par exemple, l'attribut numérique a_2 dans la table en Figure 1.1a a été discrétisé en deux intervalles : lorsque $a_2 \leq 10$ l'attribut booléen g_2 (cf. Fig. 1.1b) est à "1". Dans le cas contraire ($a_2 > 10$), c'est l'attribut booléen g_3 qui est à "1". De la même manière, l'attribut booléen g_4 code la paire attribut-valeur (a_3, a), alors

que g_5 code la paire attribut-valeur (a_3, b). Un tableau booléen peut aussi être appelé contexte booléen ou relation binaire. Dans la suite du manuscrit, nous noterons $\mathbf{r} \subseteq \mathcal{T} \times \mathcal{G}$ un contexte booléen où $\mathcal{G}$ est l'ensemble des attributs booléens, et $r_{ij} = 1$ si l'attribut booléen g_j est vrai pour l'objet t_i, $r_{ij} = 0$ sinon.

$\mathcal{T}\|\mathcal{A}$	a_1	a_2	a_3
t_1	1	17.5	a
t_2	0	5.4	b
t_3	1	19.1	a
t_4	0	21.0	a
t_5	1	9.8	b
t_6	0	5.4	b
t_7	0	-	b

(a)

$\mathcal{T}\|\mathcal{G}$	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

(b)

FIG. 1.1 – Un tableau générique de données (a) et une possible discrétisation (b)

La partition de l'ensemble $\mathcal{T}$ que l'on cherche est composée de K classes et on la note $\mathcal{P}^{\mathcal{T}} = \{P_1^{\mathcal{T}}, \ldots, P_k^{\mathcal{T}}, \ldots, P_K^{\mathcal{T}}\}$. La partition de l'ensemble d'attributs $\mathcal{A}$ (composée de L classes) est $\mathcal{P}^{\mathcal{A}} = \{P_1^{\mathcal{A}}, \ldots, P_l^{\mathcal{A}}, \ldots, P_L^{\mathcal{A}}\}$. Une bi-partition est donc notée $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{A}})$. Un bi-cluster est un bi-ensemble que l'on notera (T_i, A_i), où $T_i \subseteq \mathcal{T}$ et $A_i \subseteq \mathcal{A}$. Un bi-cluster (T_i, A_i) est associé à une bi-partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{A}})$ si $T_i \in \mathcal{P}^{\mathcal{T}}$ et $A_i \in \mathcal{P}^{\mathcal{A}}$. Selon les méthodes de (co-)classification, le nombre de classes K est, soit fixé par l'utilisateur, soit auto-déterminé par la méthode. Par ailleurs, selon la méthode, les classes appartenant à une même partition peuvent être disjointes (on parle alors de "hard clustering"), ou elles peuvent avoir un certain degré de recouvrement (on parle alors de "soft clustering").

Nous présentons dans ce chapitre différentes méthodes de classification que nous regroupons en deux catégories majeures : les méthodes unidimensionnelles et les méthodes de co-classification. Après avoir présenté les méthodes les plus importantes en classification unidimensionnelle, nous présenterons des méthodes de classification sous contraintes, qui permettent de calculer des partitions qui satisfont un certain nombre de contraintes impliquant les objets. Ces méthodes ont été développées pour résoudre le problème de la classification semi-supervisée. Cependant, les approches semi-supervisées sont limitées à l'utilisation d'un petit nombre de contraintes qui ne sont pas forcément intéressantes dans une application non supervisée. Ensuite, nous présenterons en détail les méthodes de co-classification en insistant sur les différences entre les approches basées sur les données numériques (telles que les données d'expression), les approches basées sur les tables de co-occurrences (comme celles des données termes-documents), et les approches pour les données binaires. La distinction n'est pas seulement liée à la structure des données, mais aussi au type de bi-clusters que

l'on obtient. La co-classification étant une méthode de classification conceptuelle, nous allons en présenter aussi quelques approches classiques et plus particulièrement, l'analyse formelle des concepts. Les concepts formels sont en effet des bi-clusters maximaux de "1" dans des données booléennes. Une de ces techniques permet d'extraire efficacement des concepts formels satisfaisant des contraintes définies par l'utilisateur, et il est donc possible de piloter l'extraction vers une sous-collection de motifs pertinents, ce qui n'est pas possible avec les méthodes de (co-)classification classiques. En revanche, la taille de la collection de concepts atteint facilement des dimensions énormes et inexploitables par l'utilisateur final. De plus, on perd toute notion de bi-partition, c'est-à-dire de structuration globale des données qui, dans beaucoup d'applications, est importante. Nous ferons le bilan des avantages et des limites des méthodes présentées et nous motiverons les raisons qui nous ont amené à proposer une nouvelle méthode de co-classification pour répondre aux vrais besoins de l'utilisateur.

1.2 Classification unidimensionnelle et Co-classification

1.2.1 Les méthodes de classification unidimensionnelle

Les méthodes de classification unidimensionnelle ont été principalement proposées pour traiter des données numériques. Chaque objet est alors considéré comme un point de l'espace $\mathbb{R}^n$. Cet espace est généralement muni d'une métrique euclidienne d^2_M telle que :

$$d^2_M(t_i, t_j) = (t_i - t_j)^t M (t_i - t_j)$$

où t_i et $t_j \in \mathcal{T}$, et M est une matrice carrée de dimension $(n \times n)$, symétrique définie positive. Ainsi, d^2_I, où I est la matrice identité, est la distance euclidienne.

Méthodes hiérarchiques

La classification hiérarchique [JD88] construit une hiérarchie de classes, ou, en d'autres termes, un arbre de classes (souvent nommé *dendrogramme*). Chaque nœud de l'arbre contient les clusters fils, qui partitionnent les objets couverts par le parent commun. Une telle approche permet d'explorer les données à différents niveaux de granularité. Il existe deux principales approches en classification hiérarchique, l'approche *agglomérative* (bottom-up) et l'approche *divisive* (top-down). La classification agglomérative commence avec des classes composées d'un seul objet (singleton) et fusionne récursivement les deux classes les plus appropriées. Une approche divisive commence avec une seule classe contenant tous les objets et divise récursivement la classe la plus appropriée. Le processus continue tant que le critère d'arrêt n'est pas vérifié (e.g., le nombre K de classes).

Le critère selon lequel l'algorithme fusionne/divise les classes dépend de la (dis)similarité entre les éléments et les classes. La présomption générale est alors que les classes sont composées d'objets similaires. On utilise généralement la distance euclidienne comme mesure de la dissimilarité entre objets.

L'entité qui est la plus souvent utilisée par les algorithmes de classification hiérarchique est la matrice des distances. Une matrice des distances $\mathbf{d}$ (ou des similarités) est une matrice $m \times m$ (appelée parfois matrice de connectivité), où chaque case $d_{ij} \in \mathbf{d}$ contient la distance entre l'objet t_i et l'objet t_j.

Pour fusionner ou diviser des sous-ensembles d'objets, la distance entre deux objets doit être généralisée à une distance entre sous-classes. Cette mesure de proximité dérivée est appelée *lien* (linkage). Le type de lien utilisé influence significativement les algorithmes de classification hiérarchique. Parmi les liens le plus utilisés on trouve le lien simple, le lien moyen, et le lien complet. La mesure de dissimilarité (distance) est calculée à partir des distances entre les objets de la première classe et ceux de la deuxième. Selon la métrique utilisée, on considère le minimum (lien simple), la moyenne (lien moyen), ou le maximum (lien complet) de toutes les distances calculées.

$$d_{simple}(P_1, P_2) = \min\{d(t_i, t_j) | t_i \in P_1, t_j \in P_2\}$$
$$d_{moyen}(P_1, P_2) = moyenne\{d(t_i, t_j) | t_i \in P_1, t_j \in P_2\}$$
$$d_{complet}(P_1, P_2) = \max\{d(t_i, t_j) | t_i \in P_1, t_j \in P_2\}$$

Toutes ces métriques peuvent être dérivées en tant qu'instances de la formule de mise à jour de Lance-Williams [LW67]

$$d(P_i \cup P_j, P_k) = \alpha_i \cdot d(P_i, P_k) + \alpha_j \cdot d(P_j, P_k) + \beta \cdot d(P_i, P_j) + \gamma \cdot |d(P_i, P_k) - d(P_j, P_k)|$$

où α, β, et γ sont des coefficients correspondant à un lien particulier. Cette formule exprime la métrique de lien entre l'union des deux classes et la troisième classe en termes de composantes sous-jacentes et permet la faisabilité du calcul des liens.

L'avantage principal des méthodes hiérarchiques est la flexibilité concernant le niveau de granularité (et donc le nombre K de classes). En revanche, le choix de la fusion (ou division) appropriée, est fait selon un critère d'optimisation totalement local, qui dans le cas du lien simple conduit à la partition optimale au sens du critère optimisé.

Méthodes partitionnelles

Le principe des méthodes partitionnelles est de diviser les données en plusieurs sous-ensembles. Comme le calcul de tous les sous-ensembles possibles n'est pas faisable, on utilise des heuristiques sous la forme d'optimisation itérative de la fonction objectif. Chaque algorithme possède un schéma de relocalisation qui réassigne

itérativement les objets parmi les K classes. À la différence des approches hiérarchiques, où les classes ne sont pas revisitées après avoir été construites, les algorithmes partitionnels améliorent graduellement la qualité des classes.

Une première famille de méthodes consiste à considérer les caractéristiques des objets comme des variables aléatoires, qui suivent une certaine loi de probabilité, et l'objectif des algorithmes consiste à evaluer les paramètres de ces distributions.

Une seconde famille de méthodes utilise une fonction objectif qui évalue la qualité d'une partition en termes de similarité intra-classe et dissimilarité inter-classe. On peut utiliser les mêmes mesures de distance que celles de la classification hiérarchique, mais dans le cas des algorithmes partitionnels, le calcul de ces distances devient trop coûteux. L'utilisation d'un seul représentant par classe résout ce problème, car le calcul de la fonction objectif devient linéaire en m (et en $K << m$). Selon la façon dont les représentants sont construits, on distingue deux méthodes : *k-medoids* et *k-means*. Un médoïde est l'objet le plus similaire à tous les autres objets de la classe qu'il représente. Dans la méthode du *k-means*, une classe est représentée par son centroïde, qui est une moyenne (souvent pondérée) des objets appartenant à cette classe.

Les méthodes probabilistes Soit $\mathcal{E}$ une population d'où est extrait l'échantillon $\mathcal{T}$. T est une variable aléatoire qui, à chaque élément de $\mathcal{E}$, associe un vecteur de $\mathbb{R}^n$, et P est une variable aléatoire qui, à chaque élément de $\mathcal{E}$, associe une classe de la partition $\mathcal{P}$. L'objectif est alors de déterminer les distributions conditionnelles de chacune des variables aléatoires par rapport à l'autre. Pour ce faire, on est amené à réduire l'ensemble des modélisations possibles du problème. On fait alors l'hypothèse que $\mathcal{P} = \{P_1, \ldots, P_K\}$ et que, conditionnellement à la valeur sur P, les observations sont distribuées selon des lois de densité $f_k(t) = Pr(t|P = P_k)$. Ainsi, on considère que les objets sont autant d'observations issues d'un mélange de densités de probabilité (*mixture model*) $f_1, \ldots, f_K$ que l'on cherche à identifier. Au lieu d'assigner chaque objet à une classe, on lui associe une probabilité d'appartenance.

Le modèle probabiliste permet d'expliquer les variations entre les objets d'une même classe. Dans cette approche, on fait l'hypothèse de l'existence d'une distribution de probabilité des caractéristiques décrivant les objets d'une même classe. Le principe général est alors de décomposer une distribution multimodale en un certain nombre de distributions unimodales [Wol70].

Lorsque l'on fait l'hypothèse simplificatrice que la forme des distributions est fixée, la maximisation de la fonction de vraisemblance permet d'estimer les paramètres inconnus de la densité. Soit p_k $(k = 1 \ldots K)$, la proportion d'objets qui suivent la loi de densité parametrée $f_k(t)$. La vraisemblance totale des données est leur probabilité

d'être issues d'un tirage d'un mélange de distributions donné :

$$L(T|P) = \prod_{i=1}^{m} \sum_{k=1}^{K} p_k Pr(t_i|P_k)$$

L'estimation des paramètres par le maximum de la vraisemblance consiste alors à déterminer les paramètres qui maximisent L. Pour des commodités calculatoires, on maximise $\ln(L(T|P))$, appelé log-vraisemblance de l'échantillon. La log-vraisemblance est la fonction objectif optimisée par la méthode *Expectation-Maximization* (EM), présentée dans [DLR77]. L'algorithme EM comporte une optimisation en deux phases. La phase E estime les probabilités $Pr(t|P)$, tandis que la phase M cherche une approximation d'un modèle de mélange à partir des probabilités estimées dans la phase précédente. Cela consiste à trouver les paramètres d'un modèle de mélange qui maximise la log-vraisemblance. Le processus continue tant que la convergence de la fonction objectif n'est pas atteinte.

La méthode des *K-Medoids* Dans la méthode des k-médoïdes, une classe est représentée par un ou plusieurs objets parmi les plus représentatifs de la classe. Cette solution est adaptée à tout type d'attribut et elle est robuste vis-à-vis des outliers, car les points périphériques d'une classe n'ont aucune influence sur son médoïde. Une fois les médoïdes sélectionnés, les classes sont définies comme des sous-ensembles d'objets proches de leurs médoïdes respectifs, et la fonction objectif est définie comme la distance moyenne (ou une autre mesure de dissimilarité) entre un objet et son médoïde. Parmi les algorithmes basées sur cette méthode, on trouve PAM (Partitioning Aroung Medoids) et CLARA (Clustering LARge Applications) [KR90]. PAM est un algorithme d'optimisation itératif qui, à chaque pas, remplace un des médoïdes par un objet non médoïde si cela permet d'améliorer la fonction objectif. Cet algorithme est très coûteux en pratique, et les auteurs en ont proposé une version plus efficace (CLARA) qui utilise plusieurs échantillons, chacun composé de $40 + 2K$ objets, qui sont traités avec l'algorithme PAM.

Une amélioration ultérieure a été introduite par [NH94] avec l'algorithme CLARANS (Clustering Large Applications based upon RANdomized Search), dans le cadre de la classification dans les bases de données spatiales. CLARANS utilise la recherche aléatoire pour générer le voisinage de chaque médoïde. Si un voisin représente une partition meilleure, il est pris comme nouveau médoïde, sinon un minimum local a été trouvé, et l'algorithme recommence tant qu'un nombre de minima locaux défini par l'utilisateur n'a pas été trouvé.

La méthode des *K-Means* L'algorithme des k-means [Mac67], est de loin l'outil le plus populaire utilisé dans les applications scientifiques et industrielles de classification non supervisée. Le nom dérive du fait que, pour représenter chacune des K

classes P_k, on utilise la moyenne (ou la moyenne pondérée) π_k de ses points, appelée centroïde (ou centre de masse). Chacune des n composantes du vecteur π_k est calculée par :

$$\pi_{jk} = \frac{1}{|P_k|} \sum_{t_i \in P_k} a_j(t_i)$$

Dans le cas de données numériques, cela donne un sens géométrique et statistique à la méthode. L'inertie intra-classe constitue le critère à optimiser. Elle est définie comme la moyenne des carrés des distances des objets de la classe au centre de gravité de celle-ci. On cherche ainsi à construire des classes compactes. L'inertie intra-classe associée à la classe P_k s'écrit formellement

$$I_k = \frac{1}{|P_k|} \sum_{t_i \in P_k} d^2(t_i, \pi_k)$$

L'objectif est alors de minimiser la somme de l'inertie intra-classe sur l'ensemble des classes. Ils existent deux versions majeures de la méthode des k-means. La première est similaire à l'algorithme EM, et procède en deux étapes : dans la première phase, on réassigne tous les objets au centroïde le plus proche, et dans la deuxième phase, on recalcule les centroïdes des classes qui ont été modifiées. Les deux phases sont itérativement répétées jusqu'à ce qu'un critère d'arrêt soit atteint (par exemple, si aucune modification n'a eu lieu, ou si le nombre maximum d'itérations a été atteint). La deuxième version est basée sur une analyse plus détaillée concernant les effets sur la fonction objectif causés par le déplacement d'un objet d'une classe vers une autre. Si le changement a un effet positif, l'objet est déplacé et les deux centroïdes concernés sont recalculés.

Les principaux problèmes de l'approche des $k - means$ comme des autres approches partitionnelles, sont l'influence de la partition initiale (qui est souvent choisie de façon aléatoire), et le choix du paramètre K qui n'est pas toujours évident.

Méthodes basées sur les itemsets

Nous avons vu des approches de la classification basées essentiellement sur des données numériques. Le cadre des données catégorielles a été souvent traité sous la forme de données transactionnelles où chaque objet est une transaction, qui est un ensemble fini d'éléments appelés items. Par exemple, les données concernant le panier d'un supermarché ont cette forme, car chaque transaction contient la liste des produits achetés par un client à un moment donné. Si on considère le contexte booléen $\mathbf{r}$ en Fig. 1.1b, on dit qu'un item g_j appartient à une transaction t_i si $r_{ij} = 1$. On peut donc représenter chaque transaction comme un ensemble d'items (cf. Table 1.2).

Le problème majeur avec les données transactionnelles, est que la classification devient une tâche difficile lorsque la taille de l'univers des items croît. Une solution

$\mathcal{T}$	Itemsets
t_1	$\{g_1, g_3, g_4\}$
t_2	$\{g_2, g_5\}$
t_3	$\{g_1, g_3, g_4\}$
t_4	$\{g_3, g_4\}$
t_5	$\{g_1, g_2, g_5\}$
t_6	$\{g_2, g_5\}$
t_7	$\{g_5\}$

FIG. 1.2 – Un tableau de données transactionnelles

est donc de classifier d'abord l'ensemble des items, et ensuite classifier les données. Le problème central devient ainsi la recherche de groupes d'items. Une des premières approches proposées a été celle de [HKKM97]. Les auteurs proposent une méthode pour partitionner l'ensemble des items, puis ils calculent la partition des transactions en assignant chaque transaction t_i à la classe d'items P_k pour laquelle la fonction $|T_i \cap P_k|/|P_k|$ est maximale (T_i étant l'ensemble d'items contenus dans la transaction t_i). Pour atteindre ce but, les auteurs utilisent les règles d'association et les hypergraphes. Premièrement, on génère les itemsets fréquents contenus dans l'ensemble des transactions. On associe ensuite un hyper-graphe $H = (V, E)$ à l'univers des items, tel que les sommets V sont les items. Dans un graphe commun, des paires de sommets sont connectés par des arêtes, tandis que dans un hyper-graphe, plusieurs sommets sont connectés par des hyper-arêtes. Une hyper-arête $e \in E$ dans H correspond à un itemset fréquent $\{v_1, \ldots, v_s\} \in V$ et a un poids correspondant à la moyenne des confiances calculées sur toutes les règles d'association impliquant cet itemset. Pour partitionner l'hyper-graphe, les auteurs utilisent HMETIS [KAKS97], qui est un algorithme de paritionnement multi-niveaux, et qui produit une partition équilibrée de K groupes de sommets.

Une autre approche a été proposée par [WXL99]. Les auteurs proposent une mesure de similarité basée sur la notion de larges items. Un item est large dans une classe de transaction, s'il est suffisamment fréquent à l'intérieur de la classe. Le critère qu'ils optimisent correspond à l'hypothèse qu'une bonne classification est telle qu'il y a beaucoup d'items larges à l'intérieur d'une classe, et qu'il y a peu de chevauchements de ces items entre les classes. Si on appelle $Small_k$ l'ensemble d'items non fréquents dans la classe P_k et $Large_k$ l'ensemble d'items fréquents dans P_k, la méthode cherche à minimiser la fonction objectif :

$$Cost(\mathcal{P}) = |\cup_{k=1}^{K} Small_k| + \Sigma_{k=1}^{K} |Large_k| - |\cup_{k=1}^{K} Large_k|$$

L'algorithme assigne initialement chaque transaction (dans l'ordre d'apparition dans la base) à une nouvelle classe ou à une classe existante afin de minimiser la fonction $Cost(\mathcal{P})$. Ensuite il affine les classes en réassignant chaque transaction (prise toujours

dans le même ordre) à une classe qui contient au moins deux transactions. Ainsi, les classes contenant un seul élément sont susceptibles d'être éliminées. Le nombre K de classes n'est donc pas fixé a priori. En revanche, il dépend de l'ordre dans lequel les transactions sont considérées. La phase de minimisation de la fonction objectif étant très coûteuse, les auteurs proposent une méthode pour la mise à jour de ses différentes composantes, par le moyen d'une table de hash, et d'un B-arbre.

D'autres méthodes ont été proposées, comme dans [FWE03], dont le but est de construire une hiérarchie de documents, en se basant sur une classification à base d'itemsets fréquents.

Dans ces méthodes, un problème central est la notion d'itemsets fréquents. L'utilisateur doit être en mesure de pouvoir choisir un seuil de fréquence pour l'extraction des itemsets, qui est une étape préalable dans toutes les approches que nous avons présentées. Or, ce choix n'est pas toujours évident, notamment dans certains jeux de données particulièrement bruités.

Autres approches

Dans cette section nous avons présenté un ensemble de techniques de classification non supervisée. Ce domaine de la fouille de données et de l'apprentissage automatique, a fait l'objet d'un très grand nombre de travaux. Parmi les méthodes les plus intéressantes, on trouve les approches basées sur la densité [EKSX96], qui permettent de découvrir de classes de forme arbitraire (tandis que les approches classiques trouvent plutôt des classes sphériques). Dans ces approches, le concept crucial est la notion de voisinage et de connectivité. La complexité de calcul du voisinage est telle que la méthode n'est efficace que dans les données spatiales à peu de dimensions.

Une autre catégorie de méthodes est basée sur les grilles [SE97, WYM97], et consiste à diviser l'espace en unités (segments, cubes, cellules ou régions) qui composent une grille. La partition des données et induite par l'appartenance d'un objet à une unité issue de la partition de l'espace.

Enfin, parmi les méthodes plus populaires, on trouve aussi celles basées sur les réseaux de neurones, en particulier les SOM (Self-Organized Map), ou cartes de Kohonen [RK89], où le principe est de plonger les centroïdes dans un espace à deux dimensions. Le principe est similaire à celui des *k-means*, mais les objets sont traités un par un, et on modifie de façon incrémentale les vecteurs de référence, et, par le biais d'une fonction noyau, les vecteurs voisins.

1.2.2 Méthodes de classification sous contraintes

Dans la section précédente nous avons vu des approches d'extraction de motifs sous contraintes. Ces approches permetent à l'utilisateur d'extraire seulement les motifs, (e.g., les concepts formels) qui satisfont un certain nombre de contraintes. Il serait intéressant de pouvoir bénéficier d'un certain contrôle même lors de l'utilisation d'un algorithme de classification. Dans cette section nous allons explorer l'approche dite de classification semi-supervisé, et en particulier les méthodes qui sont basées sur la spécification de contraintes pour la construction d'une partition.

Le problème de la classification semi-supervisée

Il est clair qu'il existe un lien très étroit entre classification supervisée et classification non supervisée. Bien que les deux techniques soient souvent utilisées dans des contextes différents (l'un en prédiction, l'autre en analyse), la classification non supervisée est de plus en plus utilisée pour améliorer les performances des classifieurs. Cela est évident quand on considère les approches de classification semi-supervisée qui ont vu le jour dans ces cinq dernières années [WC00].

Le terme "classification semi-supervisée" indique un ensemble de techniques parfois très différentes qui ont pour objectif de résoudre un problème très courant en classification supervisée. L'hypothèse d'être toujours en mesure de pouvoir manipuler un ensemble de données d'apprentissage complètement étiqueté, est souvent irréaliste. Dans beaucoup d'applications, le coût nécessaire pour étiqueter les données (phase qui est souvent effectuée par un expert) est excessif. Le résultat est que, en pratique, seul un petit nombre d'objets est étiqueté, ce qui rend la phase d'apprentissage impossible.

L'intuition qui est derrière la classification semi-supervisée, est que l'information portée par les données étiquetées, même si elle est insuffisante, peut guider un processus de classification non supervisée dans le but de trouver une partition des données "meilleure" que celle obtenue sans l'apport des instances étiquetées. La définition de "meilleure" dépend beaucoup de l'application. Si on se place dans un cadre supervisé, une partition "meilleure" est celle qui permet une précision majeure dans la phase de test. Si on est dans un cadre non supervisé, la définition de "meilleure" est encore une fois très subjective. On verra, que dans le but de valider leur méthode de classification semi-supervisée, la plupart des auteurs ont choisi une approche supervisée.

Il existe différentes approches en classification semi-supervisée, qui peuvent être regroupées en deux catégories : les méthodes basées sur une métrique, et les méthodes basées sur des contraintes. Dans les méthodes basées sur une métrique, on emploie un algorithme standard qui utilise une mesure de distance, mais la métrique est d'abord entraînée pour satisfaire les étiquettes ou les contraintes dans les données supervisées.

Plusieurs mesures de distance ont été proposées pour la classification semi-supervisée basée sur une métrique, par exemple la distance euclidienne qui est entraînée par un algorithme de type "chemin le plus court" [KKM02], ou les distances de Mahalanobis entraînées avec l'optimisation convexe [XNJR02, BHHSW03]. Les méthodes basées sur les contraintes consistent à exploiter un certain nombre de contraintes tout au long du processus de classification. Nous allons présenter dans la suite quelques unes de ces méthodes.

Méthodes de classification semi-supervisée basées sur les contraintes Le première approche de classification sous contraintes à été proposée par K. Wagstaff en 2001 [WCRS01]. L'idée est que, lorsqu'on cherche à produire une partition sur des données, on peut exploiter les connaissances *a priori* à travers deux contraintes au niveau des instances. Les auteurs considèrent deux types de contraintes pouvant impliquer des couples d'objets :

- La contrainte **must-link** spécifie que deux objets doivent être dans la même classe.
- La contrainte **cannot-link** spécifie que deux objets ne doivent pas être placés dans la même classe.

La contrainte must-link définit une relation binaire transitive sur les objets. Par conséquent, les auteurs considèrent la fermeture transitive de toutes les contraintes. L'ensemble complet de toutes les contraintes est donc présenté à leur algorithme de classification basé sur *k-means*, qu'ils appellent *COP-KMEANS*. L'algorithme considère un ensemble de contraintes must-link (noté $Con_{=}$) et un ensemble de contraintes cannot-link (noté $C_{\neq}$). Le changement majeur par rapport à *k-means* est que, dans la phase de mise à jour des classes, l'algorithme s'assure qu'aucune des contraintes n'est violée. L'objet est donc assigné à la classe la plus proche qui n'entraîne pas une violation des contraintes. Cette méthode montre clairement ses limites, car le fonctionnement dépend fortement de l'ordre dans lequel les objets sont considérés. Néanmoins, les auteurs arrivent à démontrer qu'il est possible d'améliorer sensiblement la précision du partitionnement, même avec un petit nombre de contraintes.

Davidson et Ravi [DR05b] ont étudié le problème de la faisabilité de la classification en présence de plusieurs combinaisons de contraintes dans une approche de type *k-means*. Il est en effet évident que pour certaines combinaisons de contraintes, la solution pour un K fixé n'existe pas toujours. Il suffit d'imaginer une situation où, avec $K = 2$, nous avons une contrainte cannot-link entre chaque couple des trois objets x, y et z. On aurait besoin d'au moins trois classes pour pouvoir assurer la satisfaction de la contrainte, donc un algorithme de classification devrait être en mesure de s'apercevoir de cette contradiction avant d'essayer le calcul de la partition. Les auteurs introduisent aussi deux nouvelles contraintes :

- La δ**-contrainte** (ou contrainte de séparation minimum), qui impose une distance minimale $\delta > 0$ entre chaque couple d'objets appartenant à deux classes

différentes.
- La **ϵ-contrainte** qui impose que, pour chaque objet x faisant partie d'une classe, il existe, dans la même classe, un autre objet y tel que la distance entre x et y est au maximum $\epsilon > 0$.

Les auteurs montrent que ce deux contrainte peuvent être représentées comme une conjonction (δ-contrainte) ou une disjonction (ϵ-contrainte) d'un certain nombre de contraintes must-link appropriées.

Les auteurs, en partant des résultats de [KKM02], arrivent à établir la complexité de calcul nécessaire pour décider de la faisabilité de différentes combinaisons de contraintes (cf. Table 1.2), et proposent un algorithme générique pour résoudre le problème de la faisabilité, ainsi qu'une version de l'algorithme *k-means* avec une adaptation de la fonction de distorsion qui prend en compte la satisfaction des contraintes.

Contrainte	K **fixé**	K **non fixé**
Must-Link	**P**	**P**
Cannot-Link	**NP**-Complet	**P**
δ-contrainte	**P**	**P**
ϵ-contrainte	**P**	**P**
Must-Link et δ	**P**	**P**
Must-Link et ϵ	**NP**-Complet	**P**
δ et ϵ	**P**	**P**
Must-link, Cannot-link, δ et ϵ	**P**	**NP**-Complet

TAB. 1.2 – Complexité pour la résolution du problème de la faisabilité

L'application de cet algorithme a montré qu'il est possible d'améliorer la précision de la classification ainsi que la vitesse de convergence pour atteindre la solution.

Les mêmes auteurs [DR05a] ont traité le problème de la faisabilité des contraintes lorsque le nombre de classes n'est pas fixé *a priori*, comme dans le cas d'un algorithme de classification hiérarchique. Le problème, ici, n'est plus de déterminer s'il existe une solution avec K classes pour une combinaison donnée de contraintes, mais s'il existe une partition quelconque qui satisfait les contraintes et, éventuellement, quel est le nombre minimum et maximum de classes compatibles avec une combinaison de contraintes donnée. Les auteurs parviennent aux résultats de la Table 1.2, et proposent un algorithme de classification hiérarchique ascendante. Le nombre maximum de classes K_{max} correspond au nombre de fermés transitifs des contraintes must-link plus le nombre d'objets qui ne sont pas concernés par ces contraintes. Cela correspond donc au niveau le plus bas dans la hiérarchie de classes. Le nombre K_{min} est au contraire déterminé par l'algorithme ascendant, et correspond aux nombre de nœuds

pour lequel aucune fusion n'est possible sans introduire une violation des contraintes. Un problème s'ajoute ici, à savoir l'impossibilité pratique, pour certaines combinaisons de contraintes, d'atteindre le nombre théorique K_{min} de classes, si les nœuds sont fusionnés de façon arbitraire (*dead-end*). Les auteurs montrent que ce risque existe en présence de contraintes cannot-link.

Une autre contrainte est introduite dans cet article. Il s'agit de la **γ-contrainte**, qui impose une distance maximum pour deux nœuds de l'arbre qui peuvent être fusionnés. L'objectif de cette contrainte est d'améliorer les performances de l'algorithme de classification hiérarchique, en réduisant le nombre de calculs des distances entre les objets.

Encore une fois, pour valider leur approche, les auteurs montrent le gain en terme de qualité de la partition (par rapport à la variable de classe), et de réduction du nombre de distances calculées (grâce aux contraintes γ et δ). Il semblerait donc que ces nouvelles contraintes ont pour objectif d'optimiser le calcul de la partition plutôt que de résoudre un problème réel de classification où le besoin objectif de l'utilisateur peut s'exprimer en termes de combinaison de contraintes.

Une autre approche de classification semi-supervisée sous contrainte a été proposée dans [BBM04a]. Il s'agit d'une version de l'algorithme *k-means* qui prend en compte les contraintes soit dans la phase d'initialisation, soit dans la phase d'affinement des centroïdes. L'initialisation des centroïdes est obtenue à travers la fermeture transitive des contraintes must-link qui determinent, avec les contraintes cannot-link étendues à tous les éléments des fermetures, une table de voisinages. Cette table donne l'information pour construire les K centroïdes initiaux. Pour assurer au maximum la satisfaction des contraintes pendant la phase d'affinement des centroïdes, la fonction objectif à minimiser prend en compte le nombre de contraintes violées. Chaque contrainte possède donc un poids, qui intervient dans le calcul de la fonction objectif de la manière suivante :

$$f = \frac{1}{2}\sum_{t_i \in \mathcal{T}} ||\boldsymbol{t_i} - \boldsymbol{\pi_{P(t_i)}}||^2 + \sum_{t_i,t_j \in \mathcal{C}_=} w_{ij} \mathbb{I}[P(t_i) \neq P(t_j)] + \sum_{t_i,t_j \in \mathcal{C}_\neq} \overline{w}_{ij} \mathbb{I}[P(t_i) = P(t_j)]$$

où $\boldsymbol{t_i}$ est le vecteur de l'objet t_i, $P(t_i)$ est une fonction qui donne l'indice de la classe à laquelle l'objet t_i appartient, $\boldsymbol{\pi_{P(t_i)}}$ est le vecteur du centroïde de la classe $P(t_i)$, w_{ij} et $\overline{w}_{ij}$ sont les poids associés respectivement à une contrainte must-link et à une contrainte cannot-link impliquant les objets t_i et t_j, et $\mathbb{I}[condition]$ est une fonction qui vaut 1 si la condition est vrai, 0 sinon.

Cette approche introduit donc une notion de contrainte molle, c'est-à-dire une contrainte dont la satisfaction n'est pas assurée à la fin du processus de classification. Cette approche a été ensuite améliorée avec l'introduction d'une étape d'entraînement de la métrique, donnant lieu à une approche de classification semi-supervisée hybride [BBM04b]. Dans cette approche aussi, les auteurs ont montré une amélioration de la

précision par rapport à la variable de classe.

1.2.3 Les méthodes de co-classification

Nous avons vu dans la section précédente un certain nombre d'approches de la classification unidimensionnelle, dont le but est de trouver une partition de l'ensemble des objets, en utilisant des notions de distance, similarité ou densité, calculées sur l'espace des attributs. Dans le cas du calcul de partitions basé sur les itemsets, l'ensemble des items et préalablement partitionné pour ensuite déduire la partition sur les transactions. Nous avons donc une espèce de bi-partitionnement où une partition est tout d'abord construite, et l'autre est le résultat direct de la première.

Le mot co-classification comporte, au contraire, une certaine idée de simultanéité dans la phase de construction des deux partitions. Avant de présenter les principales approches de co-classification, nous allons spécifier de façon formelle les différents termes utilisés dans la suite. Cette spécification n'est pas sans ambiguïté dans la littérature, mais elle nous servira pour souligner des caractéristiques qui différencient les méthodes entre elles.

Dans la suite, le terme *bi-partitionnement* est associé aux algorithmes qui construisent une partition de l'ensemble des objets à laquelle on associe une partition sur les attributs. Une bi-partition est donc telle que tout objet/atribut est associé à (au moins) une classe. La nature de cette association n'est pas spécifiée, et peut être le résultat de deux classifications effectuées de façon indépendante sur les deux ensembles. Un *bi-cluster* est un couple d'ensembles composé d'une classe sur une dimension et d'une deuxième classe sur l'autre dimension.

Le terme *bi-classification* (*bi-clustering* en anglais) est associé aux algorithmes qui extraient des bi-clusters, qui ne font pas forcément partie d'une bi-partition, mais où les deux ensembles sont associés par une quelconque corrélation. Dans ce cas on utilise le mot plus générique de *bi-ensemble*.

Le terme *co-classification* (*co-clustering* en anglais) se rapporte à tous les algorithmes qui construisent une bi-partition en optimisant un critère qui prend en compte les deux partitions, et pour lesquels il existe un lien entre une partition et une autre. Parfois, ce lien est bijectif, c'est-à-dire à une classe dans une partition correspond une et une seule classe dans l'autre partition.

Le bi-clustering a été développé surtout dans le cadre de l'analyse des données d'expression génique, où la matrice de données contient des valeurs réelles correspondant à la mesure de l'expression des gènes (les attributs) dans différentes situations biologiques (les objets). Le but des algorithmes de bi-clustering est donc de trouver des groupes de synexpressions, c'est-à-dire un certain nombre de gènes qui sont co-exprimés ensemble, et un certain nombre de situations biologiques qui expliqueraient

cette co-expression.

Les approches dites de co-classification, ont été développées dans le cadre des données catégorielles, dans le but de trouver simultanément une partition sur les objets et une partition sur les attributs. La plupart des méthodes utilisent une table de co-occurrence, qui mesure la fréquence d'apparition de chaque paire variable-modalité dans chacun des objets. Ce type de méthode est très utilisé en particulier pour l'analyse des données documents, où chaque attribut correspond à la fréquence d'apparition d'un mot dans le document (objet).

Les méthodes de bi-clustering dans les données numériques

L'objectif des méthodes de bi-clustering est d'extraire des couples d'ensembles de lignes et d'ensembles de colonnes d'un tableau de données, qui sont pertinents pour un objectif d'analyse donné. Dans la suite de ce chapitre, nous serons en présence de données matricielles (i.e., $\mathbf{r}$ est soit une matrice de nombres soit une matrice booléenne). La question de la pertinence est clairement dépendante de l'objectif d'analyse à un instant donné. Nous allons voir dans l'exemple suivant comment assister la découverte de groupes de synexpression.

Le tableau 1.3 représente le niveau d'expression de 5 gènes (colonnes) dans 10 conditions expérimentales (lignes). La figure 1.3 à gauche montre les niveaux d'expression des 5 gènes pour toutes les conditions expérimentales. Il apparaît que les gènes n'ont pas de profils d'expression identiques sur les 10 conditions. En revanche, si l'on considère l'ensemble des conditions $\{c_1 c_3 c_{10}\}$ et l'ensemble des gènes $\{g_1 g_5\}$ (voir la figure 1.3 à droite), une régularité apparaît.

	g_1	g_2	g_3	g_4	g_5
c_1	22	12	8	5	21
c_2	6	7	3	7	14
c_3	24	2	12	6	22
c_4	12	10	6	3	11
c_5	15	16	7	14	28
c_6	30	10	11	5	2
c_7	8	10	4	9	18
c_8	36	14	18	9	4
c_9	6	25	21	18	8
c_{10}	21	20	10	21	21

TAB. 1.3 – Données d'expression de gènes

Un premier modèle pour les groupes de synexpression sont des bi-ensembles que l'on pourrait appeler bi-ensembles presque "constants". En effet, ils ne contiennent

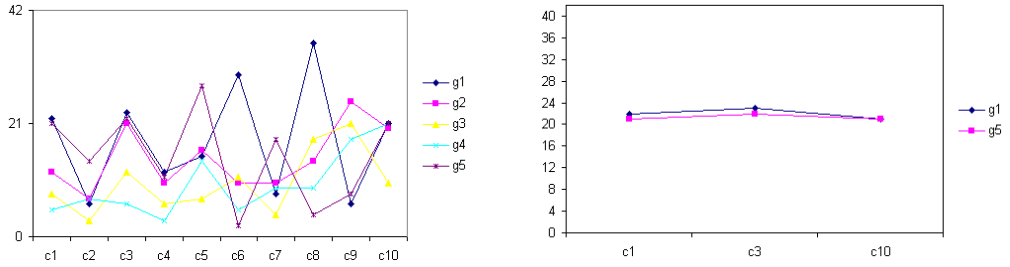


FIG. 1.3 – Exemple de motifs dans les données de la table 1.3

que des valeurs presque identiques. L'hypothèse biologique qui est faite pour ces motifs est que chaque gène répond de la même façon dans chaque condition expérimentale pour le même processus biologique. Or, cette hypothèse n'est pas complètement satisfaisante. Par exemple, on peut s'intéresser à des bi-ensembles ayant des valeurs identiques à une constante près, constante liée au gène et à la condition expérimentale. Ces constantes permettent d'exprimer le fait qu'un gène peut répondre différemment dans deux conditions, et qu'une condition peut induire des réponses différentes au sein d'un même mécanisme biologique. Les figures 1.4 (a) et (b) sont des exemples de ces modèles dit additifs. Ce différentiel d'expression peut aussi s'exprimer à l'aide de facteurs multiplicatifs (voir la figure 1.4 (c)). La figure 1.4 (d) montre un quatrième type de motifs. Dans cet exemple, l'ensemble des gènes $\{g_2, g_3, g_4\}$ ont des profils assez similaires dans les conditions $\{c_1, c_6, c_9\}$ mais surtout leurs profils sont très différents de ceux des gènes g_1 et g_5. Cette contrainte particulière permet de définir les motifs à la fois par rapport aux données dans le bi-ensemble, mais aussi par rapport aux données extérieures. Cette contrainte fait référence à une forme de maximalité des motifs : il satisfait le modèle et aucun autre élément ne peut être ajouté au motif sans violer le modèle.

Le second problème majeur posé par les méthodes de bi-clustering est lié à la combinatoire de la recherche. En effet, si l'on travaille sur un jeu de données contenant n conditions expérimentales et m gènes, il y a 2^{n+m} bi-ensembles possibles. Par exemple, en prenant 1000 gènes et 24 conditions ($n = 24$ et $m = 1000$), il y a 2^{1024} bi-ensembles possibles, c'est-à-dire plus de 10^{308} bi-ensembles possibles. Il n'est pas envisageable d'énumérer l'ensemble de ces candidats. Les algorithmes de bi-clustering doivent donc être capables d'extraire les motifs sans parcourir tout l'espace de recherche. Il faut néanmoins que l'extraction des motifs soit faisable ou du moins soit assez rapide pour préserver la dynamique des processus d'extraction. Certains algorithmes utilisent des méthodes d'optimisation locale, c'est-à-dire qui cherchent de bonnes solutions mais sans être sûr d'atteindre une solution optimale. Une telle méthode est utilisée lorsque les motifs sont définis à partir d'une contrainte dont

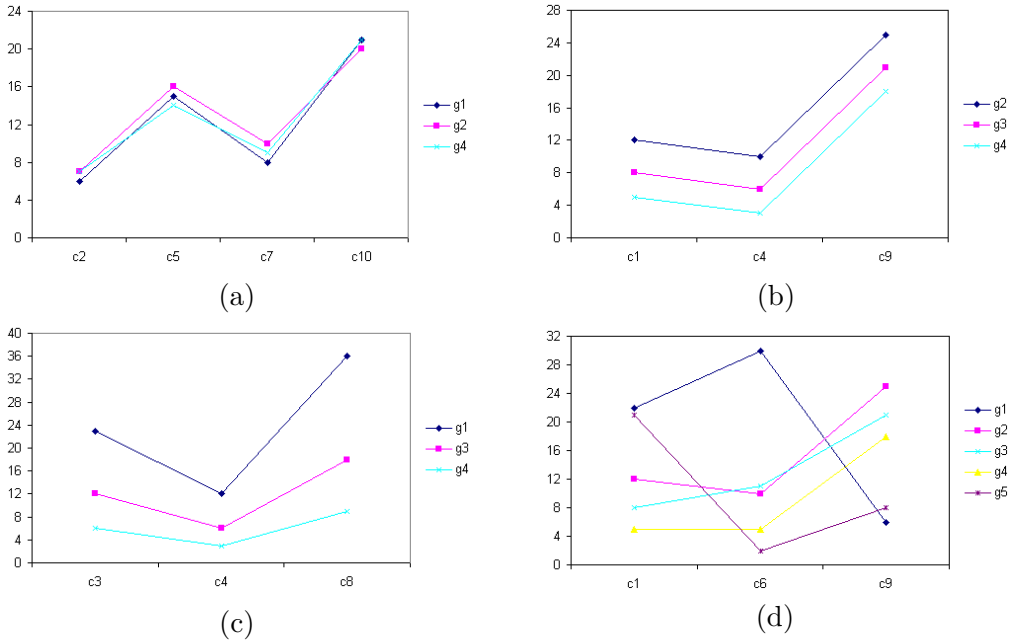


FIG. 1.4 – Exemples de modèles de bi-clustering

aucune solution exacte n'est connue, ou parce qu'elle est trop coûteuse à calculer. Dans ce cas, l'algorithme va chercher à s'approcher le plus près possible d'une (des) solution(s).

Bi-ensemble presque "constant" Une première façon de définir les groupes de synexpression consiste à considérer que ce sont des bi-ensembles presque "constants", c'est-à-dire tels que les valeurs dans chaque motif sont presque identiques. Cette définition informelle peut en fait se décliner sous la forme de différentes définitions visant à définir formellement le terme "presque". Nous allons ainsi voir que les trois méthodes présentées dans cette section utilisent des définitions différentes du terme "presque". Ces trois méthodes s'expriment toutes en termes de minimisation de l'inertie intra-classe de telle sorte que chaque motif et le modèle qui le définit soient les plus proches possibles, au sens d'une certaine distance. Une métrique qui définit la qualité globale d'une collection doit aussi être définie. Elle est d'ailleurs souvent définie à partir des distances entre le modèle et les bi-ensembles.

Pour la suite, nous désignerons par Moy_{XY} la moyenne des valeurs des lignes de X sur les colonnes de Y et par M_{xy} la valeur pour la ligne x et la colonne y.

Hartigan a été le premier à proposer, dès 1972 [Har72], une méthode appelée "Block Clustering" cherchant à extraire des bi-ensembles "constants". Il définit ainsi

le modèle de base : les bi-ensembles doivent contenir des valeurs proches de la moyenne des valeurs contenues dans le motif. Le bi-ensemble "parfait" ne contient que des valeurs identiques. Ensuite, il utilise pour mesurer la distance entre le modèle et les bi-ensembles, la variance des valeurs du bi-ensemble. En effet, plus la variance est faible et plus le motif est constant. La variance est la somme des différences au carré entre les valeurs et la moyenne du bi-ensemble (X,Y) :

$Variance(X,Y) = \sum_{x \in X, y \in Y} (M_{xy} - \text{Moy}_{X,Y})^2$

Ensuite, la mesure de qualité d'une collection va être simplement la somme des variances des différents motifs.

Cette mesure pose néanmoins un problème récurrent dans ce type de méthodes : la collection composée de tous les motifs de taille $1*1$ (avec une seule ligne et une seule colonne) a une mesure optimale. Pour pallier ce problème, la solution habituellement utilisée est d'ajouter une autre contrainte, visant à fixer a priori le nombre de motifs que doit contenir la collection extraite. Hartigan propose donc d'extraire les K bi-ensembles les plus "constants". La matrice initiale est découpée successivement en plusieurs sous-matrices et l'algorithme s'arrête lorsque la collection formée des K bi-ensembles a une distance globale inférieure à un seuil fixé par l'utilisateur. Une autre méthode vise à normaliser la mesure de qualité par la taille de la collection extraite.

Busygin et al. [BJK02] proposent d'utiliser des cartes auto-organisatrices de Kohonen (SOM) qui peuvent être considérées comme une généralisation de la méthode K-MEANS. La méthode proposée consiste à partitionner l'ensemble des conditions expérimentales et l'ensemble des gènes à l'aide des cartes auto-organisatrices de Kohonen (SOM) [Koh95] et à forcer le lien entre les deux partitions par l'intermédiaire d'une bijection associant à chaque nœud (le vecteur représentant chaque classe) d'un des deux espaces (conditions ou gènes) un nœud de l'autre espace appelé conjugué. Le procédé itératif consiste à construire l'une des deux partitions à l'aide de la méthode SOM, e.g., la partition des conditions expérimentales. Ensuite, les coordonnées des nœuds de la partition de l'autre espace, e.g., des gènes, sont calculées par le produit matriciel de la matrice d'expression gènes$\times$ conditions, dont chaque ligne a été normalisée pour être un vecteur unité, et de son conjugué qui vient d'être estimé. Une partition des gènes est alors construite à partir des coordonnées des nœuds à nouveau calculées avec la méthode SOM. On réestime ensuite les coordonnées des vecteurs, associés aux nœuds de la partition des conditions, par le produit matriciel entre la matrice conditions$\times$gènes dont chaque ligne a été normalisée. Le procédé est réitéré jusqu'à ce que les partitions se stabilisent. Cette méthode fournit alors une collection de bi-ensembles formant une partition des gènes et une partition des conditions expérimentales. Les classes des gènes discriminent les classes de conditions expérimentales et réciproquement. Cette méthode a comme avantage de converger relativement rapidement. Les collections de motifs extraits sont exclusives et partitionnent l'ensemble des gènes et des conditions. La figure 1.5 illustre comment ces

motifs sont calculés. Les ronds représentent des conditions, les rectangles des gènes et les croix les représentants des classes. Dans cet exemple, on cherche deux bi-ensembles. D'abord, les deux représentants des conditions sont choisis aléatoirement sur $\mathbb{R}^2$ et les conditions leur sont associées, noir pour le premier représentant et gris pour le second (voir deuxième figure). Ensuite les représentants pour les gènes sont calculés (voir troisième figure). Ils sont recalculés (voir quatrième figure). Ils sont de nouveau projetés sur l'espace des conditions. Cette configuration est un point fixe, la méthode s'arrête et deux bi-ensembles sont ainsi obtenus.

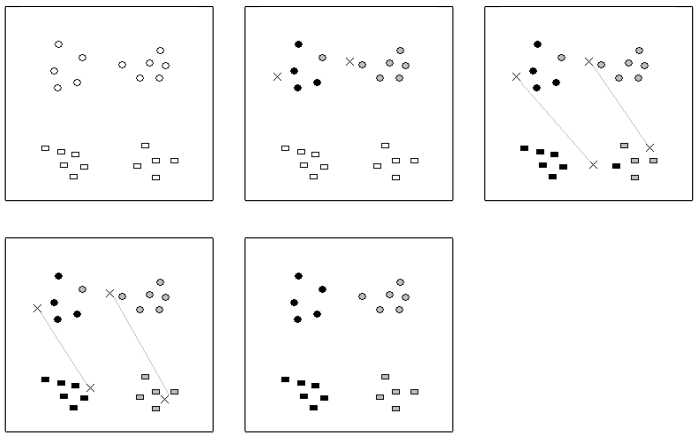


FIG. 1.5 – Bi-clustering avec SOM

Eisen et al. [ESBB98] proposent une méthode basée sur le clustering hiérarchique ascendant. Cette méthode est très largement utilisée sur les données d'expression de gènes. Le clustering hiérarchique ascendant regroupe successivement les classes d'éléments les plus proches en formant ainsi un dendogramme. Malheureusement, dans l'approche proposée par les auteurs, les conditions expérimentales et les gènes sont partitionnés de manière complètement indépendante. L'avantage principal de cette méthode réside dans son passage à l'échelle c'est-à-dire que des données contenant des dizaines de milliers de gènes et des dizaines de colonnes peuvent être utilisées. Il faut noter aussi que le succès de cette méthode est principalement dû à sa faculté à offrir une visualisation simple et intuitive des motifs extraits. La figure 1.6 montre un exemple de cette méthode. On peut voir (en haut) un dendogramme issu de la classification des colonnes (des gènes) et un autre (sur le côté) issu des lignes.

Bi-ensemble plus réaliste Cheng et Church [CC00] proposent un modèle un peu plus satisfaisant pour extraire des groupes de synexpression. Cette méthode est une amélioration du modèle proposé par Hartigan. En effet, les réponses transcriptionnelles des gènes ne sont pas identiques dans toutes les conditions biologiques. Deux gènes peuvent répondre dans une condition biologique mais à des niveaux différents.

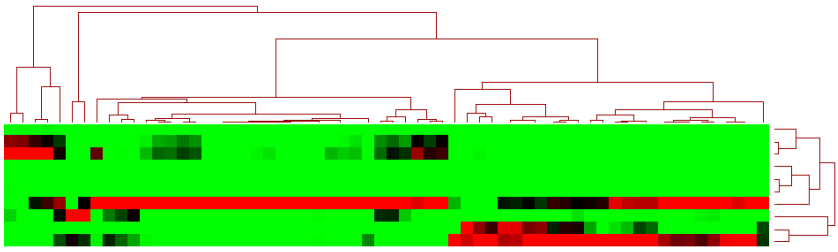


FIG. 1.6 – Bi-partitionnement pour le clustering hiérarchique sur les deux dimensions

Deux conditions peuvent induire des réponses très différentes pour un même gène. Ainsi, si l'on s'intéresse à des bi-ensembles représentant des groupes de synexpression, le modèle simple de Hartigan n'est pas complètement satisfaisant. Cheng et Church proposent d'ajouter à la valeur moyenne dans un bi-ensemble deux autres valeurs : une liée à l'influence du gène, et l'autre liée à la condition expérimentale. Ils proposent d'utiliser pour ces deux valeurs la moyenne Moy_{Xy} sur le gène y (resp. Moy_{xY} sur la condition expérimentale x) des valeurs d'expression pour toutes les conditions expérimentales (resp. pour tous les gènes) contenues dans le bi-ensemble (X,Y). Ils utilisent la distance H suivante pour mesurer la qualité d'un bi-ensemble :

$$H(X,Y) = \frac{\sum_{x \in X, y \in Y} (Moy_{xy} - D_{x,Y} - D_{X,y} + D_{X,Y})^2}{\mid X \mid\mid Y \mid}$$

avec $D_{x,Y} = \frac{Moy_{x,Y}}{|Y|}$, $D_{X,y} = \frac{Moy_{X,y}}{|X|}$ et $D_{X,Y} = \frac{Moy_{X,Y}}{|Y||X|}$.

La qualité globale d'une collection est la somme des distances pour chaque motif de la collection. Les auteurs emploient le terme de résidu pour la différence entre la valeur attendue et celle qui est dans la matrice. Le modèle de Hartigan peut être retrouvé en réalisant un simple pré-traitement sur la matrice. Comme les moyennes sont calculées sur l'ensemble des lignes et/ou des colonnes, il suffit d'enlever à chaque valeur du tableau la moyenne des valeurs de sa ligne et la moyenne des valeurs de sa colonne pour retrouver exactement le résultat de [Har72].

En revanche, Cheng et Church proposent plusieurs heuristiques pour extraire les motifs. L'une d'entres elles consiste à enlever itérativement des gènes et des conditions expérimentales un à un jusqu'à ce que la mesure de distance soit inférieure à δ ; c'est donc une approche divisive. Une limite de cette approche est que le nombre de bi-ensembles à rechercher est fixé par l'utilisateur tout comme le seuil δ utilisé pour la mesure de qualité. [YWWY02] généralise le travail réalisé par [CC00] en permettant la prise en compte des valeurs manquantes.

Lazzeroni et al. [LO00] proposent d'améliorer le modèle précédent. L'hypothèse

qui est faite est que, si un gène peut intervenir dans différents phénomènes biologiques, alors son niveau d'expression est lié à la combinaison de ces phénomènes. Pour capturer ces phénomènes, il faut non pas chercher des bi-ensembles presque "constants", mais ceux dont les valeur résultents de cette combinaison. La figure 1.7 montre un exemple de deux bi-ensembles dont la valeur (5) qui appartient aux deux bi-ensembles est la somme de la valeur du premier (2) et du deuxième (3).

2	2	2	
2	2	2	
2	**5**	**5**	3
	3	3	3

FIG. 1.7 – Exemple de deux bi-ensembles

Ainsi, d'une manière algébrique le niveau d'expression d'un gène i dans une condition expérimentale j est modélisé par :

$$Y_{ij} = \mu_0 + \sum_k (\mu_k + \alpha_{ik} + \beta_{jk})\, \rho_{ik}\psi_{jk}$$

où μ_0 représente le bruit de fond, μ_k la couleur du calque k, ρ_{ik} vaut 1 si i appartient au bi-ensemble k et 0 sinon, ψ_{jk} vaut 1 si la condition expérimentale j appartient au bi-ensemble k, et vaut 0 sinon, α_{ik} est un facteur correctif pour le gène i, et β_{jk} est un facteur correctif pour la condition expérimentale j. La méthode consiste alors à rechercher le modèle minimisant la distance euclidienne entre les valeurs d'expression observées et celles modélisées. L'estimation des paramètres se fait itérativement et ne produit qu'une valeur approchée. Les expérimentations fournies dans l'article semblent produire des résultats intéressants. Cette méthode est similaire aux méthodes de décomposition en valeurs singulières [KBCG03], mais ici les vecteurs ne sont pas contraints à être orthogonaux entre eux.

Califano et al. [CST00] présentent une autre façon de définir des bi-ensembles presque "constants". Les bi-clusters doivent contenir des valeurs comprises dans un intervalle de taille σ. Cette fois-ci, la mesure utilisée pour mesurer la qualité des motifs est la probabilité d'apparition "par chance" d'un motif. Cette probabilité est calculée à partir d'un ensemble de contrôle. Cette méthode est une méthode supervisée.

Une méthode complète Wang et al. [WWYY02] proposent une méthode exacte pour extraire des bi-ensembles presque "constants" appelés *pCluster* (pattern Cluster). Le modèle impose que les moyennes des valeurs des colonnes et/ou des lignes doivent appartenir à un intervalle de taille σ. Pour pouvoir extraire tous les motifs sans parcourir tout l'espace de recherche des bi-ensembles, ils proposent d'ajouter une contrainte supplémentaire aux bi-ensembles : ils imposent que tous les bi-ensembles

de taille 2*2 inclus dans un *pCluster* doivent aussi satisfaire le modèle. Ainsi, si l'un de ces bi-ensembles ne le satisfait pas alors il n'est pas nécessaire de regarder ses sur-ensembles. Plus précisément, tous les bi-ensembles de taille 2*2 ($\{o_1, o_2\}, \{a_1, a_2\}$) inclus dans un *pCluster* doivent satisfaire la contrainte suivante :

$$M_{o_1a_1} + M_{o_2a_2} - M_{o_1a_2} - M_{o_2a_1} < \sigma$$

Les auteurs cherchent de plus à extraire les bi-ensembles qui sont maximaux au niveau des colonnes pour un ensemble d'objets donné. Cet algorithme permet en plus d'imposer que les *pCluster* aient une taille minimale sur les lignes et les colonnes, contrainte exploitée efficacement pendant l'extraction. L'algorithme permet de ne pas calculer toutes les paires de lignes et de colonnes.

Pour extraire ces motifs, ils utilisent une méthode complète basée sur la recherche des motifs contenant 2 gènes et 2 conditions expérimentales et les éléments de l'autre dimension qui satisfont la contrainte précédente. Ces bi-ensembles sont ensuite utilisés pour générer des bi-ensembles plus grands.

Les méthodes de co-classification dans les données catégorielles

Dans cette catégorie, on trouve essentiellement des méthodes qui travaillent sur des matrices booléennes, où chaque attribut booléen code un couple variable-modalité, ou sur une table de contingence, où on registre la fréquence d'apparition de chaque modalité de la variable dans chaque objet. Les méthodes basées sur les tables de contingence (ou de co-occurrences), ont une application ultérieure dans le cadre de l'analyse des documents, où l'on stocke la fréquence d'apparition d'un mot dans chaque document.

Méthodes de co-classification de données binaires A partir de travaux de J. Hartigan [Har72], G. Govaert [CDG+88, Gov84] propose un algorithme de classification simultanée de type "nuées dynamiques" pour la construction de deux partitions liées à partir d'un tableau contenant des données binaires. La motivation de cette méthode est de travailler simultanément sur les deux ensembles qui décrivent les données afin de ne pas privilégier l'un par rapport à l'autre.

L'algorithme consiste à chercher deux partitions, une en K classes sur l'ensemble des objets, et l'autre en L classes sur l'ensemble des attributs, de telle sorte que les partitions définissent un ensemble de rectangles maximaux. Les cardinaux K et L des deux partitions sont fixés par l'utilisateur. L'objectif est d'obtenir, en réordonnant les lignes et les colonnes du tableau initial suivant les deux partitions, des blocs homogènes de 0 et de 1. La construction de ces deux partitions se fait par un processus itératif basé sur la minimisation d'une distance entre le tableau de données de départ et un tableau idéal binaire de dimension $K \times L$. La case de ce tableau indicée par

i, j contient un 1 si la classe i de la partition $\mathcal{P}^{\mathcal{T}}$ sur les objets est caractérisée majoritairement par la modalité 1 des variables de la classe j de la partition $\mathcal{P}^{\mathcal{A}}$. Le tableau idéal constitue un résumé simple à analyser du tableau de données initial. La distance utilisée pour mesurer l'écart entre le tableau initial et le tableau idéal est la distance "city-block" prise entre chaque élément du tableau originaire et l'élément du tableau idéal qui est associé aux groupes auxquels il appartient. Cette distance évalue le nombre de désaccords entre les deux tableaux :

$$D(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{A}}, Q) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{i \in P_k^{\mathcal{T}}} \sum_{j \in P_l^{\mathcal{A}}} |a_j(t_j) - q_{kl}|$$

où Q est le tableau idéal, $a_j(t_i)$ et $q_{kl} \in \{0, 1\}$.

L'algorithme consiste, en partant d'un couple de partitions $(\mathcal{P}_0^{\mathcal{T}}, \mathcal{P}_0^{\mathcal{A}})$ de départ ainsi que d'un tableau idéal Q_0, à fixer la partition $\mathcal{P}_0^{\mathcal{A}}$ pour améliorer la partition sur les objets ainsi que le tableau idéal, puis à fixer la partition $\mathcal{P}_1^{\mathcal{T}}$ précédemment obtenue pour calculer $\mathcal{P}_1^{\mathcal{A}}$. Le principal inconvénient de cet algorithme est qu'il n'existe pas de contrainte sur la forme générale du tableau. En effet, on n'impose pas, par exemple, que le tableau Q comporte un seul 1 par ligne et par colonne.

Les méthode basées sur les tables de contingence Deux variantes d'une même méthode de bi-partitionnement ont été développées de manière indépendante par Dhillon et al. [DMM03] et Robardet et al. [RF01a, RR02]. Cette méthode consiste à considérer les deux partitions cherchées comme des variables aléatoires X (associée à la partition $\mathcal{P}^{\mathcal{T}}$) et Y (associée à la partition $\mathcal{P}^{\mathcal{A}}$) à valeurs discrètes, et à concevoir la recherche d'une bi-partition comme un problème de maximisation de l'association entre ces deux variables. Il existe différentes mesures d'association qui évaluent le lien entre deux variables aléatoires à partir d'un tableau de co-occurrences $\mathbf{p}$. L'idée est d'associer à chaque colonne une classe de la partition des attributs, et à chaque ligne une classe d'objets. Un élément p_{ij} du tableau est alors égal au nombre de co-occurrences d'un attribut de la classe correspondant à la colonne j et d'un objet de la classe i. Ce tableau permet d'estimer empiriquement la distribution de la probabilité conjointe entre les deux variables représentant les partitions. Dhillon et al. [DMM03] utilisent la différence entre l'information mutuelle calculée en considérant la bi-partition discrète (i.e., la bi-partition qui associe une classe différente à chaque objet et attribut), et l'information mutuelle calculée pour un nombre $K < m$ et $L < n$ de classes. L'information mutuelle, qui mesure la quantité d'information que la variable X contient sur la variable Y (et vice versa), est égale à :

$$I(X;Y) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_{i.} p_{.j}}$$

où p_{ij} est la fréquence des relations entre un objet de la classe $P_i^{\mathcal{T}}$ et un attribut de la classe $P_j^{\mathcal{A}}$, $p_{i.} = \sum_j p_{ij}$ et $p_{.j} = \sum_i p_{ij}$.

Ainsi, pour deux bi-partitions $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{A}})$ and $(\hat{\mathcal{P}}^{\mathcal{T}}, \hat{\mathcal{P}}^{\mathcal{T}})$, la perte en information mutuelle est donné par :

$$I(X;Y) - I(\hat{X};\hat{Y})$$

où, $\hat{X}$ et $\hat{Y}$ sont les variables aléatoires associées aux deux partitions ($\hat{\mathcal{P}}^{\mathcal{T}}$ et $\hat{\mathcal{P}}^{\mathcal{T}}$).

Cette même mesure peut être représentée par la divergence de Kullback et Leibler. Cependant, il a été montré, d'un point de vue théorique et expérimental [Rob02], que les mesures de connexion étaient mieux adaptées que les mesures de divergence pour la recherche d'une bi-partition optimale. Robardet utilise dans [Rob02] la mesure de connexion τ de Goodman et Kruskal [GK54] pour évaluer la qualité de la bi-partition. Elle mesure la réduction proportionnelle de l'erreur dans la prédiction de $\mathcal{P}^{\mathcal{A}}$ donnée par la connaissance de $\mathcal{P}^{\mathcal{T}}$. Elle est calculé de la façon suivante :

$$\tau_{\mathcal{Y}} = \frac{\sum_i \sum_j \frac{p_{ij}^2}{p_{i.}} - \sum_j p_{.j}^2}{1 - \sum_j p_{.j}^2}$$

La même mesure calculée lorsque les partitions sont inversées est notée $\tau_{\mathcal{X}}$.

Chacune des deux méthodes produit une partition par un processus d'optimisation locale : [DMM03] proposent de fixer a priori le nombre de classes de chacune des deux partitions et optimisent localement la fonction en estimant itérativement une partition en fonction de l'autre jusqu'à convergence ; [Rob02] ne fixe pas a priori le nombre de classes des deux partitions et utilise alors un algorithme d'optimisation local stochastique qui procède également par ajustement itératif d'une partition en fonction de l'autre.

L'article [BDG+04] présente la recherche de bi-partitions (X,Y) optimales comme la recherche de bi-ensembles contenant le maximum d'information à l'intérieur par rapport à l'information contenu à l'extérieur du bi-ensemble. Il montre comment utiliser la divergence de Bregman pour essayer de résoudre ce problème. Cette divergence est de plus une généralisation d'un grand nombre de mesures habituellement utilisées dans ce type de problème.

1.3 Extraction de groupements locaux

Nous allons dans cette section présenter le cadre de la recherche des motifs locaux dans les données booléennes. Cet axe de recherche est aussi appelé "extraction d'itemsets fréquents" même si, depuis son émergence [AIS93], il ne se résume plus au calcul des itemsets fréquents, d'autres types de motifs ayant été proposés. Les données représentent une relation binaire $\mathbf{r}$ entre un ensemble d'objets noté $\mathcal{T}$ et un ensemble d'attributs noté $\mathcal{G}$. La relation $\mathbf{r}$ encode un lien ou une association entre un attribut et un objet. La sémantique de cette relation peut différer en fonction

des attributs et objets considérés, ainsi que du type d'association étudié par l'utilisateur. Toute propriété entre des attributs et des objets qui peut être représentée sous la forme d'une relation peut bénéficier des avancées technologiques relatives à l'extraction des motifs locaux dans les données booléennes. Nous utiliserons le terme "données booléennes" ou "données transactionnelles" pour désigner ce type de jeux de données. Une représentation sous forme matricielle est souvent adoptée ; les objets (resp. les attributs) peuvent alors être appelés des lignes (resp. des colonnes). Pour simplifier les notations, $\mathbf{r}$ sera utilisé pour désigner à la fois la relation et les données.

Mannila et Toivonen ont proposé une abstraction utile de nombreux travaux en fouille de données [MT97]. Considérons une base de données $\mathbf{r}$, un langage $\mathcal{L}$ pour l'expression de propriétés dans les données et un prédicat de sélection $\mathcal{C}$. Le prédicat $\mathcal{C}$ est utilisé pour dire si, oui ou non, une phrase $l \in \mathcal{L}$ doit être considérée comme intéressante sur $\mathbf{r}$. Une tâche d'extraction peut alors être formalisée comme le calcul de la théorie de $\mathbf{r}$ pour $\mathcal{L}$ et $\mathcal{C}$, i.e., l'ensemble $\mathcal{TH}(\mathbf{r}, \mathcal{L}, \mathcal{C}) = \{l \in \mathcal{L} \mid \mathcal{C}(l, \mathbf{r}) \text{ est vrai}\}$. On peut parler de la requête inductive $\mathcal{C}$ sur $\mathbf{r}$.

1.3.1 Bi-ensembles et 1-rectangles

Dans cette section, nous allons nous intéresser au calcul de théories de la forme $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{T}} \times 2^{\mathcal{G}}, \mathcal{C})$ avec $\mathbf{r} \subseteq \mathcal{O} \times \mathcal{A}$. Nous allons ainsi rechercher tous les bi-ensembles (voir définition 1.1) satisfaisant $\mathcal{C}$ avec $\mathbf{r}$ une relation binaire entre $\mathcal{O}$ et $\mathcal{A}$.

Définition 1.1 (bi-ensemble) *Un bi-ensemble (T, G) est un couple appartenant à $2^{\mathcal{T}} \times 2^{\mathcal{G}}$. Le terme de rectangle peut aussi être utilisé dans la mesure où il n'y pas d'ordre a priori sur les lignes et les colonnes. Les bi-ensembles sont des rectangles dans la matrice à un réarrangement près des lignes et des colonnes.*

Nous allons nous intéresser plus particulièrement à certains motifs appelés 1-rectangles (voir définition 1.2).

Définition 1.2 (1-rectangle) *Un bi-ensemble (T, G) est un 1-rectangle dans $\mathbf{r}$ si $\forall t \in T$, $\forall g \in G$, $(t, g) \in \mathbf{r}$.*

Exemple. Dans la Table 1.1b, le bi-ensemble $(\{t_1, t_3\}, \{g_1, g_3\})$ est un 1-rectangle dans $\mathbf{r}$. Par contre, le bi-ensemble $(\{t_3, t_5\}, \{g_1, g_2\})$ n'est pas un 1-rectangle dans $\mathbf{r}$ car $(t_3, g_2) \notin \mathbf{r}$.

Ces motifs sont particulièrement intéressants car ils permettent d'identifier des ensembles d'objets T et d'attributs G qui sont en relation, c'est-à-dire tels que tous les objets de T sont en relation avec tous les attributs de G et inversement.

Parmi ces motifs, les concepts formels sont très intéressants car les ensembles T et G sont maximaux, ce qui facilite l'interprétation des bi-ensembles.

1.3.2 L'analyse formelle des concepts

Avant de nous intéresser à la façon dont est appréhendée la notion de concept en classification non supervisée, étudions comment les psychologue cogniticiens la définissent [WB93]. Ils distinguent deux modes d'élaboration d'un concept. Dans le premier cas, un concept peut être caractérisé par des éléments ou des attributs invariants. Former un concept consiste alors à identifier ces invariants. Ce type de concept, dit "catégoriel", peut être défini en extension, par une collection d'objets représentatifs, et en compréhension, par la liste des attributs caractérisant les objets représentatifs du concepts. En référence à la conception rationaliste d'un concept, c'est-à-dire à la façon dont en science par exemple on élabore des concepts, des travaux de certains psychologues ont conduit à critiquer l'approche précédente. On considère alors qu'un concept est une entité élaborée par l'Homme en vue de pouvoir expliquer et agir. Ce type de concept est un ensemble d'hypothèses permettant d'unifier des idées et des faits. Ce sont des objets de pensée modifiables.

Introduite par R. Wille en 1982 [Wil82], et reprise dans [Wil89], l'analyse des concepts formels traite des concepts au sens philosophique du terme : un concept est un ensemble d'objets, l'*extension*, auxquels s'appliquent un ensemble d'attributs, la *compréhension* ou *intention*. Cette représentation est hiérarchique : un *super-concept* comprend un sur-ensemble des objets de son *sous-concept* (ou de ses sous-concepts) et un sous-ensemble de ses attributs. Par exemple, le concept "mammifère" comprend notamment les objets "vache", "dauphin", "souris" et les attributs "mobile" ou "allaite". Le concept "animal" ajoute l'objet "poule" et retranche l'attribut "allaite" : c'est donc un super-concept de "mammifère".

Un contexte formel, c'est-à-dire une matrice booléenne, est aussi un ensemble de concepts.

Le problème abordé dans cette section est celui de l'extraction de l'ensemble de tous les concepts d'un contexte formel $\mathbf{r}$, c'est-à-dire une relation binaire sur $\mathcal{T} \times \mathcal{G}$. D'une manière intuitive, un concept est un ensemble d'objets T et un ensemble de propriétés booléennes G tels que tous les objets de T sont en relation avec toutes les propriétés de G. De plus, T et G sont maximaux, c'est-à-dire que l'on ne peut ni rajouter un objet à T ni une propriété à G sans violer la propriété précédente.

On va s'intéresser à la famille des sous-ensembles de $\mathcal{G}$ que l'on notera $\mathcal{L}_{\mathcal{G}} = 2^{\mathcal{G}}$. De manière duale, on s'interressera à l'ensemble des sous-ensembles de $\mathcal{T}$ que l'on notera $\mathcal{L}_{\mathcal{T}} = 2^{\mathcal{T}}$. L'ensemble, noté $\mathcal{B}$, contenant tous les concepts de $\mathbf{r}$ est inclus dans $\mathcal{L}_{\mathcal{T}} \times \mathcal{L}_{\mathcal{G}}$. Pour permettre le calcul de $\mathcal{B}$ sans parcourir l'ensemble des éléments de $\mathcal{L}_{\mathcal{T}} \times \mathcal{L}_{\mathcal{G}}$, on utilise le fait que les ensembles $\mathcal{L}_{\mathcal{T}}$ et $\mathcal{L}_{\mathcal{G}}$, munis de l'inclusion ensembliste

(au sens large) $\subseteq$, sont des ensembles partiellement ordonnés.

Définitions

Définition 1.3 (fermeture d'un ensemble) *Soit E un ensemble partiellement ordonné par une relation binaire notée $\leq$. Une fonction ϕ de E dans E est une fermeture sur E si elle satisfait les propriétés suivantes :*

1. *ϕ est isotone : $\forall x, y \in E$, $x \leq y$ implique $\phi(x) \leq \phi(y)$ (ϕ est ici monotone croissante)*
2. *ϕ est extensive : $\forall x \in E$, $x \leq \phi(x)$*
3. *ϕ est idempotente : $\forall x \in E, \phi(\phi(x)) = \phi(x)$*

On dira qu'un ensemble est fermé s'il est égal à sa fermeture : x est fermé ssi $x = \phi(x)$

Définition 1.4 (connexion de Galois) *Si $T \subseteq \mathcal{T}$ et $G \subseteq \mathcal{G}$, notons $f(T) = \{g_j \in \mathcal{G} \mid \forall t_i \in T,\ r_{ij} = 1\}$ et $g(G) = \{t_i \in \mathcal{T} \mid \forall g_j \in G,\ r_{ij} = 1\}$. f représente l'ensemble de toutes les propriétés communes à un ensemble d'objets et g représente l'ensemble de tous les objets partageant un ensemble de propriétés. Le couple (f, g) définit la connexion de Galois entre $\mathcal{T}$ et $\mathcal{G}$.*

Propriété 1.1 *$h = f \circ g$ et $h' = g \circ f$ sont des opérateurs de fermeture. On les appelle opérateurs de la fermeture de Galois.*

Définition 1.5 (concept) *Si $G \in \mathcal{L}_\mathcal{G}$ et $T \in \mathcal{L}_\mathcal{T}$, on dit que (T, G) est un concept si $T = g(G)$ et $G = f(T)$. Par construction G et T sont des ensembles fermés. En effet, f et g conservent la propriété de fermeture : si G est fermé, $g(T)$ l'est également, de même pour T et $f(T)$.*

Si on observe le tableau 1.1b, selon la définition 1.5, on peut identifier les huit concepts de Table 1.4 :

Définition 1.6 (treillis) *Soit E un ensemble partiellement ordonné par une relation binaire notée $\leq$ et F un sous-ensemble de E. L'élément **minimum** (resp. maximum)*

b_i	G	T
b_1	$\{\emptyset\}$	$\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$
b_2	$\{g_1\}$	$\{t_1, t_3, t_5\}$
b_3	$\{g_5\}$	$\{t_2, t_5, t_6, t_7\}$
b_4	$\{g_2, g_5\}$	$\{t_2, t_5, t_6\}$
b_5	$\{g_3, g_4\}$	$\{t_1, t_3, t_4\}$
b_6	$\{g_1, g_2, g_5\}$	$\{t_5\}$
b_7	$\{g_1, g_3, g_4\}$	$\{t_1, t_3\}$
b_8	$\{g_1, g_2, g_3, g_4, g_5\}$	$\{\emptyset\}$

TAB. 1.4 – Liste des concepts dans **r**

de F est l'unique élément $a \in F$ tel que $\forall x \in F$, $a \leq x$ (resp.$\forall x \in F$, $a \geq x$). Une ***borne supérieure*** *de F est un élément $u \in E$ tel que $\forall x \in F$, $x \leq u$. La borne supérieure minimale de F est la plus petite des bornes supérieures de F. C'est le* ***supremum*** *de F.*

Un treillis est un ensemble E partiellement ordonné tel que chaque couple d'éléments de E, x, y possède un infimum, noté $x \wedge y$, et un supremum, noté $x \vee y$.

Propriété 1.2 *$\mathcal{B}$ est un treillis.*

Un exemple de treillis de Galois est celui qui est représenté dans la figure 1.8 est qui correspond à l'ensemble des fermés du tableau 1.4.

Définition 1.7 (contrainte) *Une contrainte $\mathcal{C}$ est une fonction de E dans $\{vrai, faux\}$.*

Définition 1.8 (contrainte anti-monotone) *Une contrainte $\mathcal{C}$ sur E est dite anti-monontone ssi $\forall x,\ y \in E,$ tel que $x \leq y$ alors $\mathcal{C}(y) \Rightarrow \mathcal{C}(x)$*

Définition 1.9 (contrainte monotone) *Une contrainte $\mathcal{C}$ sur E est dite monontone ssi $\forall x,\ y \in E,$ tel que $x \leq y$ alors $\mathcal{C}(x) \Rightarrow \mathcal{C}(y)$*

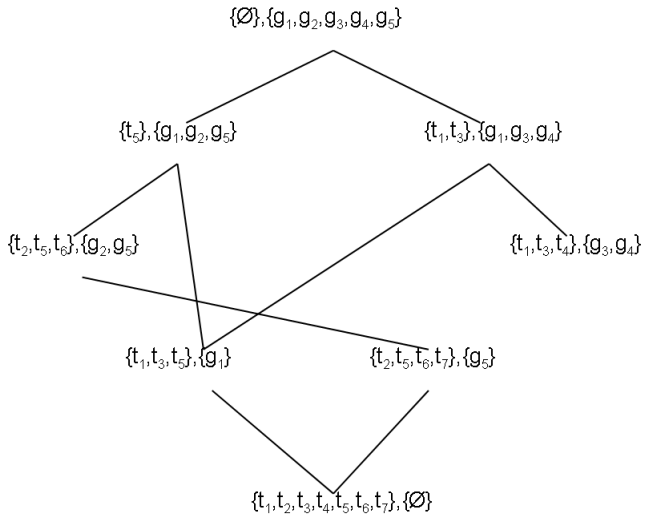


FIG. 1.8 – Treillis de Galois pour le tableau 1.4.

Les algorithmes d'extraction de concepts existants

Deux communautés de recherche très différentes ont travaillé sur le calcul d'ensembles vérifiant certaines propriétés dans un contexte $\mathbf{r}$. La première communauté (cf. par exemple [Gué90, Bor86, BS04]) a étudié, selon une approche mathématique, les treillis de concepts, et a proposé des algorithmes permettant de calculer l'ensemble $\mathcal{B}$ de tous les concepts de $\mathbf{r}$ ainsi que les arêtes du diagramme de Hasse représentant la relation de subsomption entre ceux-ci. La deuxième communauté a étudié, selon une approche algorithmique, l'extraction de collections d'ensembles vérifiant différentes contraintes anti-monotones, essentiellement la fréquence. Ces contraintes sont exploitées pour élaguer l'espace de recherche et permettre le calcul de telles collections (cf. par exemple [PHM00, ZH02, BBR03, Jeu02, STB+02, BRB04b]).

Extraction de concepts sous contraintes Les algorithmes d'extraction de concepts formels ou d'ensembles fermés comme CLOSET [PHM00], AC-MINER [BB00, BBR03] et CHARM [ZH02] peuvent exploiter durant l'extraction des contraintes anti-monotones sur les attributs et monotones sur les objets. Par exemple, avec ces algorithmes, on peut extraire tous les concepts formels (T, G) tels que $\sharp(G) \geq \sigma_1 \wedge \sharp(T) < \sigma_2$. Dans les contextes difficiles c'est-à-dire quand l'extraction de tous les concepts est impossible, ces contraintes permettent de rendre les extractions faisables. Malheureusement, les motifs qui sont alors extraits sont composés de beaucoup d'objets mais de peu d'attributs. Or, dans beaucoup de domaines d'application, les utilisateurs finals sou-

haitent obtenir des motifs composés d'un nombre minimal d'objets et d'attributs. Effectivement, il faut fixer une contrainte de taille minimale sur les objets (fréquence minimale), puis post-traiter la collection finale afin de ne conserver que les motifs ayant un nombre minimal d'attributs. De plus, il faut alors être capable d'exploiter la contrainte de taille minimale à la fois sur les objets et sur les attributs.

Ainsi, Besson a proposé un nouvel algorithme d'extraction de concepts formels appelé D-Miner [BRB04b, BRBR05] qui permet d'exploiter, durant l'extraction, des contraintes monotones sur les deux dimensions. Il est de plus très efficace dans les contextes relativement petits mais très denses.

D-Miner permet d'extraire les théories suivantes : $\mathcal{TH}(\mathbf{r}, 2^{\mathcal{T}} \times 2^{\mathcal{G}}, \mathcal{C})$ avec $\mathcal{C}$ une contrainte monotone sur $(2^{\mathcal{T}} \times 2^{\mathcal{G}}, \subseteq)$.

Parmi ces contraintes monotones, certaines sont particulièrement intéressantes. La première permet de fixer une taille minimale sur les deux dimensions des concepts. On peut s'intéresser seulement aux motifs contenant au moins un certain nombre de lignes et un certain nombre de colonnes. La deuxième contrainte intéressante permet d'imposer que l'aire des concepts extraits (T, G) soit supérieure à une valeur donnée, par exemple $|T| \times |G| > 20$. Cette dernière a aussi la particularité d'utiliser conjointement les deux dimensions $\mathcal{T}$ et $\mathcal{G}$, c'est-à-dire qu'elle ne peut pas être décomposée sous la forme de deux contraintes l'une sur $\mathcal{T}$ et l'autre sur $\mathcal{G}$. Enfin, on peut aussi imposer la présence de certains éléments dans les concepts extraits.

Ces contraintes sont particulièrement utiles lorsqu'on cherche à analyser une matrice d'expression binaire. D'autres travaux [RR05, ERR05] ont été effectués dans le cadre de l'extraction de motifs fréquents (et en particulier de motifs fréquents maximaux) dans des données puces à ADN.

Comme il a été montré dans [BRBR05], des contexte relativement petits, peuvent contenir un grand nombre de concepts formels. L'analyse des concepts formels, bien qu'elle soit souvent considérée comme une méthode de classification, faillit à un des objectifs principals qui est celui de trouver une représentation compacte de l'ensemble des données en termes de classes et de leur description.

Durand a donc proposé une méthode de sélection de concepts formels, qui permet de pallier cet inconvénient [DC02b, DC02a, Dur04]. Le but est de construire une classification approchée et de découvrir un ensemble de classes significatives avec un léger chevauchement d'éléments. La sélection est effectuée sur la base d'une mesure d'intérêt calculée pour chaque concept formel (T, G) comme :

$$interet(G) = \frac{homogeneite(G) + concentration(G)}{2}$$

où l'homogénéité est une fonction du seul concept formel (donc calculable dynamiquement) qui représente la similarité intra-classe, et la concentration permet d'évaluer

la dissimilarité inter-classe. L'homogénéité est calculée de la façon suivante :

$$homogeneite(G) = \frac{|T| \cdot |G|}{\sum_{t \in T} |f(t) - G| + (|T| \cdot |G|)}$$

La somme qui apparaît au dénominateur, est appelée divergence, et permet de mesurer l'erreur des transactions qui contiennent G par rapport à G. La concentration est calculée de la façon suivante :

$$concentration(G) = \frac{1}{|T|} \cdot \sum_{t \in T} \frac{1}{f\prime(t)}$$

où $f\prime(t)$ est égale au nombre de concepts formels qui contiennent t. L'algorithme de sélection, nommé *ECCLAT*, évalue les mesures de concentration et d'homogénéité dans une collection de concepts formels extraits avec une contrainte de taille minimale pour l'ensemble T.

Cette méthode permet, dans de jeux de données peu bruités, de construire un pavage de la matrice, avec relativement peu de chevauchement. En revanche, si le nombre de concepts formels est trop élevé, il faudra utiliser une taille minimale de l'ensemble T assez grande. Le résultat est que certaines transactions ne seront pas classées, surtout si on veut minimiser le chevauchement.

Une autre méthode pour réduire la taille de la collection des concepts formels, consiste à utiliser des étiquettes associées à chaque objet (telles que des variables de classe ou de type). Ces étiquettes fournissent des partitions des données qui sont ensuite utilisées pour reformuler la définition d'extension à travers un paramètre α, qui détermine le degré d'appartenance d'un objet à une classe [VSL04]. En jouant sur la valeur du paramètre α, on obtient donc un effet "zoom" sur le treillis de Galois (on parle de *Alpha Galois Lattice*). Les possibilités d'application de cette méthode dépendent donc de la disponibilité des variables de classe. De plus, le calcul des treillis de Galois Alpha est très laborieux, et dans beaucoup d'applications, il ne permet pas d'explorer le treillis jusqu'au niveau des instances (c'est-à-dire, pour des petites valeurs du paramètre α).

1.4 Problèmes ouverts et motivation du cadre L2G

Au cours de ce chapitre, nous avons rappelé les principes sur lesquels reposent les méthodes de classification existantes. En particulier nous avons étudié les approches unidimensionnelles, et nous avons souligné les avantages des approches de co-classification par rapport aux premières. Dans un cadre non supervisé, il apparaît fondamental de pouvoir disposer d'une description des classes, et les méthodes de co-classification permettent de partitionner l'ensemble des objets ainsi que celui des attributs. Cependant, les approches classiques en co-classification présentent des limites.

Les méthodes numériques sont extrêmement dépendantes du type d'homogénéité que l'on veut chercher dans les bi-ensembles. De plus, ces méthodes ne cherchent pas une bi-partition des données, mais se contentent de produire une collection de bi-ensembles, avec un certain niveau de recouvrement. Un certain nombre de méthodes conçues expressément pour les données catégorielles, ont l'avantage de trouver une vraie bi-partition dans les données, c'est-à-dire une partition des objets à laquelle est associée une partition sur les attributs. Le problème de ces méthodes est que les classes qu'elles trouvent sont complètement disjointes, alors que dans certaines applications, (par exemple en biologie) on devrait admettre des chevauchements entre celles-ci. D'ailleurs, même si la fonction objectif prend en compte les deux partitions, les algorithmes utilisent une approche itérative qui consiste à modifier une partition pendant que l'autre reste fixée. La construction des deux partitions n'est donc pas vraiment simultanée.

Ensuite, nous avons présenté le cadre plus générique de la classification conceptuelle, et en particulier l'analyse des concepts formels. Un concept résout le problème de l'interprétation, grâce aux propriétés de la connexion de Galois qui fait que à chaque classe d'objets (l'extension), correspond une et une seule classe de propriétés (l'intention) et vice versa. De plus, il est possible d'extraire une sous-collection des concepts qui satisfait un certain nombre de contraintes spécifiées par l'utilisateur. En revanche, une petite table peut contenir un nombre important de concepts, ce qui rend la tâche d'analyse plus difficile, au lieu de la simplifier. Les concepts formels ne peuvent pas être utilisés pour donner une représentation compacte de la table des données, vu leur nombre et leur sensibilité au bruit. Il serait pourtant intéressant de pouvoir spécifier des contraintes pour la classification de la même manière que dans le cadre de l'extraction des concepts.

La classification sous contrainte, comme nous l'avons vu, a été proposée comme solution au problème de la classification semi-supervisée, où il est possible de spécifier un certain nombre de contraintes pour prendre en compte les connaissances a priori sur les données. Le type de contraintes définies dans ce cadre, est malheureusement très limité, et il concerne uniquement des contraintes portant sur les objets, ou qui sont exprimables avec des contraintes de ce type-là. De plus, les méthodes proposées dans le cadre de la classification semi-supervisée sont toutes unidimensionnelles. Nous n'avons donc pas à disposition une interprétation des classes.

Nous avons résumé le principales caractéristiques de différentes approches dans la Table 1.5.

Il est évident qu'aucune des approches ne possède toutes les caractéristiques positives d'un algorithme de classification "idéal". La capacité à fournir une description est commune à toutes les méthodes conceptuelles, mais la facilité d'interprétation est une prérogative des méthodes d'extraction de bi-ensembles dans les données numériques, et des méthodes qui extraient les concepts formels dans des contextes

	Clust.	Bi-clust.	Co-clust.	Concepts	Semi-sup.
Description		X	X	X	
Bi-partition			X		
Chevauchement	X	X		X	
Modèle	X		X		X
Contraintes		X		X	X
Facilité interp.		X		X	

TAB. 1.5 – Résumé des caractéristiques des approches de classification

booléens. En revanche, ces deux types de méthodes ne fournissent pas une bi-partition de la matrice des données, et, le nombre de bi-ensembles pouvant être exponentiel avec la taille de la matrice, ils ne peuvent pas être utilisés comme représentation compacte et donc un modèle des données. Cependant ils ont l'avantage d'admettre le chevauchement des classes, et de permettre la spécification de contraintes.

En particulier, si on se place dans le cadre des données catégorielles, un bon compromis serait une méthode qui puisse combiner la facilité d'interprétation des concepts formels avec la capacité de modélisation des bi-partitions.

Nous considérons qu'il est important que les analystes puissent spécifier leurs attentes (intérêt subjectif) au moyen de contraintes, et qu'il faudrait des techniques de co-classification qui produisent des résultats cohérents vis-à-vis de ces spécifications. Le modèle simple de [Man97] aide à formaliser ce point de vue. Une classification peut être vue comme un processus d'évaluation d'une requête inductive qui calculerait $\{\phi \in \mathcal{L} \mid q(r, \phi) \text{ est vrai}\}$, où $\mathbf{r}$ est une matrice booléenne, $\mathcal{L}$ désignerait le langage des bi-partitions sur une telle matrice, et le prédicat q spécifierait les propriétés attendues sur la bi-partition ϕ. Une vision plutôt classique est que ce prédicat va exprimer une contrainte d'optimisation sur la fonction objectif utilisée. On peut également trouver d'autres contraintes comme la définition du nombre de bi-clusters, le fait que certains objets (resp. propriétés) doivent (resp. ne doivent pas) être ensemble, etc. Autrement dit, nous aimerions pouvoir réaliser la tâche comme une sélection de bi-partitions en supposant que toutes les bi-partitions aient été calculées a priori. Nous savons bien qu'un tel calcul est impossible. On comprend donc la nature heuristique des algorithmes de classification qui utilisent des méthodes d'optimisation locale de la fonction objectif (i.e. la satisfaction de la contrainte d'optimisation globale ne peut pas être garantie). Combiner ces heuristiques avec la satisfaction d'autres contraintes sur les bi-clusters est clairement un problème difficile.

Dans la suite de ce mémoire nous allons proposer une méthode pour la co-classification des données booléennes et l'interprétation des classes. Cette méthode utilise des motifs locaux (des bi-ensembles tels que les concepts formels), à la fois pour construire une bi-partition (éventuellement avec recouvrement) et pour l'interpréter. Nous avons appelé cette approche "cadre L2G" (Local-to-Global), et nous allons

montrer comment il est possible de l'étendre facilement pour exploiter de nouveaux types de contraintes portant sur la bi-partition.

Deuxième partie

Contribution méthodologique

Introduction

Dans le partie précédente, nous avons présenté un panorama des méthodes de recherche de regroupements intéressants. En particulier nous nous sommes intéressé à la co-classification et nous en avons mis en évidence certaines de ses limites. Premièrement il existe un problème d'interprétation. La description fournie par une bi-partition étant globale, certaines associations localement fortes pourraient échapper à l'analyste. Néanmoins, en présence d'un grand nombre d'attributs, cette description devient difficile à interpréter. Deuxièmement, il est impossible, pour un analyste, d'exprimer ses besoins en termes de contraintes. Nous avons vu que certaines approches de classification unidimensionnelle sous contraintes ont été étudiées, mais les contraintes qui ont été définies jusqu'ici ne sont qu'un petit nombre par rapport aux vrais besoins d'un utilisateur expert. De plus, il n'existe aucune méthode de co-classification basée sur les contraintes. Enfin, les solutions algorithmiques qui ont été proposées jusqu'à présent, sont basées sur un modification alternée de la partition des lignes et des colonnes, bien que la fonction objectif soit commune aux deux partitions. Il manque donc une réelle simultanéité de la phase de construction de la bi-partition.

Pour répondre à ces limites, nous proposons un cadre générique de travail que nous nommons "L2G" (Local-To-Global), et qui consiste à utiliser les motifs locaux (par exemple, des concepts formels) pour la construction de bi-partitions. Un autre aspect de notre travail (que l'on pourrait appeler "G2L", Global-To-Local), consiste à décrire une bi-partition préalablement calculée (ou induite par des variables de classe) avec une collection de motifs locaux. Les motifs locaux tels que les concepts formels capturent des associations localement fortes, c'est-à-dire, un phénomène local, circonscrit à une partie des données, et donc plus facilement explicable. Nous allons montrer comment ces phénomènes locaux peuvent non seulement contribuer à l'interprétation d'une bi-partition, mais aussi à sa construction. De plus, nous présenterons un nouveau problème, celui de la co-classification sous contraintes, en montrant comment notre cadre "L2G" peut favoriser le traitement des contraintes pour la co-classification. Nous allons d'abord présenter quelques notations et définitions dont nous aurons besoin pour la description de notre travail.

Contexte booléen Dans la suite de ce mémoire, nous allons nous intéresser au bi-partitionnement et à la caractérisation des bi-partitions d'un contexte booléen quelconque $\mathbf{r}$ composé d'un ensemble de m objets $\mathcal{T} = \{t_1, \ldots, t_m\}$, et un ensemble de n propriétés booléennes $\mathcal{G} = \{g_1, \ldots, g_n\}$. $r_{ij} = 1$ si et seulement si la propriété booléenne g_j est vraie pour l'objet t_i. On dit également que l'objet t_i contient l'item g_j.

$\mathcal{T}\|\mathcal{G}$	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

TAB. 1.6 – Un contexte booléen $\mathbf{r}$

Nous allons introduire ici un exemple de contexte booléen qui va nous servir dans les prochains chapitres pour présenter et expliquer nos méthodes.

Exemple. Le contexte booléen de la Table 1.6 représente la relation binaire entre l'ensemble d'objet $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ et l'ensemble de propriétés booléennes $\{g_1, g_2, g_3, g_4, g_5\}$. Dans ce contexte, la propriété g_3 est vraie pour l'objet t_4 (l'objet t_4 contient g_3), car $r_{43} = 1$.

Bi-partitions Une classe sur $\mathcal{T}$ est un ensemble d'objets $P^{\mathcal{T}} \subseteq \mathcal{T}$. Une classe sur $\mathcal{G}$ est un ensemble de propriétés $P^{\mathcal{G}} \subseteq \mathcal{G}$. Un couple de classes $(P^{\mathcal{T}}, P^{\mathcal{G}})$ forme un bi-cluster dans $\mathbf{r}$. Une partition sur $\mathcal{T}$ est un ensemble de K classes $\mathcal{P}^{\mathcal{T}} = \{P_1^{\mathcal{T}}, \ldots, P_K^{\mathcal{T}}\}$ telles que $\bigcup_{k=1}^{K} P_k^{\mathcal{T}} = \mathcal{T}$ et $\bigcap_{k=1}^{K} P_k^{\mathcal{T}} = \emptyset$. Une partition sur $\mathcal{G}$ est un ensemble de L classes $\mathcal{P}^{\mathcal{G}} = \{P_1^{\mathcal{G}}, \ldots, P_L^{\mathcal{G}}\}$ telles que $\bigcup_{l=1}^{L} P_l^{\mathcal{G}} = \mathcal{G}$ et $\bigcap_{l=1}^{L} P_l^{\mathcal{G}} = \emptyset$. Si les intersections ne sont pas vides, on parle de "soft clustering" et de classes chevauchantes. Une bi-partition est un couple de partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$. Dans la suite on va s'intéresser uniquement aux bi-partitions avec un même nombre de classes (noté K) sur les deux dimensions. Un algorithme de co-classification fournit une bi-partition munie d'une fonction qui associe les couples de classes. Lorsque les deux partitions ont le même nombre de classes, cette fonction est bijective, c'est-à-dire il existe une fonction bijective $\mathcal{M}$ qui, à un élément de $\mathcal{P}^{\mathcal{T}}$, associe un et un seul élément de $\mathcal{P}^{\mathcal{G}}$, et sa fonction inverse $\mathcal{M}^{-1}$ qui, à un élément de $\mathcal{P}^{\mathcal{G}}$, associe un et un seul élément de $\mathcal{P}^{\mathcal{T}}$. Nous allons donc considérer uniquement les bi-clusters $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ tels que $P_k^{\mathcal{G}} = \mathcal{M}(P_k^{\mathcal{T}})$ et $P_k^{\mathcal{T}} = \mathcal{M}^{-1}(P_k^{\mathcal{G}})$.

Exemple. Dans le contexte de la Table 1.6, $\mathcal{P}^{\mathcal{T}} = \{\{t_1, t_3, t_4\}, \{t_2, t_5, t_6, t_6\}\}$ est une

partition sur $\mathcal{T}$ et $\mathcal{P}^{\mathcal{G}} = \{\{g_1, g_3, g_4\}, \{g_2, g_5\}\}$ est une partition sur $\mathcal{G}$. La bi-partition résultante $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$ est donc composée des deux bi-clusters $(\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\})$ et $(\{t_2, t_5, t_6, t_6\}, \{g_2, g_5\})$.

Bi-ensembles Un bi-ensemble est un couple d'ensembles (T, G), où $T \subseteq \mathcal{T}$ et $G \subseteq \mathcal{G}$. Un bi-ensemble définit un rectangle dans la matrice à des permutations de colonnes et de lignes près. Dans un bi-ensemble quelconque, la nature (le nombre de 1 et 0) de ce rectangle n'est pas définie. Si, par contre, on considère un concept formel (cf. Définition 1.5), il s'agit d'un bi-ensemble maximal (T, G) où toutes les propriétés de G sont vraies pour tous les objets de T.

Exemple. Dans le contexte booléen de la Table 1.6, $(\{t_1, t_2, t_3\}, \{g_1, g_2, g_3\})$ est un bi-ensemble quelconque, $(\{t_1, t_3, t_4\}, \{g_3, g_4\})$, $(\{t_1, t_3\}, \{g_1, g_3, g_4\})$, $(\{t_5\}, \{g_1, g_2, g_5\})$ sont des concepts formels dans $\mathbf{r}$, mais $(\{t_5, t_6\}, \{g_2, g_5\})$ n'est pas un concept formel car il n'est pas maximal.

On notera $\mathcal{B}$ une collection de N bi-ensembles dans $\mathbf{r}$.

Motifs locaux et motifs globaux Dans les définitions que nous venons d'introduire, il peut apparaître que des objets tels qu'un bi-cluster et un bi-ensemble paraissent identiques. Or, nous avons eu besoin de deux dénominations car, même si d'un point de vue formel ce sont deux couple d'ensembles de $\mathcal{T}$ et $\mathcal{G}$, la manière dont ils sont calculés leur confère des qualités bien différentes. En effet, un bi-cluster est issu d'une bi-partition résultant d'un calcul où l'ensemble de la matrice de données a été prise en compte afin d'extraire les associations les plus fortes permettant de partitionner "au mieux" la matrice dans son ensemble. On est ici face à ce que Hand a défini comme un *motif/modèle global* [Han02]. En revanche, un bi-ensemble est défini par des contraintes qui sont évaluées localement dans les données. Ces contraintes ne visent pas à structurer le jeu de données dans son ensemble, mais permettent d'identifier des zones anormalement denses localement dans les données. Ce sont les *motifs locaux*.

Pour mieux expliquer la différence entre un motif local et un motif global, on peut considérer la définition que D.J. Hand donne de motif [Han02] : "un motif est un vecteur de données qui sert à décrire une densité anormalement haute de points de données". Le mot "anormale" implique qu'il y a quelque chose avec quoi la densité locale peut être comparée. C'est-à-dire qu'il y a une valeur de fond, de base ou attendue pour la densité. Hand précise que cette définition doit s'entendre dans le sens de "densité locale".

On pourrait dire qu'un motif global est un motif qui prend son sens dans un modèle qui représente la totalité des données. En classification, une partition est un modèle qui est défini à partir de la similarité intra-classe et de la dissimila-

rité inter-classes. Un bi-cluster n'a pas de sens (sémantique) si on ne le considère pas dans le contexte d'une bi-partition. En revanche, un motif local est un motif qui représente une association localement et anormalement forte dans le contexte, et dont le sens ne dépend d'aucun autre motif local. Un concept formel est défini uniquement par les fonctions de Galois qui définissent l'extension et l'intention du concept indépendamment des autres concepts formels. On pourrait alors être amené à penser qu'un treillis de Galois est un modèle des données. Mais si on considère que le nombre des concepts formels dans certains contextes est exponentiel par rapport à la taille de la matrice, on perd une caractéristique fondamentale d'un modèle, qui est celle d'être une représentation compacte et simple des données.

Cela cadre à la perfection avec le point de vue de Hand, selon lequel les données ne sont que la somme d'un modèle de fond (par exemple une bi-partition), plus des motifs locaux (par exemple des concepts formels) et une composante aléatoire (par exemple du bruit). Dans cette partie du mémoire nous allons justement explorer comment ces trois parties peuvent intervenir dans l'analyse des données, et en particulier dans la découverte de connaissances. Dans le Chapitre 2, nous allons présenter une méthode pour l'interprétation des bi-partitions à l'aide de motifs locaux. On abordera aussi la thématique du bruit dans les données, en présentant un nouveau type de bi-ensemble tolérant aux exceptions. Dans le Chapitre 3, nous présenterons notre approche de co-classification (nommée "cadre L2G") basée sur les motifs locaux. Le Chapitre 4 est dédié à la pertinence des bi-partitions et à l'utilisation de contraintes dans un contexte de co-classification, et en particulier dans le cadre L2G.

Chapitre 2

Caractérisation des classes

2.1 Introduction

Dans le premier chapitre, nous avons analysé un certain nombre d'approches de classification et de co-classification. Nous avons mis en évidence le principal avantage de la co-classification, c'est-à-dire, la possibilité de fournir à la fois une partition sur les objets et une description de chaque classe par un ensemble d'attributs. Ces ensembles d'attributs forment également une partition.

Toutefois, notre expérience nous suggère que cette première étape vers la caractérisation n'est pas suffisante, surtout dans des données avec un grand nombre de dimensions. Dans ce type de données, les motifs globaux tels que les bi-partitions ne reflètent pas les associations localement fortes et inattendues entre certains ensembles d'objets, et certains ensembles de propriétés. De plus, dans certaines applications, les jeux de données possèdent déjà des variables de classe (déduites par les experts, ou à partir de certains attributs catégoriels). On pourrait donc s'intéresser à trouver des associations fortes qui caractériseraient ces classes.

Notre proposition est de combiner la co-classification avec une phase de caractérisation, basée sur une collection de motifs locaux. Nous considérons, dans ce chapitre, qu'une bi-partition d'un jeu de données booléen est déjà disponible (elle a été calculée avec, par exemple, l'algorithme de Dhillon [DMM03]). Notre contribution à la caractérisation des bi-partitions, qui a été présentée dans [PB05b], est la suivante. Premièrement, nous allons introduire une technique originale pour la caractérisation des bi-clusters, basée sur l'extraction de bi-ensembles sous contraintes, c'est-à-dire, des bi-ensembles dont les ensembles qui les composent satisfont certaines contraintes. Nous allons montrer comment mesurer le fait qu'un bi-ensemble donné soit un motif précis pour la caractérisation d'un bi-cluster donné. Grâce à cette mesure de précision, il est possible de considérer des règles de caractérisation qui peuvent

aider à la découverte de connaissances à partir des résultats de la co-classification. La méthode est illustrée sur deux types de bi-ensembles, les concepts formels, et une nouvelle classe de motifs, appelée δ-bi-ensembles. Ce dernier type de motif est nouveau, bien que basé sur un travail antérieur portant sur les représentations condensées approximatives pour les motifs fréquents [BBR03]. Intuitivement, un δ-bi-ensemble est un "rectangle à valeurs vraies avec un nombre borné d'exceptions par colonne". Nous allons enfin valider notre approche par une série d'expériences sur des jeux de données benchmark et réels.

2.2 Caractérisation d'une bi-partition à l'aide de bi-ensembles

Notre objectif est de fournir une aide à l'interprétation des bi-clusters, via une collection de bi-ensembles qui précisent localement des associations intéressantes entre des groupes d'objets et des groupes de propriétés. Nous supposons ici qu'une collection de N bi-ensembles $\mathcal{B} = b_1, \ldots, b_N$ a été extraite dans les données.

La première phase du processus de caractérisation consiste à associer chacun des N bi-ensembles à l'un des K bi-clusters. Chaque bi-ensemble caractérise le bi-cluster auquel il est associé avec un certain degré de précision. Nous définissons alors une mesure de similarité entre un bi-ensemble $b = (T, G)$ et un bi-cluster $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ de la manière suivante :

$$sim\left(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}})\right) = \frac{|T \cap P_k^{\mathcal{T}}| \cdot |G \cap P_k^{\mathcal{G}}|}{|T \cup P_k^{\mathcal{T}}| \cdot |G \cup P_k^{\mathcal{G}}|}$$

Intuitivement, (T, G) et $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ définissent deux rectangles dans la matrice (à des permutations de lignes et de colonnes près) et nous mesurons la surface de leur intersection normalisée par la surface de leur union.

Chaque bi-ensemble b est un motif de caractérisation candidat et peut être assigné au bi-cluster k $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ tel que $sim(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}}))$ est maximum pour k. Nous avons donc K groupes de bi-ensembles potentiellement caractérisants, que l'on note $\{B_1, \ldots, B_K\}$. Chaque groupe est donc construit de la façon suivante :

$$b_i \in B_k \text{ ssi } \{k = \underset{k \in \{1 \ldots K\}}{\operatorname{argmax}} sim(b_i, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}}))\}$$

Exemple. Dans le contexte booléen **r** de la Table 1.6, une partition possible est

$$\{(P_1^{\mathcal{T}}, P_1^{\mathcal{G}}), (P_2^{\mathcal{T}}, P_2^{\mathcal{G}})\} \quad = \quad \{(\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}), (\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\})\}$$

Si on considère le bi-ensemble $b_1 = (\{t_1, t_3, t_5\}, \{g1\})$, ses mesures de similarité par rapport à $(P_1^{\mathcal{T}}, P_1^{\mathcal{G}})$ et $(P_2^{\mathcal{T}}, P_2^{\mathcal{G}})$ sont :

$$sim(b_1, ((P_1^{\mathcal{T}}, P_1^{\mathcal{G}}))) = \frac{2 \cdot 1}{3 \cdot 1 + 3 \cdot 3 - 2 \cdot 1} = 0.2$$

$$sim(b_1, ((P_2^{\mathcal{T}}, P_2^{\mathcal{G}}))) = \frac{1 \cdot 0}{3 \cdot 1 + 4 \cdot 2 - 1 \cdot 0} = 0$$

Le bi-ensemble b_1 est donc associé au premier bi-cluster. Si on considère maintenant le bi-ensemble $b_2 = (\{t_5\}, \{g_1, g_2, g_5\})$, on obtient :

$$sim(b_2, (P_1^{\mathcal{T}}, P_1^{\mathcal{G}})) = \frac{0 \cdot 1}{1 \cdot 3 + 3 \cdot 3 - 0 \cdot 1} = 0$$

$$sim(b_2, (P_2^{\mathcal{T}}, P_2^{\mathcal{G}})) = \frac{1 \cdot 2}{1 \cdot 3 + 4 \cdot 2 - 1 \cdot 2} = 0.22$$

Ce bi-ensemble b_2 est donc associé au deuxième bi-cluster.

Enfin, nous pouvons utiliser une mesure de précision (ou d'erreur) pour avoir un moyen de sélectionner les bi-ensembles les plus pertinents. Nous proposons d'utiliser comme mesure la proportion d'exceptions pour les deux composantes du bi-ensemble.

Soit (T, G) un bi-ensemble et $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ un bi-cluster, la proportion d'exceptions peut être calculée de la manière suivante :

$$\begin{aligned} \epsilon_t(T, P_k^{\mathcal{T}}) &= \frac{|\{t_i \in T|\ t_i \notin P_k^{\mathcal{T}}\}|}{|T|} \\ \epsilon_g(G, P_k^{\mathcal{G}}) &= \frac{|\{g_i \in G|\ g_i \notin C_k^{\mathcal{G}}\}|}{|G|} \end{aligned}$$

Exemple. Dans notre exemple de la Table 1.6, le bi-ensemble $b_1 = (\{t_1, t_3, t_5\}, \{g_1\})$ contient l'objet t_5 qui n'appartient pas à $P_1^{\mathcal{T}}$; nous avons donc une proportion d'exceptions égale à :

$$\epsilon_t(\{t_1, t_3, t_5\}, \{t_1, t_3, t_4\}) = \frac{1}{3} = 0.33$$

Le bi-ensemble $b_2 = (\{t_5\}, \{g_1, g_2, g_5\})$ contient la propriété g_1 qui n'appartient pas à $P_2^{\mathcal{G}}$; nous avons donc :

$$\epsilon_g(\{g_1, g_2, g_5\}, \{g_2, g_5\}) = \frac{1}{3} = 0.33$$

Il est possible de considérer des seuils pour sélectionner seulement les bi-ensembles qui caractérisent les bi-clusters avec un nombre d'exceptions petit, c'est-à-dire, $\epsilon_t < \varepsilon_t$

et $\epsilon_g < \varepsilon_g$, où $\varepsilon_t, \varepsilon_g \in [0, 1]$. Il y a différentes interprétations possibles pour ces seuils. Si on s'intéresse à la caractérisation d'une classe d'objets, on peut chercher tous les ensembles de propriétés (itemsets) tels que les valeurs de ϵ_t sont inférieures à un seuil ε_t. Si on s'intéresse, au contraire, à la caractérisation d'une classe de propriétés, on peut chercher tous les ensembles d'objets tels que les valeurs de ϵ_g sont inférieures à un seuil ε_g. Alternativement, on peut considérer tout le bi-cluster et le caractériser avec tous les bi-ensembles tels que les deux proportions d'exceptions ϵ_t et ϵ_g sont inférieures aux seuils ε_t et ε_g.

2.3 Choix du type de bi-ensemble pour la caractérisation

Nous discutons dans cette section des types de bi-ensembles qui peuvent être post-traités pour la caractérisation des bi-clusters. Il est clair que les bi-clusters sont par construction des bi-ensembles intéressants pour la caractérisation, mais ils fournissent seulement une interprétation globale. Nous sommes intéressés par des associations fortes entre des ensembles d'objets et des ensembles de propriétés qui peuvent expliquer localement ce comportement. Il paraît alors clair que les concepts formels sont parmi les meilleurs candidats.

Un problème majeur avec les concepts formels, est que la connection de Galois donnée par les fonctions (f, g) (cf. Section 1.3.2) est, d'une certaine façon, une relation trop forte (elle capture tout ensemble fermé d'objets et les propriétés qui lui sont associées). Par conséquent, le nombre de concepts formels, même dans des matrices petites, peut être énorme. En effet, il est très commun d'obtenir plusieurs millions de concepts formels même dans de petites matrices. Une solution est de chercher plutôt des rectangles "denses" dans la matrice, c'est-à-dire des bi-ensembles avec une majorité de valeurs vraies mais aussi avec un nombre borné (petit) de valeurs fausses ou exceptions. Des approches pour l'extraction de bi-ensembles denses ont été proposées par Besson [BRB04b, BRB05]. Ici, nous allons présenter un nouveau type de bi-ensembles qui peut être efficacement calculé, et qui est un extension des concepts formels vers la tolérance au bruit.

2.3.1 Extraction de δ-bi-ensembles

Nous voulons calculer efficacement des collections plus petites de bi-ensembles qui capturent encore des associations fortes. Nous allons rappeler quelques définitions sur la tâche d'extraction de règles d'association [AIS93] qui va nous servir à la fois pour la définition du type de motif nommé δ-bi-ensemble et pour la caractérisation des bi-clusters.

Définition 2.1 (règle d'association) *Soit* $\mathbf{r}$ *un contexte formel. Une règle d'asso-*

ciation dans $\mathbf{r}$ *est une expression de la forme* $X \Rightarrow Y$*, où* $X, Y \subseteq \mathcal{G}$*,* $Y \neq \emptyset$ *et* $X \cap Y = \emptyset$*. Sa fréquence absolue est* $|g(X \cup Y, \mathbf{r})|$ *et sa confiance est* $|g(X \cup Y, \mathbf{r})|/|g(X, \mathbf{r})|$*, où* $g(G) = \{t_i \in \mathcal{T} \mid \forall g_j \in G,\ r_{ij} = 1\}$*.*

Dans une règle d'association $X \Rightarrow Y$ à forte confiance, les propriétés dans Y sont presque toujours vraies pour un objet, lorsque les propriétés dans X sont vraies. Intuitivement, $X \cup Y$ associé à $g(X, \mathbf{r})$ est donc un bi-ensemble dense : il contient un nombre petit de valeurs fausses. On considère maintenant une technique pour calculer des règles d'association à forte confiance, les règles δ-fortes [BBR00, BBR03].

Définition 2.2 (règle δ-forte) *Soit* δ *un entier, une règle* δ*-forte dans* $\mathbf{r}$ *est une règle d'association* $X \Rightarrow Y$ $(X, Y \subset \mathcal{G})$ *telle que* $|g(X, \mathbf{r})| - |g(X \cup Y, \mathbf{r})| \leq \delta$*, c'est-à-dire, la règle est violée dans au plus* δ *objets.*

On peut calculer efficacement des collections intéressantes de règles δ-fortes avec une partie gauche minimale à partir des ensembles δ-libres [BBR00, BBR03, CB02] et de leur δ-fermeture.

Définition 2.3 (ensemble δ-libre, δ-fermeture) *Soit* δ *un entier et* $X \subset \mathcal{G}$*,* X *est un ensemble* δ*-libre dans* $\mathbf{r}$ *si et seulement si il n'existe aucune règle* δ*-forte entre deux de ses sous-ensembles stricts. La* δ*-fermeture de* X *dans* $\mathbf{r}$*,* $h_\delta(X, \mathbf{r})$*, est le sur-ensemble maximal* Y *de* X *tel que* $\forall g \in Y \setminus X$*,* $|g(X \cup \{g\})| \geq |g(X, \mathbf{r})| - \delta$*. En d'autres termes, la fréquence de la* δ*-fermeture de* X *dans* $\mathbf{r}$ *est presque égale à la fréquence de* X *lorsque* $\delta << |\mathcal{T}|$ *et* X *est fréquent. De plus,* $\forall g \in h_\delta(X) \setminus X$*,* $X \Rightarrow g$ *est une règle* δ*-forte.*

Exemple. Dans les données de la Table 1.6, les ensembles δ-libres d'items sont $\{g_1\}$, $\{g_2\}$, $\{g_3\}$, $\{g_4\}$, $\{g_5\}$, $\{g_1, g_2\}$, et $\{g_1, g_5\}$. Un exemple de 1-fermeture pour $\{g_1\}$ est $\{g_3, g_4\}$. Les règles d'association $\{g_1\} \Rightarrow \{g_3\}$ et $\{g_1\} \Rightarrow \{g_4\}$ ont seulement une exception.

La δ-liberté est une propriété anti-monotone telle qu'il est possible de calculer les ensembles δ-libre (éventuellement combinés avec une contrainte de fréquence minimale) dans des jeux de données très grands. Il est important de préciser que $h_0 \equiv f \circ g$, c'est-à-dire l'opérateur de fermeture classique. Chercher l'ensemble 0-libre X, et sa 0-fermeture Y, fournit l'ensemble fermé $X \cup Y$, et donc le concept formel $(g(X \cup Y, \mathbf{r}), X \cup Y)$.

Définition 2.4 (δ-bi-ensemble) *Un* δ*-*γ*-bi-ensemble* (T, G) *dans* $\mathbf{r}$ *est construit sur un ensemble* δ*-libre* $X \subset \mathcal{G}$ *avec* $T = g(X, \mathbf{r})$ *et* $G = h_\gamma(X, \mathbf{r})$*. Lorsque* $\delta = \gamma$ *nous l'appelons* δ*-bi-ensemble.*

Exemple. Dans le contexte de la Table 1.6, les 1-bi-ensembles dérivés des ensembles 1-libres $\{g_3\}$ et $\{g_5\}$ sont $(\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\})$ et $(\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\})$.

Lorsque $\delta << |T|$, les δ-bi-ensembles sont des ensembles denses avec un nombre petit d'exceptions par colonne.

Algorithme Pour l'extraction des bi-ensembles nous avons adapté une implémentation de l'algorithme MIN-EX décrit dans [BBR03]. Nous avons tout simplement ajouté la capacité de générer automatiquement l'ensemble d'objets qui supporte chaque ensemble δ-libre. MIN-EX est une instance typique de l'algorithme de recherche par niveaux présenté dans [MT97]. Grâce à l'anti-monotonicité de la conjonction de la contrainte de δ-liberté et de la contrainte de fréquence minimale, il explore le treillis des itemsets (par rapport à l'inclusion) niveau par niveau, en partant de l'ensemble vide et jusqu'au niveau de l'ensemble δ-libre fréquent le plus large. Plus précisément, la collection des candidats est initialisée avec l'ensemble vide comme seul membre (le seul ensemble de taille 0), et l'algorithme, à chaque itération, évalue les candidats (pour tester la δ-liberté et la fréquence minimale) et génère les candidats plus larges. À l'$i$ème itération, ils parcourt les données pour trouver quels candidats de taille i sont δ-libres fréquents, et il calcule leur δ-fermeture. Ensuite, il génère les candidats pour l'itération suivante, en prenant tout ensemble de taille $i+1$ tel que tous ses sous-ensembles stricts sont δ-libres fréquents. L'algorithme s'arrête lorsqu'il n'y a plus de candidat. Notre implémentation renvoie aussi l'ensemble des lignes qui supportent chaque ensemble δ-libre de colonnes découvert.

Propriétés Nous allons étudier quelques propriétés des δ-bi-ensembles. Il est clair que les 0-bi-ensembles sont les concepts formels. Toutefois, quelques propriétés importantes des concepts formels ne sont plus vérifiées pour les δ-bi-ensembles lorsque $\delta > 0$. En particulier, il nous manque une fonction qui associe l'ensemble G à l'ensemble T et vice versa. Par conséquent, nous n'avons plus de connexion de Galois. D'ailleurs, cela rend le processus d'interprétation (en terme de classification) moins naturel.

Exemple. Dans la Table 1.6, $(\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\})$ (le δ-bi-ensemble généré par l'ensemble δ-libre $\{g_5\}$), et $(\{t_2, t_5, t_6\}, \{g_2, g_5\})$ (le δ-bi-ensemble généré par l'ensemble δ-libre $\{g_2\}$) ont le même ensemble de propriétés, tandis que le premier ensemble d'objets inclut le deuxième.

Nous allons maintenant considérer comment les paramètres δ et γ influencent les propriétés de la collection des δ-γ-bi-ensembles.

Propriété 2.1 *Soit* $\mathbf{r}$ *un contexte booléen, soient* μ *et* δ *deux entiers positifs tels que* $\mu < \delta$. *On note avec* $Free_\delta(\mathbf{r})$ *la collection des ensembles* δ*-libres dans* $\mathbf{r}$, *et* $Free_\mu(\mathbf{r})$

la collection des ensembles μ-libres dans $\mathbf{r}$*, nous avons :*

$$Free_{\delta}(\gamma, \mathbf{r}) \subseteq Free_{\mu}(\gamma, \mathbf{r})$$

Preuve. X est un ensemble δ-libre ssi $\forall Y \subset X \ |\psi(Y, \mathbf{r})| - |\psi(X, \mathbf{r})| > \delta$. Alors $|\psi(Y, \mathbf{r})| - |\psi(X, \mathbf{r})| > \mu$ et X est aussi un ensemble μ-libre.

Par conséquent, chaque collection d'ensembles δ-libres ($\delta > 0$) est incluse dans la collection des ensembles 0-libres.

	g_1	g_2	g_3	g_4
t_1	0	1	1	1
t_2	0	1	1	0
t_3	1	0	0	0
t_4	1	1	1	0
t_5	1	1	1	1
t_6	1	0	1	0
t_7	0	0	1	0

TAB. 2.1 – Contexte booléen $\mathbf{r}\prime$

Exemple. Considérons le jeu de données booléennes de la Table 2.1. L'ensemble de propriétés $A = \{g_1, g_3\}$ est 0-libre (cf. Table 2.2), mais il n'est pas 1-libre (cf. Table 2.3). Sa 1-fermeture est $\{g_1, g_2, g_3\}$. Le δ-bi-ensemble correspondant est $b_A = (\{t_4, t_5, t_6\}, \{g_1, g_2, g_3\})$ avec une exception sur g_2. Comme A n'est pas dans la collection des ensembles 1-libres, ce bi-ensemble ne peut pas être construit en utilisant A, et nous n'avons aucun autre ensemble 1-libre qui peut générer b_A, ou aucun autre bi-ensemble qui couvre b_A (cf. Table 2.3).

X	$h_1(X, \mathbf{r}\prime)$	$g(X, \mathbf{r}\prime)$
$\{\emptyset\}$	$\{g_3\}$	$\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$
$\{g_1\}$	$\{g_1, g_3\}$	$\{t_3, t_4, t_5, t_6\}$
$\{g_2\}$	$\{g_2, g_3\}$	$\{t_1, t_2, t_4, t_5\}$
$\{g_3\}$	$\{g_3\}$	$\{t_1, t_2, t_4, t_5, t_6, t_7\}$
$\{g_4\}$	$\{g_1, g_2, g_3, g_4\}$	$\{t_1, t_5\}$
$\{g_1, g_2\}$	$\{g_1, g_2, g_3, g_4\}$	$\{t_4, t_5\}$
$\{g_1, g_3\}$	$\{g_1, g_2, g_3\}$	$\{t_4, t_5, t_6\}$
$\{g_1, g_4\}$	$\{g_1, g_2, g_3, g_4\}$	$\{t_5\}$

TAB. 2.2 – Ensembles 0-libres, 1-fermeture et ensembles d'objets de support dans $\mathbf{r}\prime$

Propriété 2.2 *Soit* $\mathbf{r}$ *un contexte booléen, soient* ρ *et* γ *deux entiers positifs tels que* $\rho \leq \gamma$. *Soit* $X \subseteq \mathcal{P}$ *un ensemble, nous avons :*

$$h_\rho(X) \subseteq h_\gamma(X)$$

Lorsque le paramètre γ croît, la composante des attributs du δ-γ-bi-ensemble croît aussi.

Propriété 2.3 *Soit* X *un ensemble* δ*-libre,* $\forall Y \subset X$*, alors* $X \not\subseteq h_\delta(Y, \mathbf{r})$*, i.e.,* X *n'est pas inclus dans la* δ*-fermeture d'aucun de ses sous-ensembles stricts.*

Preuve. Si $Y \subset X$, et $X \subseteq h_\delta(Y, \mathbf{r})$, alors il existe $Z \subset X$, $Z \cap Y = \emptyset$, tel que $Y \Rightarrow Z$ est une règle δ-forte, i.e., il existe une règle δ-forte entre deux sous-ensembles stricts de X (Y et Z), mais cela entre en contradiction avec le fait que X est un ensemble δ-libre.

Par conséquent, lorsque $\gamma > \delta$, un ensemble X peut appartenir à la γ-fermeture d'un de ses sous-ensembles stricts, tandis que cela n'est pas vrai lorsque $\gamma \leq \delta$.

X	$h_1(X, \mathbf{r\prime})$	$g(X, \mathbf{r\prime})$
$\{\emptyset\}$	$\{g_3\}$	$\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$
$\{g_1\}$	$\{g_1, g_3\}$	$\{t_3, t_4, t_5, t_6\}$
$\{g_2\}$	$\{g_2, g_3\}$	$\{t_1, t_2, t_4, t_5\}$
$\{g_4\}$	$\{g_1, g_2, g_3, g_4\}$	$\{t_1, t_5\}$
$\{g_1, g_2\}$	$\{g_1, g_2, g_3, g_4\}$	$\{t_4, t_5\}$

TAB. 2.3 – Ensembles 1-libres, 1-fermetures et ensembles d'objets de support dans $\mathbf{r\prime}$

Exemple. Dans le contexte de la Table 1.6, nous avons cinq ensembles 1-libres : $\{\emptyset\}$, $\{g_1\}$, $\{g_2\}$, $\{g_4\}$ et $\{g_1, g_2\}$. La collection des ensembles 0-libres contient deux autres ensembles $\{g_1, g_3\}$ et $\{g_1, g_4\}$, qui sont contenus dans la 1-fermeture de $\{g_1\}$ (i.e., $\{g_1, g_3, g_4\}$ qui est aussi la 1-fermeture de $\{g_3\}$ et $\{g_4\}$). L'ensemble d'objets du support pour $\{g_1, g_3\}$ et $\{g_1, g_4\}$ est $\{t_1, t_3\}$ et il s'agit d'un sous-ensemble de l'ensemble d'objets composant le support de $\{g_1\}$ (i.e., $\{t_1, t_3, t_5\}$), ainsi que de l'ensemble d'objets du support pour $\{g_3\}$ et $\{g_4\}$. En effet, les deux 0-δ-bi-ensembles sont déjà inclus dans le bi-ensemble plus large obtenu à partir des ensembles 1-libres.

Pour résumer, nous avons considéré plusieurs paramètres alternatifs pour calculer les δ-bi-ensembles. Comme la γ-fermeture d'un ensemble 0-libre X est égale à la γ-fermeture de $h_0(X, \mathbf{r})$, en calculant les 0-bi-ensembles nous avons soit les concepts formels (0-fermeture d'un ensemble 0-libre), soit leur extension vers la tolérance au

bruit bornée par colonne. Calculer les concepts formels à partir des ensembles libres et de leur fermeture peut se révéler intraitable dans certaines jeux de données, alors que l'extraction des ensembles δ-libres pour $\delta > 0$ reste faisable à condition de perdre certaines associations. D'un autre coté, utiliser une valeur de δ supérieure à γ peut entraîner la perte ultérieure d'information, même si la taille de la collection des bi-ensembles produits pourrait se réduire. La réponse à la question que nous nous sommes posés précédemment, est qu'utiliser la même valeur de δ pour calculer les ensembles libres et leur fermeture est un bon compromis pour préserver l'information et pour réduire à la fois l'espace de recherche et la taille de la collection des bi-ensembles extraite.

2.3.2 Utilisation des règles d'association

Les règles d'association peuvent être dérivées à partir des bi-ensembles extraits et utilisées pour la caractérisation des bi-clusters. Pour la caractérisation mais aussi pour la classification, des heuristiques ont été étudiées afin de sélectionner des règles d'association pertinentes selon les valeurs de leur fréquence et de leur confiance [LHM98, LHP01, RCB02]. Dans notre cas, nous proposons l'utilisation de la proportion d'exceptions sur les bi-ensembles extraits pour produire des règles de caractérisation. Elles sont de la forme $X \Rightarrow k$, où X est un ensemble de propriétés (resp., objets) et k est une propriété qui dénote une classe d'objets (resp. une propriété qui dénote une classe de propriétés). Lorsque on considère des concepts formels, dériver des règles de caractérisation est facile.

Propriété 2.4 *Soit $(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})$ un bi-cluster. Si (T, G) est un concept formel, alors $G \Rightarrow k$ (resp. $T \Rightarrow k$) est une règle de fréquence égale à $|T| \cdot (1 - \epsilon_t(T, P_k^{\mathcal{T}}))$ (resp. $|G| \cdot (1 - \epsilon_g(G, P_k^{\mathcal{G}}))$ et de confiance égale à $1 - \epsilon_t(T, P_k^{\mathcal{T}})$ (resp. $1 - \epsilon_g(G, P_k^{\mathcal{G}})$).*

	g_1	g_2	g_3	g_4	g_5	p_1^t	p_2^t
t_1	1	0	1	1	0	1	0
t_2	0	1	0	0	1	0	1
t_3	1	0	1	1	0	1	0
t_4	0	0	1	1	0	1	0
t_5	1	1	0	0	1	0	1
t_6	0	1	0	0	1	0	1
t_7	0	0	0	0	1	0	1
p_1^g	1	0	1	1	0		
p_2^g	0	1	0	0	1		

TAB. 2.4 – Un contexte booléen $\mathbf{r}_1$

Exemple. Dans notre exemple jouet (Table 1.6), nous ajoutons deux colonnes correspondant aux deux valeurs de la variable de classe sur les objets $v_t \in \{p_1^t, p_2^t\}$

et deux lignes correspondant à la variable de la classe des propriétés $v_t \in \{p_1^g, p_2^g\}$. Pour chaque objet appartenant à $P_1^{\mathcal{T}}$ (resp. $P_2^{\mathcal{T}}$), nous avons $p_1^t = 1$ et $p_2^t = 0$ (resp. $p_1^t = 0$ and $p_2^t = 1$). Nous obtenons le contexte booléen de la Table 2.4. Le bi-ensemble $b_1 = (T_1, G_1) = (\{t_1, t_3, t_5\}, \{g_1\})$ est un concept formel, il peut donc être utilisé pour construire la règle d'association $g_1 \Rightarrow p_1^t$. Sa fréquence relative est $|T_1| \cdot (1 - \epsilon_t(T_1, P_1^{\mathcal{T}}))/|\mathcal{T}| = 3 \cdot (1 - 1/3)/7 = 29\%$; sa confiance est $(1 - \epsilon_t(T_1, P_1^{\mathcal{T}})) = (1 - 1/3) = 67\%$.

Le concept formel $b_2 = (T_2, G_2) = (\{t_5\}, \{g_1, g_2, g_5\})$ forme la règle d'association $t_5 \Rightarrow p_2^g$. Sa fréquence relative est $|G_2| \cdot (1 - \epsilon_g(G_2, P_2^{\mathcal{G}}))/|\mathcal{G}| = 3 \cdot (1 - 1/3)/5 = 40\%$; sa confiance est $(1 - \epsilon_g(G_2, P_2^{\mathcal{G}})) = (1 - 1/3) = 67\%$

Lorsque l'on utilise les δ-bi-ensembles au lieu des concepts formels la propriété 2.4 n'est plus vraie car $|g(G, \mathbf{r})| < |T|$. En revanche, si on s'intéresse à la caractérisation d'une classe d'objets, on peut utiliser la propriété suivante :

Propriété 2.5 *Soit $P_k^{\mathcal{T}}$ une classe, (T, G) un δ-bi-ensemble, et $X \subseteq G$ un ensemble δ-libre, alors $X \Rightarrow k$ est une règle d'association dont la fréquence est égale à $|T| \cdot (1 - \epsilon_o(T, P_k^{\mathcal{T}}))$ et la confiance est égale à $1 - \epsilon_o(T, P_k^{\mathcal{T}})$.*

Exemple. Considérons notre exemple jouet dans la Table 1.6, et son extension (Table 2.4). Le bi-ensemble $b_\delta = (T_\delta, G_\delta) = (\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\})$ est un 1-bi-ensemble généré par l'ensemble 1-libre $X_\delta = \{g_3\}$. Il est associé à $(P_1^{\mathcal{T}}, P_1^{\mathcal{G}})$ car $sim(b_\delta, (P_1^{\mathcal{T}}, P_1^{\mathcal{G}})) = 1$, et $\epsilon_o(T_\delta, P_1^{\mathcal{T}}) = 0$. Alors, X_δ forme une règle d'association $g_3 \Rightarrow p_1^t$ avec une fréquence relative $|T_\delta| \cdot (1 - 0)/|\mathcal{T}| = 43\%$ et confiance de 100%.

De telles règles sont intéressantes en pratique parce que X est souvent un ensemble très petit, et donc son interprétation est simple. Cependant, cette approche ne peut pas être appliquée aux jeux de données avec un grand nombre de propriétés (e.g., pour les jeux de données d'expression où nous avons des milliers de propriétés). Dans ce cas, nous proposons d'utiliser les mesures ϵ_t et ϵ_g. Il faut toutefois noter qu'un travail récent a étudié l'extraction des ensembles δ-libres lorsque le nombre de propriétés [HC05] est très grand.

2.4 Exemples de requêtes

Note approche suggère l'utilisation de requêtes de caractérisation, c'est-à-dire, des requêtes dans lesquelles l'analyste peut utiliser les mesures de précision proposées pour sélectionner des motifs caractérisants pertinents. Des exemples de requêtes de caractérisation typiques pourraient être les suivants :

– Sélectionner tous les bi-ensembles qui caractérisent le bi-cluster $(P^{\mathcal{T}}, P^{\mathcal{G}})$ avec un rapport d'exception maximal ε soit pour les objets, soit pour les propriétés.

– Sélectionner toutes les règles d'association avec une partie gauche minimale qui caractérisent le bi-cluster $(P^{\mathcal{T}}, P^{\mathcal{G}})$ avec une fréquence minimale f, une confiance minimale c, et avec un rapport d'exception maximal de ε pour l'ensemble des propriétés.
– Sélectionner toutes les règles d'association avec une partie gauche minimale qui caractérisent le bi-cluster $(P^{\mathcal{T}}, P^{\mathcal{G}})$ avec une fréquence minimale f, une confiance minimale c, et avec un rapport d'exception **minimal** de ε pour l'ensemble des propriétés.

Le deux premiers types de requêtes sont très utiles pour la caractérisation des bi-clusters. Le troisième type concerne plus la découverte de connaissances grâce à son apport potentiel d'information inattendue. En effet, il peut produire des motifs qui sont des exceptions, c'est-à-dire des motifs qui concernent des objets appartenant au bi-cluster $(P^{\mathcal{T}}, P^{\mathcal{G}})$ qui sont caractérisés par des propriétés appartenant à d'autres bi-clusters.

2.5 Validation de la caractérisation

Dans cette section nous allons présenter les résultats concernant l'application de notre méthode de caractérisation sur différents jeux de données. En particulier, nous avons testé notre approche sur des données benchmark classiques, sur un jeu de données médicales et sur un jeu de données d'expression.

2.5.1 Caractérisation d'un jeu de données benchmark

Nous avons d'abord appliqué notre méthode de caractérisation sur le jeu de donnée benchmark très connu voting-records [BM98]. Il contient 435 objets et 48 attributs booléens (en enlevant la variable de classe). Nous avons utilisé Cocluster [DMM03] pour obtenir deux bi-clusters :

bi-cluster	$\lvert\boldsymbol{\tau}\rvert$	rep.	dem.	$\lvert\boldsymbol{\gamma}\rvert$
bi-cluster1	193	153	40	16
bi-cluster2	242	15	227	32
total	435	168	267	48

Pour caractériser chaque bi-cluster, nous avons utilisé D-Miner [BRB04a] pour extraire tous les concepts formels, et notre extension d'AcMiner pour extraire deux collections de δ-bi-ensembles (δ=1,2). Nous avons obtenu 227 031 concepts formels, 130 313 1-bi-ensembles et 66 908 2-bi-ensembles. Les collections ont été post-traitées pour chercher des règles avec des valeurs croissantes de fréquence relative minimale (15% à 40%) et de confiance (90% à 100%). Les résultats pour le premier bi-cluster

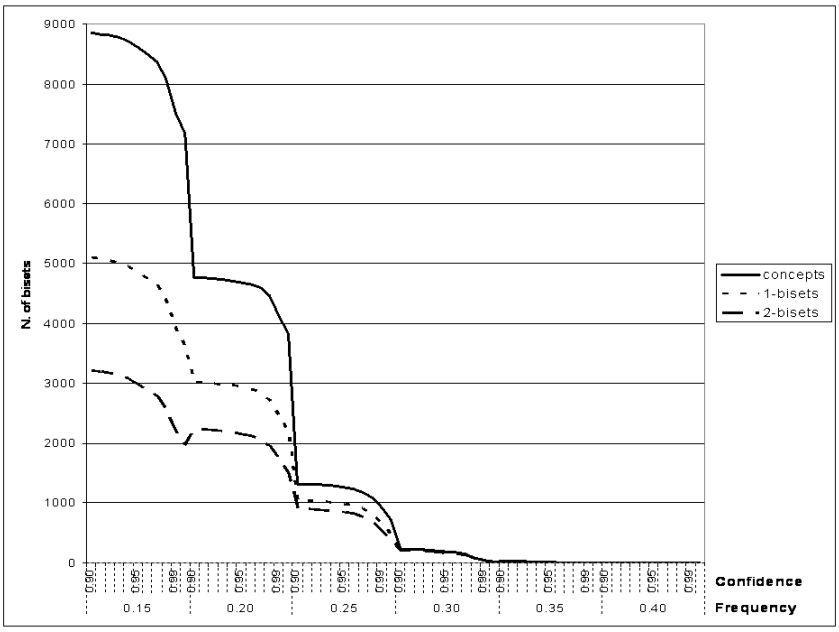


FIG. 2.1 – Motifs caractérisant le bi-cluster1 dans voting-records par rapport à des valeurs différentes des seuils de fréquence et confiance minimales

sont dans la Figure 2.1. Les résultats pour le second bi-cluster sont similaires. Le nombre de règles de caractérisation décroît lorsqu'on augmente les seuils de fréquence et de confiance. Lorsqu'on utilise les δ-bi-ensembles, on se retrouve avec des collections significativement plus petites à traiter. Deux exemples de règles de caractérisation qui sont consistantes avec la connaissance du domaine associé à voting-records sont données ici. La première (resp. la deuxième) a une fréquence relative de 42% (resp. 31%) et les deux ont une confiance de 100%, c'est-à-dire, nous avons $\epsilon_t = 0$.

`el-salvador-aid = yes` $\wedge$ `anti-satellite-test-ban = yes`

$\wedge$ `aid-to-nicaraguan-contras = yes` $\Rightarrow$ `bi-cluster2`

`handicapped-infants = no` $\wedge$ `physician-fee-freeze = yes`

$\wedge$ `el-salvador-aid = yes` $\Rightarrow$ `bi-cluster1`

2.5.2 Caractérisation d'un jeu de données médicales

Nous avons appliqué la méthode au jeu de données médicales réelles meningitis déjà utilisé dans [RCB02]. Il a été recueilli sur des enfants hospitalisés pour une forme aiguë de méningite. Les données booléennes pré-traitées sont constituées

de 329 patients décrits par 60 attributs booléens qui codent des signes cliniques (troubles hémodynamiques, trouble de conscience, ...), l'analyse cytochimique du fluide cérébro-spinal (protéines C.S.F., glucose C.S.F., ...), et l'analyse du sang (taux de sédimentation, nombre de globules blancs, ...) des malades. Dans meningitis, la plupart des cas sont identifiés comme infections virales, tandis qu'environ un quart sont reconnus causés par des bactéries. De plus, on dispose de connaissances médicales qui peuvent être utilisées pour valider la pertinence de la caractérisation. En utilisant COCLUSTER, nous avons obtenu deux bi-clusters :

bi-cluster	$\lvert\boldsymbol{\tau}\rvert$	bact.	vir.	$\lvert\boldsymbol{\gamma}\rvert$
bi-cluster1	100	81	19	21
bi-cluster2	229	3	226	39
total	329	84	245	60

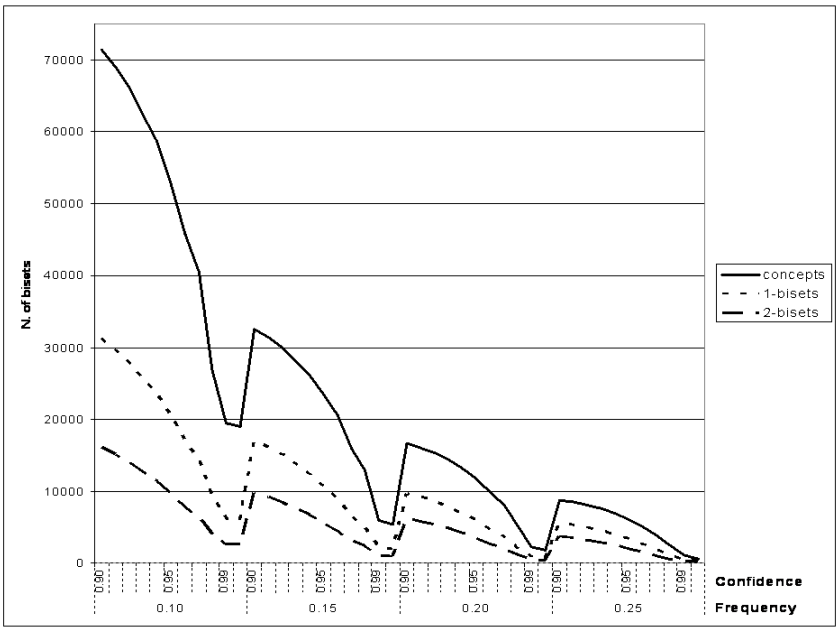


FIG. 2.2 – Motifs caractérisant le bi-cluster1 dans meningitis par rapport à des valeurs différentes des seuils de fréquence et confiance minimales

Le premier bi-cluster contient une majorité de cas bactériens, tandis que le second contient presque uniquement des cas viraux. Nous avons sélectionné des règles de caractérisation basées sur une collection de concepts formels et deux collections de δ-bi-ensembles (δ=1,2). Nous avons obtenu les résultats de la Figure 2.2. Encore une fois, l'utilisation des δ-bi-ensembles conduit à des collections plus petites de motifs candidats. Le nombre de règles caractérisant le premier bi-cluster est toujours très

petit, et il ne change pas de façon significative lorsqu'on utilise les δ-bi-ensembles au lieu des concepts formels. Si on sélectionne les règles avec un corps minimal, un seuil de fréquence de 10%, un seuil de confiance de 98%, et telles que le rapport d'exception ϵ_g est égal zéro, nous obtenons seulement 9 règles qui sont consistantes avec la connaissance médicale (cf. [RCB02] pour plus de détail). Des exemples de règles sont :

$$\texttt{presence of bacteria in C.S.F. analysis = yes} \Rightarrow \texttt{bi-cluster1}$$

$$\texttt{polynuclear percent} > 80 \wedge \texttt{C.S.F. proteins} > 0.8 \Rightarrow \texttt{bi-cluster1}$$

$$\texttt{C.S.F. proteins} > 0.8 \wedge \texttt{C.S.F. glucose} < 1.5 \Rightarrow \texttt{bi-cluster1}$$

2.5.3 Caractérisation d'un jeu de données d'expression

Nous avons effectué une expérience sur un jeu de données puces à ADN (malaria [BLP+03]) qui concerne le transcriptome du cycle de développement intraerythrocytique du Plasmodium Falciparum, i.e. un agent responsable de la malaria humaine. Les données fournissent le profil d'expression de 3 719 gènes dans 46 échantillons biologiques. Chaque échantillon correspond à un instant de temps du cycle de développement : il commence avec l'invasion des globules rouges du sang par le mérozoïte, et il est divisé en trois phases : anneau, trophozoïte et schizonte (concernant respectivement le moustique, le foie, et le sang). Après 48 heures, la cellule se réplique et se divise. Aux instants 17h et 29h il y a deux transitions brusques. Les données d'expression numériques présentées dans [BLP+03] ont été discretisées en utilisant une des méthodes de codage des propriété décrites dans [BBJ+02] : pour chaque gène g, nous avons affecté la valeurs booléenne 1 aux échantillons dont le niveau d'expression était supérieur à X% de la valeur maximale. Nous avons choisi X=25%. Nous donnerons plus de détails dans le prochain chapitre. Nous avons utilisé Cocluster pour obtenir les bi-clusters suivants :

bi-cluster	$\lvert\boldsymbol{\tau}\rvert$	ring	troph	schiz.	$\lvert\boldsymbol{\gamma}\rvert$
bi-cluster1	20	15	5	0	558
bi-cluster2	16	0	5	11	1699
bi-cluster3	10	6	0	4	1462
total	46	21	10	15	3719

Nous avons extrait des collections de bi-ensembles pour caractériser les classes des échantillons biologiques avec des ensembles de gènes. Dans ce cas, cependant, le nombre de propriétés (colonnes) est trop grand pour être traité, et nous avons extrait la collection des δ-bi-ensembles dans la matrice transposée. Il est clair que la fréquence et la confiance n'ont plus de sens car elles sont calculées sur les ensembles d'échantillons, alors que nous cherchons des ensembles de gènes. Nous avons donc

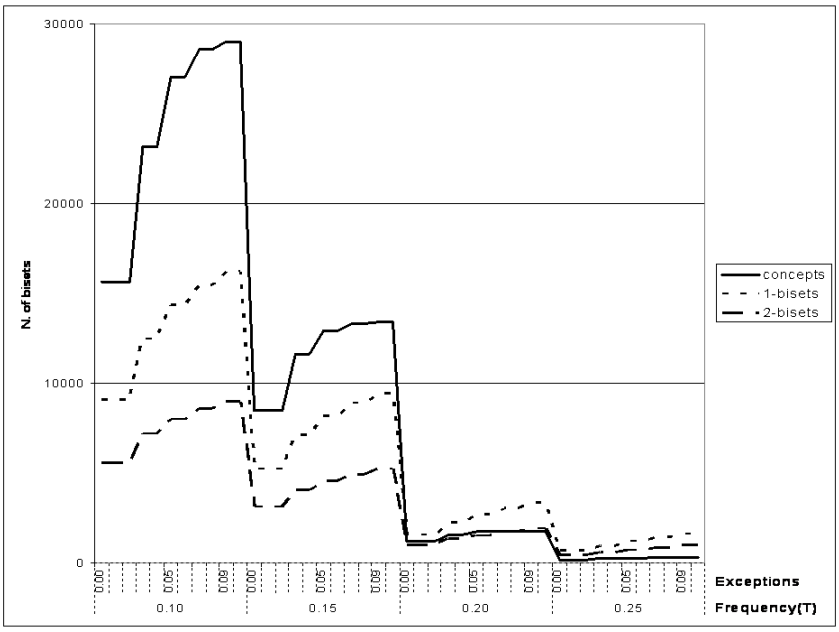


FIG. 2.3 – Motifs caractérisant le bi-cluster1 dans malaria par rapport à des valeurs différentes de la taille minimale et du rapport d'exception maximal

utilisé les tailles des bi-ensembles $|T|$ et $|G|$, et leurs rapports d'exception ϵ_t et ϵ_g. Les résultats pour une taille minimale de 10% à 25% de $|\mathcal{T}|$ et pour des valeurs maximales de ϵ_t de 0% jusqu'à 10% sont dans la Figure 2.3.

Si on considère le bi-cluster1, on peut analyser les 2-bi-ensembles dont la taille minimale de leurs ensembles d'objets est 25% de $|\mathcal{T}|$ et dont le rapport d'exception maximal $\varepsilon_t = 0$. Parmi le 442 bi-ensembles caractérisant le bi-cluster1, seuls 4 d'entre eux concernent des gènes qui appartiennent au même bi-cluster. Dans chacun d'entre eux, nous avons repéré au moins un gène appartenant au groupe "cytoplasmic translation machinery", qui est connu pour être actif dans la phase anneau (cf. [BLP+03] pour plus de détails), i.e., l'étape du développement majoritairement représentée dans le bi-cluster1.

2.6 Conclusion

Nous avons présenté une nouvelle méthode de caractérisation basée sur les motifs locaux extraits et, plus précisément, sur les concepts formels et les δ-bi-ensembles. Aujourd'hui il est possible d'utiliser des techniques efficaces pour l'extraction de différents types de motifs locaux sous contraintes. Si une bi-partition fournit déjà une

caractérisation globale et, en quelque sorte, attendue, les collections sélectionnées des bi-ensembles de caractérisation mettent en évidence des associations locales qui pourraient apporter une information moins attendue, mais toujours pertinente. Autant les motifs locaux que les motifs globaux sont utiles pendant les processus de découverte de connaissances, et il est important d'aider ces processus intrinsèquement interactifs.

Si un motif global tel qu'une bi-partition capture des structures intéressantes dans les données, il semblerait aussi intéressant de regarder la collection des associations locales qui sont en quelque sorte, loin de celle-ci. Si on fait l'hypothèse que l'association populaire R, qui met en évidence des transactions fréquentes avec "bière" et "couches" parmi les client mâles d'un supermarché, est valide, une co-classification sur un jeu de données de paniers complet pourrait regrouper la bière avec les client mâles dans un bi-cluster, et les couches avec les clients femelles dans un deuxième bi-cluster. Dans ce cas, une requête qui sélectionnerait des règles d'association fréquentes et à forte confiance avec un rapport d'exception élevé sur les propriétés ($\epsilon > \varepsilon$), pourrait aider la découverte de cette association "inattendue" dans R.

Dans le prochain chapitre, nous allons développer l'autre aspect important du cadre L2G, qui est, en quelque sorte, complémentaire de celui que nous avons présenté ici. Il s'agit d'utiliser les motifs locaux non pas pour la caractérisation, mais pour la construction même de la bi-partition.

Chapitre 3

Co-classification à partir de motifs locaux

3.1 Introduction

Dans le chapitre précédent, nous avons montré que les motifs locaux peuvent apporter une contribution significative à la phase d'interprétation d'une bi-partition. Dans ce chapitre nous explorons le cadre que nous avons appelé "L2G" [PRB05].

Nous proposons d'utiliser des motifs locaux pour améliorer la qualité d'une bi-partition. Notre idée de base est qu'une collection de bi-ensembles (par exemple de concepts formels), contient de l'information de valeur améliorer l'optimisation de la fonction objectif dont l'optimisation correspond à une bonne bi-partition. Dans la littérature, on trouve des algorithmes pour extraire efficacement tous les concepts formels, où une sous-collection de concepts ayant une surface ($|T| \times |G|$) supérieure à un seuil [BRBR05][1].

Un contrôle de la part de l'utilisateur sur le chevauchement entre bi-clusters semble aussi important. D'un coté, la plupart des algorithmes connus de co-classification calculent des collections de bi-clusters non chevauchants. D'un autre coté, nombre de domaines d'application pourraient bénéficier des bi-clusters chevauchants. C'est le cas, par exemple, de l'analyse des données d'expression de gènes, où les bi-clusters mettent en évidence des ensembles de gènes qui ont tendance à être co-exprimés, et des ensembles de situations biologiques qui semblent réguler cette co-expression. Sous l'hypothèse raisonnable qu'une fonction biologique peut être assignée théoriquement à chaque bi-cluster, la découverte de bi-clusters chevauchants est importante, car nous savons qu'un même gène peut participer à plusieurs fonctions biologiques différentes.

[1] Les collections de concepts formels sont en général énormes, e.g., [BRBR05] reporte l'extraction de plusieurs millions de concepts formels dans une matrice booléenne relativement petite.

De la même manière, dans un contexte de fouille de données d'utilisation du web, les bi-clusters des utilisateurs associés aux ressources qu'ils téléchargent à partir de certains sites internet, peuvent être utilisés pour découvrir des groupes intéressants et, ici encore, on peut capturer des groupes plus pertinents si le chevauchement est permis.

Notre contribution dans ce chapitre consiste à définir un nouveau cadre de co-classification. Il permet de calculer des bi-partitions en regroupant des motifs locaux qui capturent des associations localement fortes entre les objets et les propriétés, i.e., des bi-ensembles qui satisfont certaines contraintes définies par l'utilisateur. Des types différents de motifs locaux sont candidats pour un tel processus, e.g., itemsets fréquents associés aux ensembles d'objets les supportant, concepts formels, δ-bi-sets (cf. chapitre précédent).

Deuxièmement, nous étudions une instance de notre cadre, l'algorithm CDK-Means, qui construit simultanément deux partitions liées, l'une d'objets, et l'autre de propriétés. Plus précisément, au lieu de partitionner directement l'ensembles des objets et l'ensemble des propriétés, nous appliquons un algorithme de type *k-means* à une collection de bi-ensembles. Comme résultat, les objets et les propriétés sont associés intrinsèquement aux classes, selon leur poids dans les centroïdes calculés à la fin. Une validation expérimentale complète, analysant les différents aspects de notre approche, est présentée dans la dernière partie du chapitre.

3.2 Un cadre générique

Nous définissons ici un nouveau modèle de co-classification de données catégorielles sous la forme de matrices booléennes (éventuellement grandes). Intuitivement, étant donné un contexte booléen et une collection de bi-ensembles qui capturent des associations localement fortes à l'intérieur des données, ce cadre permet de construire une partition de K classes de bi-ensembles et successivement de produire une collection de bi-clusters (éventuellement chevauchant) d'objets et de propriétés. Les principes sont illustrés via un algorithme de type *k-means* appliqué aux bi-ensembles. La nature des bi-ensembles n'est pas spécifiée, mais il est clair qu'un type de bi-ensemble candidat qui se prête bien à cet objectif est le concept formel (et son extension naturelle, c'est-à-dire le δ-bi-ensemble).

Nous allons d'abord présenter le cadre générique. Ensuite nous allons décrire une instance particulière du cadre, qui est réalisée avec un algorithme de type *k-means*.

3.2.1 Une approche local-vers-global (L2G)

Notre approche est en quelque sorte l'envers du processus de caractérisation que nous avons présenté dans le chapitre précédent. La tâche de co-classification que nous allons présenter, peut être définie de la façon suivante : nous cherchons à calculer une partition de K classes d'objets (notées $\{P_1^{\mathcal{T}} \dots P_K^{\mathcal{T}}\}$) et une partition de K classes de propriétés (notée $\{P_1^{\mathcal{G}} \dots P_K^{\mathcal{G}}\}$) avec une fonction bijective entre les deux partitions, telle que chaque classe d'objets soit caractérisée par une seule classe de propriétés (et vice versa). Notre idée est que la bi-partition peut être calculée à partir de bi-ensembles et en particulier à partir de bi-ensembles maximaux de 1 (comme les concepts formels) ou de bi-ensembles denses (comme les δ-bi-ensembles).

Nous faisons l'hypothèse qu'une collection de N bi-ensembles *a priori* intéressants (notée $\mathcal{B} = \{b_1, \dots, b_N\}$) a été extraite dans $\mathbf{r}$ précédemment. Nous décrivons chaque bi-ensemble $b_i = (T_i, G_i)$ ($T_i \subseteq \mathcal{T}$, $G_i \subseteq \mathcal{G}$) avec le vecteur booléen

$$< \mathbf{t}_i >, < \mathbf{g}_i > = < t_{i1}, \dots, t_{im} >, < g_{i1}, \dots, g_{in} >$$

où $t_{ij} = 1$ if $t_j \in T_i$ (0 sinon) and $g_{ij} = 1$ if $g_j \in G_i$ (0 sinon).

Nous cherchons K classes de bi-ensembles $\{B_1, \dots, B_K\}$ ($B_k \subseteq \mathcal{B}$). Nous définissons le centroïde d'une classe de bi-ensembles C_k comme :

$$\mu_k = < \tau_k >, < \gamma_k > = < \tau_{k1}, \dots, \tau_{km} >, < \gamma_{k1}, \dots, \gamma_{kn} >$$

où τ and γ sont les composantes habituelles d'un centroïde :

$$\tau_{kj} = \frac{1}{|B_k|} \sum_{b_i \in B_k} t_{ij}, \quad \gamma_{kj} = \frac{1}{|B_k|} \sum_{b_i \in B_k} g_{ij}$$

où $|B_k|$ est le cardinal de B_k.

L'objectif (de tout algorithme de classification) est de maximiser la similarité intra-classe et la dissimilarité inter-classe. Les bi-ensembles définissent des rectangles dans la matrice (à des permutations de lignes et colonnes près), donc, pour mesurer la similarité intra-classe, une solution "naturelle" serait de mesurer l'intersection entre un bi-ensemble et un centroïde. On pourrait utiliser la mesure présentée dans le chapitre précédent, mais dans ce cas, son calcul pourrait se révéler très coûteux. D'un autre côté, nous devons prendre en compte la valeur des composantes du centroïde. Ici nous allons considérer chaque bi-ensemble comme un parallélépipède dont un côté représente les objets et l'autre les propriétés, et dont la hauteur vaut 1 si l'objet et la propriété sont en relation dans $\mathbf{r}$. Un centroïde est un parallélépipède dont la hauteur est proportionnelle au nombre de bi-ensembles qui contiennent une association entre la ligne et la colonne (cf. Figure 3.1) et varie entre 0 et 1.

Nous définissons donc la distance entre un bi-ensemble et un centroïde de la manière suivante :

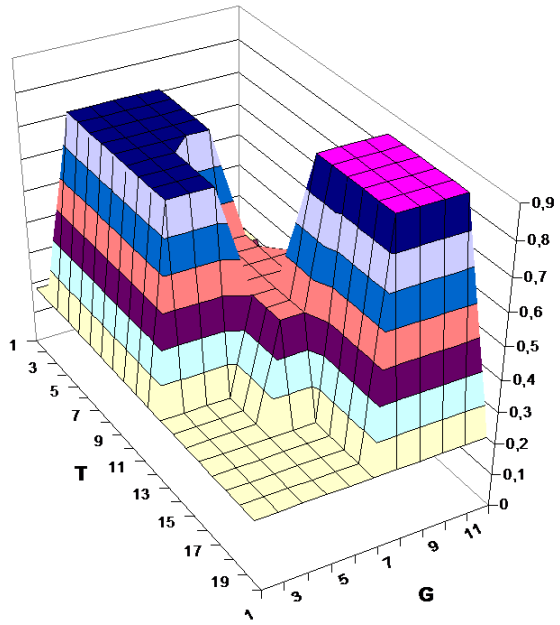


FIG. 3.1 – Exemple de représentation tridimensionnelle d'un centroïde

$$d(b_i, \mu_k) = \frac{1}{2}\left(\frac{|\mathbf{t}_i \cup \boldsymbol{\tau}_k| - |\mathbf{t}_i \cap \boldsymbol{\tau}_k|}{|\mathbf{t}_i \cup \boldsymbol{\tau}_k|} + \frac{|\mathbf{g}_i \cup \boldsymbol{\gamma}_k| - |\mathbf{g}_i \cap \boldsymbol{\gamma}_k|}{|\mathbf{g}_i \cup \boldsymbol{\gamma}_k|}\right)$$

Il s'agit de la moyenne des différences symétriques pondérées des composantes du bi-ensemble et de celles du centroïde. Les tailles pondérées de l'intersection et l'union entre les objets du centroïde et les objets du bi-ensemble b_i sont définies de la manière suivante :

$$|\mathbf{t}_i \cap \boldsymbol{\tau}_k| = \sum_{j=1}^{m} a_j \frac{t_{ij} + \tau_{kj}}{2}, \quad |\mathbf{t}_i \cup \boldsymbol{\tau}_k| = \sum_{j=1}^{m} \frac{t_{ij} + \tau_{kj}}{2}$$

où $a_j = 1$ si $t_{ij} \cdot \tau_{kj} \neq 0$, 0 sinon.

Intuitivement, l'intersection est égale à la moyenne entre le nombre d'objets communs, et la somme des poids de leur centroïde. L'union est la moyenne entre le nombre d'objets, et la somme des poids de leur centroïde. La taille pondérée de l'intersection et de l'union pour les propriétés est définie d'une manière similaire.

Nous considérons que les K centroïdes sont ceux qui optimisent la similarité intra-classe et la dissimilarité inter-classe. L'étape suivante consiste donc à construire une bi-partition sur la base de la partition de bi-ensembles. Notre solution consiste à assigner les objets t_j (resp. les propriétés g_j) à l'une des K classes (notée k) telle que

b_i	$< t_{i1}, t_{i2}, t_{i3}, t_{i4}, t_{i5}, t_{i6}, t_{i7} >$	,	$< g_{i1}, g_{i2}, g_{i3}, g_{i4}, g_{i5} >$
b_1	$< 1, 1, 1, 1, 1, 1, 1 >$	,	$< 0, 0, 0, 0, 0 >$
b_2	$< 1, 0, 1, 0, 1, 0, 0 >$	,	$< 1, 0, 0, 0, 0 >$
b_3	$< 0, 1, 0, 0, 1, 1, 1 >$	,	$< 0, 0, 0, 0, 1 >$
b_4	$< 0, 1, 0, 0, 1, 1, 0 >$	,	$< 0, 1, 0, 0, 1 >$
b_5	$< 1, 0, 1, 1, 0, 0, 0 >$	,	$< 0, 0, 1, 1, 0 >$
b_6	$< 0, 0, 0, 0, 1, 0, 0 >$	,	$< 1, 1, 0, 0, 1 >$
b_7	$< 1, 0, 1, 0, 0, 0, 0 >$	,	$< 1, 0, 1, 1, 0 >$
b_8	$< 0, 0, 0, 0, 0, 0, 0 >$	,	$< 1, 1, 1, 1, 1 >$

TAB. 3.1 – Liste des vecteurs correspondant au 8 concepts formels de la Table 1.4

μ_i	$< \tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}, \tau_{i5}, \tau_{i6}, \tau_{i7} >$	,	$< \gamma_{i1}, \gamma_{i2}, \gamma_{i3}, \gamma_{i4}, \gamma_{i5} >$
μ_1	$< 0.80, 0.20, 0.80, 0.40, 0.40, 0.20, 0.20 >$	,	$< 0.60, 0.20, 0.60, 0.60, 0.20 >$
μ_2	$< 0.00, 0.67, 0.00, 0.00, 1.00, 0.67, 0.33 >$	,	$< 0.33, 0.67, 0.00, 0.00, 1.00 >$

TAB. 3.2 – Liste des vecteurs correspondants à deux classes possibles de concepts formels

τ_{kj} (resp. γ_{kj}) soit maximum. Nous soulignons le fait que les valeurs des composantes dans les deux vecteurs dépendent exclusivement du nombre de bi-ensembles impliqués dans le centroïde.

Exemple. Les vecteurs booléens correspondant à la collection de bi-ensembles de la Table 1.4 (contenus dans le contexte formel de la Table 1.6) sont dans la Table 3.1. Une solution possible pour $K = 2$, est donnée par les deux classes de bi-ensembles $B_1 = \{b_1, b_3, b_4, b_6, b_8\}$ et $B_2 = \{b_2, b_5, b_7\}$. Les vecteurs booléens μ_1 et μ_2 correspondant à ces deux classes sont dans la Table 3.2.

En post-traitant ces vecteurs, chaque objet t_j (resp. propriété g_j) est assigné à la classe P_k^{CT} (resp. P_k^{CG}) telle que τ_{kj} (resp. γ_{kj}) est maximal. Par exemple, l'objet t_1 est assigné à $P_1^{\mathcal{T}}$, car $\max\{\tau_{11}, \tau_{21}\} = \tau_{11} = 0.80$. De manière similaire, la propriété g_1 est assignée à $P_1^{\mathcal{G}}$, car $\max\{\gamma_{11}, \gamma_{21}\} = \gamma_{11} = 0.60$. Les bi-clusters finaux sont

$$(P^{\mathcal{T}}, P^{\mathcal{G}})_1 = \{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}$$

$$(P^{\mathcal{T}}, P^{\mathcal{G}})_2 = \{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}$$

Chevauchement des classes

Nous présentons ici la possibilité de contrôler le chevauchement des bi-clusters et sa relation avec la classification floue (fuzzy clustering).

Nous pouvons permettre qu'un certain nombre d'objets et/ou propriétés puissent appartenir à plus d'une classe, en contrôlant la taille de la partie chevauchante de chaque classe. Grâce à notre définition d'appartenance à une classe, déterminée par les valeurs de $\boldsymbol{\tau}_i$ et $\boldsymbol{\gamma}_i$, nous avons juste besoin d'adapter l'étape d'affectation. Pour cela, nous introduisons les paramètres δ_t et δ_g à valeurs dans [0,1] pour quantifier l'appartenance de chaque élément à une classe. Nous disons qu'un objet t_j appartient à une classe $P_k^{\mathcal{T}}$ si $\tau_{kj} \geq (1-\delta_t) \cdot max_k(\tau_{kj})$. De la même manière, une propriété g_j appartient à une classe $P_k^{\mathcal{G}}$ si $\gamma_{kj} \geq (1-\delta_g) \cdot max_k(\gamma_{kj})$. Le nombre d'objets (resp. propriétés) chevauchants, dépend de la distribution des valeurs de $\boldsymbol{\tau}_k$ (resp. $\boldsymbol{\gamma}_k$).

Notons que si le chevauchement est permis, $\delta = 0$ n'implique pas que chaque objet et chaque propriété soient assignés à une seule classe (c'est-à-dire que la disjonction n'est pas garantie). Le choix d'une valeur pertinente pour δ est clairement dépendant de l'application. Quand une structure de bi-partitionnement tient implicitement dans les données, de petites valeurs de δ ne sont pas suffisantes pour fournir un chevauchement pertinent. D'un autre coté, dans les contextes bruités, même de petites valeurs de δ peuvent produire des zones chevauchantes consistantes. Il est intéressant de remarquer qu'en considérant les valeurs des centroïdes comme des coefficients d'appartenance, nous incorporons le cas de la classification floue [Bez81]. En effet, nous pouvons dériver facilement des bi-clusters flous μ_k^f à partir des vecteurs booléens μ_k, en divisant chaque composante par la somme des valeurs prises par la composante dans le K classes :

$$\mu_k^f = < \frac{\tau_{k1}}{\Theta_1}, \ldots, \frac{\tau_{kj}}{\Theta_j}, \ldots, \frac{\tau_{km}}{\Theta_m} >, < \frac{\gamma_{k1}}{\Gamma_1}, \ldots, \frac{\gamma_{kj}}{\Gamma_j}, \ldots, \frac{\gamma_{kn}}{\Gamma_n} >$$

avec

$$\Theta_j = \sum_{k=1}^{K} \tau_{kj}, \quad \Gamma_j = \sum_{k=1}^{K} \gamma_{kj}$$

Exemple. Dans notre exemple jouet de la Table 1.6, si nous permettons le chevauchement sur les objets avec $\delta_t = 0.4$, alors, l'objet t_7, est assigné aussi à la classe $P_1^{\mathcal{T}}$, car $\tau_{17} \geq (1-0.4) \cdot \max\{\tau_{17}, \tau_{27}\} = 0.2$ (cf. Table 3.2). Par conséquent, le premier bi-cluster dévient

$$(P^{\mathcal{T}}, P^{\mathcal{G}})_1 = \{t_1, t_3, t_4, t_7\}, \{g_1, g_3, g_4\}$$

.

Les deux vecteurs μ_1 et μ_2 peuvent être considérés comme une solution finale si on considère le cas d'un co-classification floue. En appliquant le simple post-traitement décrit précédemment, nous obtenons les deux bi-clusters flous de la Table 3.3.

μ_i^f	$< \tau_{i1}^f, \tau_{i2}^f, \tau_{i3}^f, \tau_{i4}^f, \tau_{i5}^f, \tau_{i6}^f, \tau_{i7}^f >$	,	$< \gamma_{i1}^f, \gamma_{i2}^f, \gamma_{i3}^f, \gamma_{i4}^f, \gamma_{i5}^f >$
μ_1^f	$< 1.00, 0.23, 1.00, 1.00, 0.29, 0.23, 0.37 >$	,	$< 0.64, 0.23, 1.00, 1.00, 0.17 >$
μ_2^f	$< 0.00, 0.77, 0.00, 0.00, 0.71, 0.77, 0.63 >$	,	$< 0.36, 0.77, 0.00, 0.00, 0.83 >$

TAB. 3.3 – Liste des vecteurs pour deux possibles bi-clusters

TAB. 3.4 – Pseudo-code de CDK-MEANS

CDK-MEANS ($\mathbf{r}$ est un contexte booléen, $\mathcal{B}$ est une collection de bi-ensembles dans $\mathbf{r}$, K est le nombre de classes, MI est le nombre d'itération maximum, δ_t et δ_g sont les valeurs des seuils pour contrôler le chevauchement)

1. Soit $\mu_1 \dots \mu_K$ les centroïdes initiaux des classes. $i := 0$.
2. Répéter
   (a) Pour chaque bi-ensemble $b \in \mathcal{B}$, l'assigner à la classe B_k telle que $d(b, \mu_i)$ est minimal.
   (b) Pour chaque classe B_k, calculer $\boldsymbol{\tau_k}$ et $\boldsymbol{\gamma_k}$.
   (c) $i := i + 1$.
3. Jusqu'à (centroïdes inchangées où $i = MI$).
4. Si le chevauchement est permis, $\forall t_j \in \mathcal{T}$ (resp. $g_j \in \mathcal{G}$), l'assigner à chaque classe $P_k^{\mathcal{T}}$ (resp. $P_k^{\mathcal{G}}$) telle que $\tau_{kj} \geq (1 - \delta_t) \cdot max_k(\tau_{kj})$ (resp. $\gamma_{kj} \geq (1 - \delta_g) \cdot max_k(\gamma_{kj})$).
5. Sinon, $\forall t_j \in \mathcal{T}$ (resp. $g_j \in \mathcal{G}$), l'assigner à la première classe $P_k^{\mathcal{T}}$ (resp. $P_k^{\mathcal{G}}$) telle que τ_{kj} (resp. γ_{kj}) est maximal.
6. Renvoyer $\{P_1^{\mathcal{T}} \dots P_K^{\mathcal{T}}\}$ and $\{P_1^{\mathcal{G}} \dots P_K^{\mathcal{G}}\}$

3.2.2 Algorithme CDK-MEANS

Nous introduisons maintenant l'instance CDK-MEANS esquissée dans la Table 3.4. Elle calcule une bi-partition d'un jeu de données $\mathbf{r}$ à partir d'une collection de bi-ensembles $\mathcal{B}$ précédemment extraite dans $\mathbf{r}$, du nombre de classes désiré K, des valeurs des seuils δ_t et δ_g, et d'un nombre maximal d'itérations MI. Dans notre exemple jouet, CDK-MEANS produit la bi-partition présentée dans la section précédente. L'algorithme étant de type *k-means*, les problèmes de passage à l'échelle par rapport au processus de classification sont bien maîtrisés. La complexité est linéaire avec le nombre de bi-ensembles dans la collection $\mathcal{B}$.

L'algorithme commence (1) en prenant en compte un ensemble initial de K centroïdes. Dans notre implémentation, cette initialisation est obtenue de façon aléatoire,

mais elle pourrait aussi être guidée par l'expert ou semi-supervisée. De plus, dans le cas d'une initialisation aléatoire, les centroïdes initiaux peuvent être construits en sélectionnant aléatoirement un ensemble de propriétés et un ensemble d'objets pour chaque centroïde, ou, en joignant un ensemble de bi-ensembles sélectionnés aléatoirement pour chaque centroïde. Une fois les centroïdes initiaux choisis, l'algorithme parcourt la collection de bi-ensembles et assigne chaque bi-ensemble à l'une des classes telles que la distance (calculée comme on l'a décrit dans la Section 3.2.1) est minimale. Ensuite, pour chaque classe, on génère un nouveau centroïde en joignant les bi-ensembles affectés à cette classe (cf. Section 3.2.1). Les lignes 2a et 2b sont exécutées tant qu'au moins un bi-cluster est modifié, ou l'algorithme s'arrête lorsque le nombre maximal d'itérations MI est atteint.

La seconde partie de l'algorithme est un simple post-traitement des centroïdes calculés pour assigner chaque objet et chaque propriété à un (ou plus) bi-cluster(s). Si on admet du chevauchement entre classes (Ligne 4), l'algorithme assigne chaque objet et propriété aux bi-clusters tels que la valeur d'appartenance dans le centroïde est supérieure à (1-δ) fois la valeur maximum. Si on désire des bi-clusters disjoints (Ligne 5), l'algorithme assigne chaque propriété et objet à l'un des bi-clusters tels que la valeur d'appartenance est maximum. Dans ce cas, lorsque plus d'une affectation est possible, différents critères de sélection peuvent être adoptés. Dans notre implémentation, nous effectuons un choix arbitraire en assignant chaque propriété et objet au premier bi-cluster avec la valeur d'appartenance maximale.

3.2.3 Complexité

Notre instance CDK-Means est une adaptation assez simple de l'algorithme *k-means* plus une simple phase de post-traitement. Pour la phase de post-traitement (Lignes 4-6), l'algorithme parcourt la liste des classes $m + n$ fois (m est le nombre d'objets, n est le nombre de propriétés), pour obtenir le coefficient d'appartenance maximum pour chaque objet et propriété. La complexité pour cette étape est donc $O(K \cdot (m + n))$. Lorsque $K << (m + n)$ l'algorithme s'exécute en temps linéaire.

La complexité standard de chaque itération d'un algorithme de type *k-means*, est $O(K \cdot m \cdot n)$, où K est le nombre de classes. Normalement, $n << m$, et $K << m$, donc on dit que l'algorithme *k-means* s'exécute en temps linéaire. Le point important ici est que CDK-Means ne manipule pas les objets, mais les bi-ensembles, et que les distances doivent être calculées sur des vecteurs booléens plutôt longs. Par conséquent, la complexité pour chaque itération est $O(K \cdot N \cdot (m + n))$, où N est le nombre de bi-ensembles (i.e., $|\mathcal{B}|$). Clairement, nous pouvons avoir beaucoup plus de bi-ensembles que d'objets ou propriétés dans les données booléennes. Comme la qualité de la bi-partition extraite est en partie liée à la pertinence des bi-ensembles extraits (i.e., plus ils capturent des associations localement pertinentes, plus la bi-partition pourra être pertinente), il reste toujours possible de réduire la taille de la

collection de bi-ensembles considérée, pourvu que les associations pertinentes soient préservées. Des exemples de réduction sensibles pourraient être d'éliminer les bi-ensembles trop petits, ou d'appliquer une phase d'interprétation pour éliminer les bi-ensembles non pertinents, selon des connaissances disponibles *a priori* (ou des tests statistiques). Réduire la collection des bi-ensembles peut rendre la phase de construction des centroïdes de CDK-Means plus rapide. Nous allons montrer plus loin dans le chapitre, que la qualité de la bi-partition extraite peut aussi être améliorée lorsque les bi-ensembles extraits paraissent plus pertinents en terme d'associations capturées. Pour conclure, nous avons ici un compromis entre la complexité de calcul, et la pertinence du résultat final, mais en sélectionnant les bi-ensembles en se basant sur leur pertinence *a priori* peut améliorer à la fois le temps de calcul et la qualité de la bi-partition calculée.

3.2.4 Problèmes liés à l'utilisation des bi-ensembles sous contraintes

Nous avons déjà discuté de la possibilité d'extraire une collection de bi-ensembles *a priori* intéressante préalablement au processus de construction de bi-clusters. Cela n'est pourtant pas sans conséquences pour le résultat final. L'utilisation des contraintes est, en effet un bon moyen soit pour réduire la taille de la collection traitée, soit pour en améliorer la pertinence. En revanche, dans certains cas, nombre d'objets ou propriétés n'apparaissant que dans des bi-ensembles qui ne satisfont pas les contraintes imposées, disparaîtront de la collection à traiter. Comme le processus entier de co-classification est basé sur les bi-ensembles, aucun des bi-clusters finaux ne contiendra ces objets ou ces propriétés.

Dans beaucoup d'applications, cela ne constitue pas un réel problème. Si, par exemple, on impose une contrainte de taille minimale, et qu'il y a un certain nombre d'objets et de propriétés qui disparaissent, on peut supposer qu'ils appartiennent à une partie de la matrice plutôt peu dense, et donc, dans certaines applications, peu intéressante pour les objectifs d'analyse.

Néanmoins, si cela est nécessaire, on peut envisager la création d'un nouveau bi-cluster "poubelle" contenant tous les objets et les propriétés qui ne sont pas classés. Le résultat donc, pour une application qui demande K classes, sera une bi-partition à $K + 1$ bi-clusters.

Une autre solution consiste à ajouter à la collection, qu'elles que soient les contraintes, les deux bi-ensembles "extrêmes", c'est-à-dire le bi-ensemble $b_{\mathcal{T}} = (\{\mathcal{T}\}, \{\emptyset\}$ et $b_{\mathcal{G}} = (\{\emptyset\}, \{\mathcal{G}\}$. Si on traite une collection de concepts formels, les deux bi-ensembles correspondent à l'infimum et supremum du treillis de Galois.

3.3 Exemples de requêtes

Les outils dont nous disposons jusqu'à maintenant, nous permettent d'écrire des requêtes très simples concernant les bi-partitions. Les seuls contrôles possibles sont sur le nombre de bi-clusters souhaités, et sur le niveau de chevauchement.

- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{I}})$ optimisant la distance entre bi-ensembles avec K bi-clusters.
- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{I}})$ optimisant la distance entre bi-ensembles avec K bi-clusters, avec un seuil de chevauchement pour les objets $\delta_t = 0.2$ et un seuil de chevauchement pour les propriétés $\delta_g = 0.3$.

Dans le prochain chapitre nous donnerons des exemples de requêtes beaucoup plus complexes et plus intéressantes d'un point de vue applicatif.

3.4 Validation de CDK-Means

3.4.1 Méthodes d'évaluation de la qualité d'une co-classification

L'évaluation d'un algorithme de classification non supervisée est toujours une tâche difficile. En général, tout ce qui est non supervisé comporte une valeur ajoutée qui peut être validée uniquement par des experts du domaine. Ils existent plusieurs critères pour mesurer la qualité d'une partition ou d'une bi-partition.

Un critère général pour évaluer les résultats d'une classification consiste à comparer la partition calculée avec une partition "correcte". Cela signifie que les instances des données sont déjà associées à des étiquettes jugées correctes, et que l'on va pouvoir quantifier la conformité entre étiquettes calculées et étiquettes correctes. Des mesures classiques sont l'indice de Rand et de Jaccard [JD88] pour évaluer la conformité entre deux partitions de m éléments.

Si $\mathbf{C} = \{C_1 \dots C_s\}$ est la structure issue de la classification et que $\mathbf{P} = \{P_1 \dots P_t\}$ est une partition prédéfinie, chaque paire de points peut être affectée au même cluster ou à deux clusters différents. Soit a le nombre de paires appartenant au même cluster de $\mathbf{C}$ et au même cluster de $\mathbf{P}$. Soit b le nombre de paires dont les points appartiennent à deux clusters différents de $\mathbf{C}$ et à deux clusters différents de $\mathbf{P}$. La conformité entre $\mathbf{C}$ et $\mathbf{P}$ peut être estimée au moyen de la formule :

$$Rand(\mathbf{C}, \mathbf{P}) = \frac{a + b}{m \cdot (m-1)/2}$$

ou

$$Jaccard(\mathbf{C}, \mathbf{P}) = \frac{a}{m \cdot (m-1)/2 - b}$$

Ces indices prennent des valeurs entre 0 et 1 et ils sont maximisés quand $s = t$.

Nous utilisons l'indice de Rand et celui de Jaccard pour calculer la précision dans nos expériences.

Nous voulons aussi évaluer la qualité intrinsèque de la co-classification au moyen d'un critère interne. Pour ce faire, une mesure intéressante est le coefficient τ symétrique de Goodman and Kruskal's [GK54]. Il est évalué dans une table de contingence $\mathbf{p}$ et il discrimine correctement les bi-partitions par rapport à l'intensité du lien fonctionnel entre leurs deux partitions [RF01b]. Soit p_{ij} la fréquence des relations entre un objet d'un cluster $P_i^{\mathcal{T}}$ et une propriété d'un cluster $P_j^{\mathcal{G}}$, et $p_{i.} = \sum_j p_{ij}$ et $p_{.j} = \sum_i p_{ij}$. Nous utilisons le coefficient $\tau_{\mathcal{S}}$ qui évalue la réduction proportionnelle de l'erreur entraînée par la connaissance de $P^{\mathcal{T}}$ sur la prédiction de $P^{\mathcal{G}}$ et vice versa. Il est défini de la manière suivante :

$$\tau_{\mathcal{S}} = \frac{\frac{1}{2}\sum_i \sum_j \left(p_{ij} - p_{i.}p_{.j}\right)^2 \frac{p_{i.}+p_{.j}}{p_{i.}p_{.j}}}{1 - \frac{1}{2}\sum_i p_{i.}^2 - \frac{1}{2}\sum_j p_{.j}^2}$$

Dans certains cas il peut être utile de calculer la valeur de la fonction objectif optimisée par [DMM03]. Pour cela, on calcule l'information mutuelle, qui calcule la quantité d'information que $P^{\mathcal{T}}$ contient sur $P^{\mathcal{G}}$:

$$I(P^{\mathcal{T}}; P^{\mathcal{G}}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_{i.}p_{.j}}$$

Alors, étant données deux bi-partitions différentes $(P^{\mathcal{T}}, P^{\mathcal{G}})$ et $(\hat{P}^{\mathcal{T}}, \hat{P}^{\mathcal{G}})$, la perte d'information mutuelle est donnée par :

$$I(P^{\mathcal{T}}; P^{\mathcal{G}}) - I(\hat{P}^{\mathcal{T}}; \hat{P}^{\mathcal{G}})$$

Enfin, pour évaluer les performances de notre méthode, nous définissons le coefficient de comparaisons par la moyenne des produits du nombre d'itérations nécessaires pour compléter la classification, et le nombre des bi-ensembles, i.e. :

$$CC = \frac{\sum_i^N |\mathcal{B}| \cdot NI_i}{N}$$

où, N indique le nombre d'exécutions, $|\mathcal{B}|$ est la taille de la collection de bi-ensembles, et NI_i est le nombre d'itérations à la i^{eme} exécution.

3.4.2 Application à des données benchmark

Nous avons effectué nos validations expérimentales sur huit jeux de données "benchmark". Parmi eux, sept (voting-records, iris, zoo, breast-w, credit-a, internet-ads, mushroom) font partie de l'UCI ML Repository[2]. Le dernier (titanic) provient de

[2]http ://www.ics.uci.edu/~mlearn/MLRepository.html

l'archive JSE de l'American Statistical Association[3]. Toutes nos expériences ont été effectuées sur un ordinateur avec 1 Go de mémoire vive et un processeur Pentium 4 à 3.0 GHz, processor fonctionnant avec le système d'exploitation Linux.

Nous avons effectué chaque expérience sur trois collections de bi-ensembles. Nous avons d'abord utilisé une collection de concepts formels, ensuite nous avons utilisé deux collections de δ-bi-ensembles avec $\delta = 1$ et $\delta = 2$. De plus, nous avons effectué plusieurs expériences sur le jeu de données internet-ads pour illustrer l'impact d'une étape de sélection sur une collection de bi-ensembles. Nous avons également étudié le comportement dynamique de CDK-Means en mesurant le nombre minimal, maximal et moyen d'itérations pour chaque jeu de données. Les résultats concernant la qualité par rapport au coefficient de Goodman-Kruskal sont dans la Table 3.5 (pour les concepts formels) et dans la Table 3.12 et Table 3.13 (pour les δ-bi-enembles). Les coefficient de Jaccard correspondant à la variable de classe, sont dans la Table 3.6 (pour les concepts formels), et dans la Table 3.12 et Table 3.13 (pour les δ-bi-ensembles). Les résultats en terme de nombre d'itérations pour des collections différentes de concepts formels sont dans la Table 3.8. Pour les δ-bi-ensembles, les résultats se trouvent dans la Table 3.10 (pour $\delta = 1$) et dans la Table 3.11 (pour $\delta = 2$). Enfin, l'analyse du passage à l'échelle est résumée dans la Table 3.9.

Résultats avec les concepts formels

Sans considérer la variable de classe, nous avons d'abord traité chaque jeu de données avec D-Miner [BRBR05] sans utiliser des contraintes additionnelles (sauf pour voting-records, mushroom et credit-a qui nécessitent clairement de contraintes de taille pour pouvoir obtenir des collections plus petites mais encore pertinentes). L'algorithme a fourni des collections complètes (par rapport aux contraintes, si spécifiées) de concepts formels pour exécuter CDK-Means.

Nous avons comparé les bi-partitions obtenues avec CDK-Means avec celles obtenues avec Cocluster [DMM03], et Bi-Clust [RF01b]. L'initialisation de deux premiers algorithmes étant aléatoire, nous les avons exécutés 100 fois pour chaque jeu de données, et nous avons mesuré la valeur moyenne et la valeur maximum des coefficients de Goodman-Kruskal. Le nombre de bi-clusters désirés pour chaque expérience est égal au nombre de variables de classes, exception faite pour Bi-Clust qui détermine automatiquement le nombre de classes. Bi-Clust est disponible dans la plateforme WEKA [WF99] et nous n'avons pas pu traiter le jeu de données internet-ads (plus que 1500 propriétés). Nous résumons ces résultats dans la Table 3.5. On peut voir que, quand CDK-Means obtient les résultats les plus bas, le coefficient de Goodman-Kruskal n'est pas significativement différent des coefficients des autres algorithmes. D'un autre côté, pour internet-ads, le coefficient obtenu avec CDK-Means

[3]http ://www.amstat.org/publications/jse/jse_data_archive.html

	Bi-Clust	Cocluster		CDK-means	
Dataset	Max	Max	Mean	Max	Mean
voting	**0.320**	0.314	0.308±0.008	0.311	0.311±0.000
titanic	**0.332**	0.321	0.226±0.076	0.363	0.215±0.118
iris-2	**0.544**	**0.544**	0.357±0.195	**0.544**	0.422±0.159
iris-3	**0.544**	0.471	0.285±0.141	**0.544**	0.329±0.085
zoo-2	0.191	0.186	0.157±0.034	**0.198**	0.165±0.024
zoo-7	-	0.106	0.102±0.006	**0.110**	0.063±0.014
breast-w	0.507	**0.507**	0.413±0.196	0.498	0.498±0.000
credit-3	**0.104**	0.019	0.007±0.005	0.079	0.066±0.008
credit-2	-	0.013	0.003±0.003	**0.096**	0.086±0.023
mr-2	-	**0.198**	0.158±0.026	0.176	0.157±0.017
mr-5	**0.187**	0.142	0.111±0.010	0.118	0.096±0.011
ads	-	0.006	0.003±0.001	**0.661**	0.158±0.113

Tab. 3.5 – Valeurs du coefficient de Goodman-Kruskal pour des différents algorithmes de co-clustering (mr-2 et mr-5 concernent mushroom avec 2 et 5 classes).

est considérablement plus élevé que celui obtenu avec Cocluster. Cela est dû à la dimensionalité élevée du jeu de données qui n'est pas bien maîtrisée par les autres algorithmes. De plus, le comportement moyen est similaire à celui de Cocluster. Les valeurs moyennes, aussi que les valeurs des écarts-type de deux algorithmes, sont souvent similaires. Il faut remarquer que pour voting-records et breast-w, CDK-Means a toujours fourni la même bi-partition.

En général, CDK-Means a besoin d'un temps d'exécution plus long que les autres algorithmes, car il traite des collections de bi-ensembles parfois grandes. Dans ces benchmarks, l'extraction des concepts formels n'est pas coûteuse en tant que telle (de 1 à 20 secondes). L'utilisation d'une contrainte de taille lors de la phase d'extraction de concepts formels, permet de réduire la taille de la collection et sera l'objet d'une étude plus loin dans ce chapitre. Pour titanic, iris, et zoo, CDK-Means s'exécute en moins d'une seconde, tandis que pour breast-w, credit-a et internet-ads, le temps d'exécution moyen est moins d'une minute. Pour mushroom, le temps d'exécution moyen est autour de sept minutes car il faut traiter plus de 50 000 concepts formels, et la taille maximum des centroïdes et relativement élevée (plus de 8 200 éléments).

Dans un deuxième groupe d'expériences, nous avons utilisé l'indice de Jaccard pour comparer la conformité de la partition des objets avec celle déterminée par la variable de classe. Nous résumons les comparaisons dans la Table 3.6. Encore une fois, notre algorithme est compétitif vis-à-vis des autres méthodes de co-classification. Exception faite pour breast-w, notre algorithme obtient toujours des meilleurs résultats par rapport à Bi-Clust et Cocluster, et le comportement moyen est similaire à celui de Cocluster.

Enfin, nous avons comparé nos résultats avec ceux obtenus en appliquant deux

	Bi-Clust	Cocluster		CDK-means	
Dataset	Max	Max	Mean	Max	Mean
voting	0.647	0.653	0.624±0.019	**0.674**	0.674±0.000
titanic	0.428	0.560	0.441±0.074	**0.598**	0.434±0.070
iris-2	0.499	0.499	0.434±0.069	**0.505**	0.471±0.055
iris-3	0.493	0.524	0.432±0.094	**0.556**	0.476±0.045
zoo-2	**0.514**	0.460	0.365±0.062	0.463	0.423±0.037
zoo-7	-	0.61	0.509±0.096	**0.772**	0.372±0.092
breast-w	0.825	**0.829**	0.765±0.137	0.767	0.767±0.000
credit-3	0.423	**0.518**	0.383±0.048	0.506	0.387±0.040
credit-2	-	**0.525**	0.437±0.058	0.495	0.479±0.033
mr-2	-	0.694	0.494±0.132	**0.699**	0.480±0.120
mr-5	**0.507**	0.482	0.338±0.039	0.501	0.350±0.062
ads	-	0.432	0.432±0.000	**0.885**	0.665±0.131

Tab. 3.6 – Valeurs des coefficients de Jaccard par rapport aux variables de classe pour différents algorithmes

	K-Means		EM		CDK-means	
Dataset	Max	Mean	Max	Mean	Max	Mean
voting	0.651	0.627±0.034	0.637	0.637±0.000	**0.674**	0.674±0.000
titanic	0.554	0.466±0.075	0.443	0.425±0.002	**0.598**	0.434±0.070
iris-2	**0.512**	0.497±0.018	0.499	0.499±0.000	0.505	0.471±0.055
iris-3	0.539	0.510±0.043	0.526	0.515±0.014	**0.556**	0.476±0.045
zoo-2	0.462	0.460±0.013	0.461	0.461±0.000	**0.463**	0.423±0.037
zoo-7	**0.922**	0.588±0.144	0.856	0.761±0.047	0.772	0.372±0.092
breast-w	0.825	0.788±0.063	**0.833**	0.833±0.000	0.767	0.767±0.000
credit-3	0.444	0.382±0.029	0.441	0.371±0.035	**0.506**	0.387±0.040
credit-2	0.519	0.469±0.036	0.444	0.443±0.008	0.495	0.479±0.033
mr-2	0.687	0.537±0.134	0.694	0.550±0.141	**0.699**	0.480±0.120
mr-5	0.497	0.356±0.040	0.459	0.352±0.038	**0.501**	0.350±0.062
ads	-	-	-	-	**0.885**	0.665±0.131

Tab. 3.7 – Valeurs des coefficients de Jaccard par rapport aux variables de classe pour différents algorithmes

Dataset	N.Bi-sets	Min	Mean	Max	Failure
voting	199866	3	17.58	24	0
titanic	38	2	3.36	4	0
iris-2	50	3	4.55	6	0
iris-3	50	4	4.26	5	0
zoo-2	309	4	8.97	14	0
zoo-7	309	10	12.69	18	0
breast-w	4903	8	12.35	17	0
credit-3	29447	8	30.67	67	0
credit-2	29447	4	15.45	26	0
mr-2	53942	4	8.46	24	0
mr-5	53942	10	20.46	62	0
ads	7682	6	16.73	54	0

TAB. 3.8 – Nombre d'itérations pour différents jeux de données (mr-2 et mr-5 se réfère à mushroom avec 2 et 5 classes).

algorithmes classiques de classification, les implémentations WEKA de K-MEANS et EM (cf. Table 3.7). Á l'exception de breast-w et zoo-7, notre algorithme est compétitif par rapport aux autres. Pour la plupart des jeux de données, CDK-MEANS obtient des résultats meilleurs que ceux du K-MEANS standard et EM. Le comportement en moyenne est plus mauvais en générale, mais les moyennes et les écarts-types ne sont pas très différents de ceux obtenus par les deux autres algorithmes. Ces résultats montrent que notre méthode est une approche pertinente autant pour la tâche de classification que pour celle de co-classification.

Analyse de convergence

Du moment où la complexité de calcul de chaque itération de CDK-MEANS dépend aussi du nombre de bi-ensembles, il est intéressant d'analyser le nombre d'itérations dont CDK-MEANS a besoin pour obtenir des centroïdes stables.

Dans nos précédentes expériences, nous avons fixé à 100 le nombre maximal d'itérations. Est-il possible de décider quelle valeur est pertinente pour ce paramètre sur un certain jeu de données particulier ? Comme on utilise une étape d'initialisation aléatoire, nous avons étudié le comportement en moyenne et le nombre minimal/maximal d'itérations pour 100 exécutions sur plusieurs jeux de données. Les résultats sont donnés dans la Table 3.8.

CDK-MEANS arrive à réaliser le calcul de centroïdes stables en utilisant un nombre très petit d'itérations pour chaque jeu de données employé. Le nombre moyen d'itérations est presque toujours inférieur à 20 (sauf pour mr-5 et credit-a avec $K = 3$ où il est légèrement plus grand). Le nombre maximum d'itérations dans

(σ_p,σ_o)	$\lvert\mathcal{B}\rvert$	time(s)	τ(mean)	τ(max)	J-class	J-ref
(0,0)	7682	33	0.137 ± 0.109	0.538	0.8019	1
(4,4)	2926	8	0.194 ± 0.137	0.565	0.6763	0.6737
(5,5)	2075	5	0.254 ± 0.148	0.565	0.6862	0.7490
(5,10)	1166	2.5	0.223 ± 0.119	0.511	0.6745	0.7405
(7,10)	873	2	0.204 ± 0.095	0.549	0.6172	0.6658
(10,10)	586	1.5	0.227 ± 0.125	0.543	0.6080	0.7167

TAB. 3.9 – Résultats de la classification sur ads-internet avec différentes contraintes de taille minimale

8 jeux de données est inférieur à 25. De plus, on peut noter qu'aucune exécution n'a été arrêtée avant d'obtenir des centroïdes stables. Ces résultats nous suggèrent certaines améliorations sont possibles, avec, par exemple, un choix plus soigné des centroïdes initiaux pour améliorer la convergence de CDK-MEANS et ainsi réduire les temps d'exécution.

Passage à l'échelle

Le nombre de concepts formels qui tiennent même dans des petits jeux de données, peut être énorme, spécialement lorsque les données sont intrinsèquement bruitées. Comme CDK-MEANS a une complexité linéaire avec le nombre de bi-ensembles, il se peut que son exécution soit coûteuse. Une solution triviale, consiste à sélectionner un sous-ensemble de la collection des concepts formels, en sélectionnant par exemple ceux qui impliquent un nombre suffisant d'objets et/ou de propriétés. Il est intéressant de voir que cette contrainte de taille minimale peut être poussée par les algorithmes d'extraction de concepts formels tels que D-MINER [BRBR05]. Cela permet non seulement de faciliter l'extraction dans des contextes difficiles, mais aussi, intuitivement, d'éliminer les concepts formels qui pourraient être dûs au bruit. Nous nous demandons donc, si cela peut accroître la qualité des résultats de la classification.

Nous avons effectué d'autres expériences pour comprendre l'impact de l'utilisation des contraintes de taille minimale à la fois sur les temps d'exécution et sur la qualité de la bi-partition calculée. Nous avons considéré internet-ads pour ces expériences (ce jeu de données a une haute cardinalité pour les deux ensembles d'objets et propriétés). Soit σ_o la taille minimale de l'ensemble des objets et σ_p la taille minimale de l'ensemble des propriétés. Nous avons extrait les concepts formels en établissant des combinaisons de contraintes ($0 \leq \sigma_p < 10$ et $0 \leq \sigma_o < 10$) et en ajoutant les deux concepts "top" et "bottom". Les résultats sont résumés dans la Table 3.9. Ils montrent qu'augmenter le seuil de taille minimale permet de réduire considérablement le nombre de concepts formels extraits, et par conséquent, le temps d'exécution. De même, les temps d'extraction diminuent de 4 secondes (pour $\sigma_p = \sigma_o = 0$) à moins d'une seconde (pour $\sigma_p = \sigma_o = 10$). De plus, le coefficient de Goodman-Kruskal

maximum ne change pas de façon significative. Dans certains cas, il est plus grand que celui calculé sans l'utilisation de contraintes de taille. Les valeurs moyennes des mesures de Goodman-Kruskal, elles aussi, sont meilleures en général (alors que les valeurs des écarts-type sont similaires). Lorsqu'on calcule l'indice de Jaccard pour les différentes partitions par rapport à la variable de classe (colonne J-class) et la partition obtenue sans établir aucune contrainte (colonne J-ref), la faible variabilité des indices de Jaccard comparée aux valeurs élevées des mesures τ, montre que même s'il y a des différences entre partitions, elles sont toujours consistantes vis-à-vis de la variable de classe. Enfin, les résultats sont toujours meilleurs que ceux obtenus en utilisant COCLUSTER (cf. Figure 3.5 et Figure 3.6) dont le temps d'exécution moyen est d'environ 4,2 secondes. En d'autres termes, accroître σ_p et σ_o peut éliminer l'impact du bruit dû aux sous-régions peu denses de la matrice. En particulier, regrouper des concepts formels plus grands, peut améliorer la pertinence de la bi-partition calculée. Il faut noter que, si on n'ajoute pas $(\mathcal{T}, \emptyset)$ and $(\emptyset, \mathcal{G})$, nous obtenons de meilleurs résultats impliquant un sous-ensemble de la matrice originale : les contraintes peuvent être réglées pour chercher un compromis entre la couverture de la bi-partition et la qualité des résultats.

Expériences avec les δ-bi-ensembles

Dans la section précédente, nous avons présenté le comportement de CDK-MEANS lorsqu'on considère des collections de concepts formels. Maintenant, nous allons mettre l'accent sur la généricité du cadre en considérant des expériences sur des collections de δ-bi-ensembles. Dans la plupart des cas, on s'attend à ce que l'on obtienne des collections plus petites de bi-ensembles éventuellement plus larges, qui peuvent toutefois mieux capturer des association fortes dans les données (grâce à la tolérance au bruit). Cela devrait au moins accélérer la première phase de l'algorithme CDK-MEANS. Les expériences ont été effectuées comme on l'a décrit dans la section précédente. La seule différence concerne le type de bi-ensemble que nous avons utilisé. Nous avons calculé les δ-bi-ensembles (comme expliqué dans la Section 2.3.1) avec $\delta = 1$ (au plus une exception par colonne) et $\delta = 2$ (au plus deux exceptions par colonne). Pour voting-records, mushroom et credit-a, nous avons utilisé les mêmes contraintes de taille que pour les expériences précédentes basées sur les concepts formels.

Une première analyse concerne le gain en performances. Nous avons trois paramètres objectifs pour évaluer ce gain (ou perte) : le nombre de bi-ensembles traités, le nombre moyen d'itérations nécessaires pour fournir une bi-partition, et le gain réel en termes de coefficient de comparaison. Le gain est calculé comme la différence entre le coefficient de comparaisons obtenu en utilisant les concepts formels, et celui obtenu en utilisant les δ-bi-ensembles. Les résultats sont dans la Table 3.10 et Table 3.10 (des valeurs négatives du gain, indiquent une perte en performances). Comme on s'attendait, pour $\delta = 1$, la taille des collections de δ-bi-ensembles est plus petite que celles

Dataset	N.Bi-sets	Min	Mean	Max	Gain
voting	97330	17	18.70	21	48.20%
titanic	36	2	3.52	7	0.83%
iris-2	42	4	4.68	7	13.51%
iris-3	42	4	4.80	7	5.35%
zoo-2	258	5	7.42	11	30.93%
zoo-7	258	10	12.82	30	15.61%
breast-w	3776	7	9.06	13	43.52%
credit-3	26740	8	31.66	68	6.24%
credit-2	26740	4	16.90	28	0.66%
mr-2	26304	3	7.98	22	54.03%
mr-5	26304	7	17.63	64	57.98%
ads	6832	6	13.98	38	25.70%

TAB. 3.10 – Nombre d'itérations pour différents jeux de données en utilisant les 1-bi-ensembles (mr-2 et mr-5 se réfère à mushroom avec 2 et 5 classes).

de concepts formels.

Nous analysons maintenant le comportement dynamique de CDK-MEANS. Dans ces expériences, le nombre moyen d'itérations est presque le même que pour le traitement des concepts formels. Toutefois, le nombre de comparaisons est réduit de plus de 25% dans 6 jeux de données. Quand on utilise les 2-bi-ensembles, cette amélioration des performances est claire, (cf. Table 3.11), sauf pour mushroom. Dans ce cas, le nombre de 2-bi-ensembles est plus important que le nombre de 1-bi-ensembles, et aussi du nombre de concepts formels (cf. Table 3.8). Cela peut apparaître un peu bizarre, mais nous rappelons que nous avons utilisé une contrainte de taille minimale dans ce jeu de données particulier. Par conséquent, lorsque l'on extrait les δ-bi-ensembles avec $\delta = \delta_1$ sous une contrainte de taille minimale, il se peut que la collection résultante soit plus grosse que celle obtenue avec un $\delta = \delta_2$ où $\delta_2 < \delta_1$. En outre, les concepts formels sont des δ-bi-ensembles avec $\delta = 0$. La raison est que, même si la taille de la collection à part entière de δ_2-bi-ensembles (sans contrainte de taille minimale) est plus petite que la collection de δ_1-bi-ensembles, il peut y avoir plus de δ_1-bi-ensembles satisfaisant la contrainte que de δ_2-bi-ensemble, car, en tolérant des exceptions, ces motifs capturent des associations plus larges. Cela explique la perte en performances lorsqu'on utilise $\delta = 2$ avec mushroom (le nombre moyen de comparaisons est égal à deux fois le nombre moyen de comparaisons nécessaire lorsque $\delta = 1$).

Nous considérons maintenant l'analyse de la qualité. Nous avons analysé le comportement moyen et le maximum du coefficient τ de Goodman-Kruskal, et de l'indice de Jaccard (par rapport à la variable de classe). Les résultats sont dans la Table 3.12 (pour $\delta = 1$) et Table 3.13 (pour $\delta = 2$) (les valeurs en gras indiquent des meilleurs résultats par rapport à la Table 3.5).

Dans certains cas, nous obtenons un meilleur coefficient de Goodman-Kruskal,

Dataset	N.Bi-sets	Min	Mean	Max	Gain(FC)	Gain($\delta = 1$)
voting	55359	10	16.73	26	73.65%	49.12%
titanic	36	2	3.27	5	7.80%	7.03%
iris-2	34	3	4.76	6	28.77%	17.64%
iris-3	34	2	3.31	5	47.12%	44.13%
zoo-2	190	6	7.68	11	47.33%	23.74%
zoo-7	190	5	8.16	12	60.45%	53.14%
breast-w	2831	6	7.87	11	63.22%	34.87%
credit-3	23946	14	34.39	69	8.82%	2.75%
credit-2	23946	4	17.78	28	6.41%	5.79%
mr-2	54685	3	8.24	21	1.31%	-114.67%
mr-5	54685	9	19.55	53	3.13%	-130.53%
ads	6304	5	12.22	29	40.07%	19.35%

TAB. 3.11 – Nombre d'itérations pour différents jeux de données en utilisant les 2-bi-ensembles (mr-2 et mr-5 se réfère à mushroom avec 2 et 5 classes).

	Goodman-Kruskal		Jaccard	
Dataset	Max	Mean	Max	Mean
voting	**0.314**	0.313 ± 0.000	0.667	0.664±0.002
titanic	0.363	0.217 ± 0.115	0.598	0.428±0.071
iris-2	**0.545**	0.442 ± 0.134	**0.507**	0.483±0.046
iris-3	0.544	0.298 ± 0.092	0.556	0.473±0.059
zoo-2	0.193	0.170 ± 0.011	**0.464**	0.441±0.016
zoo-7	**0.120**	0.069 ± 0.018	**0.799**	0.376±0.095
breast-w	0.497	0.497 ± 0.000	0.763	0.763±0.000
credit-3	0.078	0.066 ± 0.008	0.447	0.397±0.043
credit-2	0.095	0.086 ± 0.019	**0.497**	0.485±0.030
mr-2	0.174	0.151 ± 0.015	0.684	0.538±0.102
mr-5	0.116	0.101 ± 0.017	**0.677**	0.635±0.019
ads	**0.577**	0.237 ± 0.154	**0.851**	0.665±0.138

TAB. 3.12 – Coefficients de Jaccard et Goodman-Kruskal pour $\delta = 1$ (mr-2 et mr-5 se réfère à mushroom avrc 2 et 5 classes).

	Goodman-Kruskal		Jaccard	
Dataset	Max	Mean	Max	Mean
voting	0.312	0.312 ± 0.000	0.647	0.647±0.000
titanic	0.363	0.171 ± 0.083	**0.602**	0.498±0.071
iris-2	**0.546**	0.442 ± 0.141	**0.509**	0.469±0.057
iris-3	**0.546**	0.343 ± 0.096	**0.563**	0.471±0.055
zoo-2	0.191	0.162 ± 0.020	**0.471**	0.442±0.029
zoo-7	**0.111**	0.068 ± 0.015	0.737	0.411±0.098
breast-w	0.420	0.420 ± 0.000	0.735	0.735±0.000
credit-3	0.077	0.067 ± 0.009	0.506	0.404±0.045
credit-2	0.094	0.082 ± 0.015	**0.661**	0.629±0.039
mr-2	0.173	0.158 ± 0.013	0.687	0.570±0.102
mr-5	0.116	0.106 ± 0.005	0.501	0.337±0.031
ads	0.567	0.311 ± 0.137	0.830	0.669±0.142

TAB. 3.13 – Coefficients de Jaccard et Goodman-Kruskal pour $\delta = 2$ (mr-2 et mr-5 se réfère à mushroom avrc 2 et 5 classes).

alors qu'il semble que l'utilisation des δ-bi-ensembles n'a pas d'impact sur le comportement en moyenne. L'indice de Jaccard semble avantager les δ-bi-ensembles contre les concepts formels. Les expériences montrent que les δ-bi-ensembles constituent une alternative valide aux concepts formels pour le calcul des bi-partitions. Le gain en performances n'implique pas une perte en qualité, mais utiliser des valeurs plus grandes de δ n'est pas toujours le meilleur choix. Ces expériences ont montré que CDK-MEANS peut fonctionner avec deux types différents de bi-ensembles qui capturent des associations fortes localement. Il est clair que d'autres types de bi-ensembles pourraient être utilisés, par exemple d'autres types de concepts formels tolérant au bruit, ou de rectangles denses [BRB04b, GMS04, BRB05].

3.4.3 Application à des données d'expression

Pour montrer la valeur ajoutée de notre approche dans un jeu de données réelles, nous avons appliqué CDK-MEANS au jeu de données puces à ADN (malaria [BLP+03]) concernant le transcriptome du cycle de développement intraerythrocytique du Plasmodium Falciparum (cf. chapitre précédent).

La première expérience consiste à identifier les trois étapes du développement. Pour cela nous avons extrait tous les concepts (sans aucune contrainte) avec D-MINER et nous avons obtenu une collection de 59 011 concepts formels. En appliquant CDK-MEANS avec $K = 3$, le score maximum que nous avons obtenu est $\tau_{\mathcal{S}} = 0.5129$. En appliquant COCLUSTER, le coefficient de Goodman-Kruskal maximal est sensiblement plus grand, ($\tau_{\mathcal{S}} = 0.6123$). Cependant, si on regarde la partition des échantillons biologiques résultante, elle est significativement différente par rapport à celle obtenue

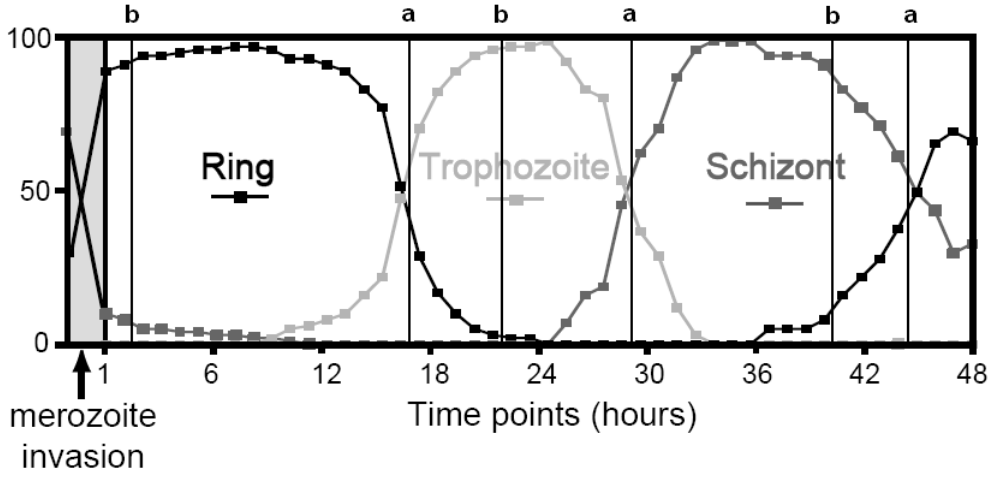


FIG. 3.2 – Frontières des classes pour CDK-MEANS (a) et COCLUSTER (b). Les courbes montrent la représentation en pourcentage des parasites dans la phase anneau, trophozoïte, où schizonte dans la culture à chaque instant de temps [BLP+03]

avec CDK-MEANS. Avec CDK-MEANS, la conformité entre la partition obtenue et les trois étapes décrites dans [BLP+03], est très élevée (cf. Figure 3.2 lignes a). Les trois groupes sont bien distingués, et les transitions entre les classes, correspondent à celles reportées dans [BLP+03]. Avec COCLUSTER, les lignes de frontière des classes sont décalées (cf. Figure 3.2 lignes b).

Ensuite, nous avons analysé la partition des gènes. Nous avons contrôlé l'assignation des classes pour 12 groupes fonctionnels de gènes. Chaque groupe contient un certain nombre de gènes avec la même fonction, et chaque fonction concerne un ou au plus deux étapes du développement. Encore une fois, ces groupes sont bien décrits dans [BLP+03] et contiennent de 6 a 135 gènes. Pour chaque groupe de gènes, la Table 3.14 représente le nombre de gènes qui ont été assignés à chaque classe, divisé par le nombre total de gènes appartenant au groupe fonctionnel (lorsque des gènes sont manquantes dans les données, la somme des rapports et inférieure à 1). Ici encore, la répartition des gènes est pertinente par rapport à la connaissance biologique disponible [BLP+03]. Par exemple, les 4 premiers groupes de gènes, sont connus pour jouer un rôle dans la phase anneau et dans le début de la phase trophozoïte. De la même manière, les groupes "proteasome", "plastid", "merozoite invasion" et "actin myosin motility" représentent des fonctions qui sont caractéristiques de la phase shizonte. Cela est évident pour les trois derniers groupes (qui incluent environ cent gènes), où les rapports valent 1, alors que dans les résultats de COCLUSTER, ces gènes sont partagés dans deux classes liées à la phase trophozoïte et schizonte.

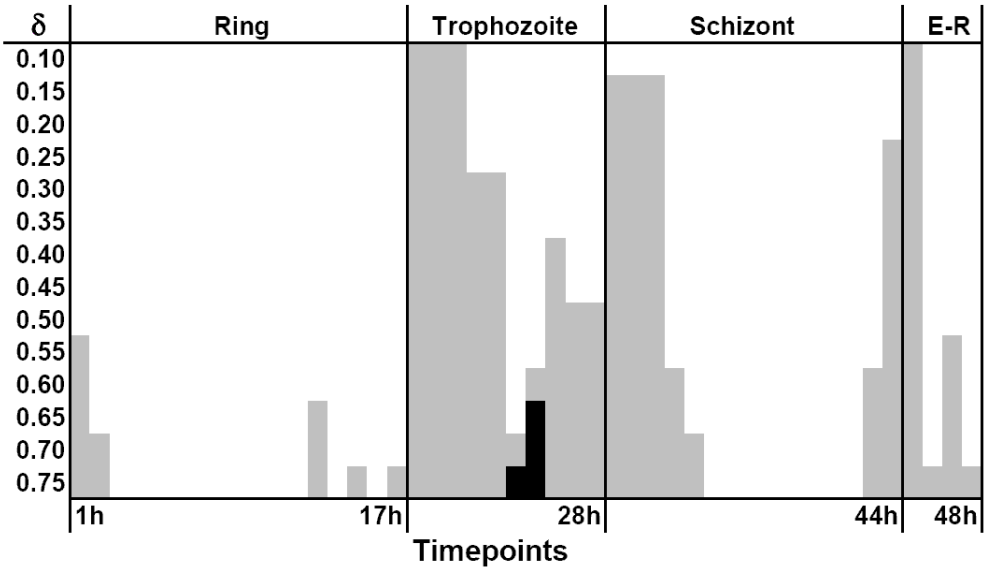


FIG. 3.3 – Zone de chevauchement des classes lorsque δ_p varie

Enfin, nous avons exécuté une classification approchée (en permettant le chevauchement) et nous avons analysé la pertinence des intersections entre classes. Nous avons utilisé les résultats précédents, et pour chaque valeur de δ_p (de 0.10 à 0.75, avec un pas égal à 0.05), nous avons construit les partitions approchées relatives. Les résultats sont dans la Figure 3.3, où les zones en gris clair sont les intersections entre deux classes, et la zone noire, est une intersection entre toutes les trois classes. Cette expérience valide notre approche lorsque $\delta_p \leq 0.5$: les intersections contiennent des échantillons de temps qui sont concernés par la transition entre deux étapes adjacentes.

3.5 Conclusion

Nous avons introduit un nouveau cadre de co-classification qui exploite les motifs locaux dans les données lors du calcul d'une collection de bi-clusters (éventuellement chevauchants). L'instance CDK-MEANS construit simultanément une partition sur les objets et une partition sur les propriétés en appliquant un algorithme de type *k-means* à une collection de concepts formels extraits. Contrairement aux autre approches de co-classification qui traitent les lignes et les colonnes séparément (même si la fonction objectif optimisée est commune), notre approche traite les objets et les propriétés de manière vraiment simultanée. De plus, la mesure de similarité utilisée prend en compte le nombre de bi-ensembles dans lequel chaque objet et chaque pro-

Function	Ring	Trophozoite	Schizont
transcription	**0.91**	0.04	0.00
cytoplasmic translation	**0.98**	0.02	0.00
glycolytic pathway	**0.43**	**0.43**	0.00
ribonucleotide synthesis	**0.67**	0.22	0.00
deoxynucleotide synthesis	0.00	0.29	**0.71**
dna replication	0.00	0.33	**0.67**
tca cycle	0.00	0.36	**0.64**
proteasome	0.00	0.09	**0.51**
plastid genome	0.00	0.00	**1.00**
merozoite invasion	0.00	0.00	**1.00**
actin myosin motility	0.00	0.00	**1.00**
early ring transcripts	**0.79**	0.00	0.21

TAB. 3.14 – Rapport d'assignation des groupes fonctionnels dans les trois bi-clusters découverts

priété sont impliqués. L'approche consiste donc à classer les associations plutôt que les objets et les propriétés directement. Nous avons montré dans ce chapitre la valeur ajoutée d'une telle approche.

Il apparaît clairement que ce cadre fournit des possibilités supplémentaires en ce qui concerne l'utilisation des contraintes. En effet, nous avons ici deux niveaux possibles d'intervention des contraintes définies par l'utilisateur. Il peut exploiter des contraintes dans la phase d'extraction des bi-ensembles, mais aussi dans la phase de construction de la bi-partition. Dans le prochain chapitre nous allons montrer comment une telle approche peut être mise en œuvre à partir de notre cadre L2G, et quels sont les nouveaux problèmes que l'on peut traiter et qui n'ont pas de contrepartie dans les approches standard de la classification ou de la co-classification.

Chapitre 4

Contraintes et co-classification

4.1 Introduction

Dans le chapitre précédent nous avons introduit un nouveau cadre pour la co-classification. L'idée est d'utiliser une collection de motifs locaux (e.g., des bi-ensembles tels que les concepts formels) pour construire une bi-partition (éventuellement chevauchante). Nous avons également introduit la possibilité d'extraire une collection de motifs locaux qui, en satisfaisant un certain nombre de contraintes, pourrait donner lieu à une partition plus pertinente.

Nous nous intéressons ici, à la pertinence des bi-partitions. Lorsque l'analyste utilise un algorithme de classification, il/elle a un très faible contrôle sur les groupes qui seront calculés. Typiquement, il est possible de choisir parmi plusieurs métriques ou bien décider d'une stratégie d'initialisation. Tous ces réglages opérationnels sont conceptuellement éloignés d'une spécification déclarative des propriétés souhaitées pour la bi-partition. Nous considérons qu'il est important que les analystes puissent spécifier leurs attentes (intérêt subjectif) au moyen de contraintes et qu'il faudrait des techniques de co-classification qui produisent des résultats cohérents vis-à-vis de ces spécifications. Le modèle simple de [Man97] aide à formaliser ce point de vue. Une classification peut être vue comme un processus d'évaluation d'une requête inductive qui calculerait $\{\phi \in \mathcal{L} \mid q(r, \phi)$ est vrai$\}$ où $\mathbf{r}$ serait une matrice booléenne, $\mathcal{L}$ désignerait le language des bi-partitions sur une telle matrice, et le prédicat q spécifierait les propriétés attendues sur la bi-partition ϕ. Une vision plutôt classique est que ce prédicat va exprimer une contrainte d'optimisation sur la fonction objectif utilisée, e.g., le coefficient τ de Goodman-Kruskal [RF01b] ou la perte d'information mutuelle dans [DMM03]. On peut également trouver d'autres contraintes comme la définition du nombre de bi-clusters, le fait que certains objets (resp. propriétés) doivent (resp. ne doivent pas) être ensemble, etc. Autrement dit, nous aimerions pouvoir réaliser la tâche comme une sélection de bi-partitions en supposant que toutes

les bi-partitions aient été calculées a priori. Nous savons bien qu'un tel calcul est impossible. On comprend donc la nature heuristique des algorithmes de classification qui utilisent des méthodes d'optimisation locales pour la fonction objectif (i.e., la satisfaction de la contrainte d'optimisation globale ne peut pas être garantie). Combiner ces heuristiques avec la satisfaction d'autres contraintes sur les bi-clusters est clairement un problème difficile. A notre connaissance, l'exploitation de contraintes n'a pas encore été étudiée pour la co-classification. L'introduction de contraintes dans des processus de classification mono-dimensionnels (e.g., K-Means, classification hiérarchique) a motivé quelques travaux (cf. Section 1.2.2).

Des contraintes simples ont été considérées comme "must-link" et "cannot-link". La principale motivation de ces études était la classification semi-supervisée (i.e., améliorer les techniques prédictives lorsque l'ensemble d'apprentissage ne contient que peu d'instances étiquetées). Non seulement nous considérons la co-classification au lieu de la classification mais aussi notre point de vue est différent. Puisque certaines des contraintes spécifient l'intérêt subjectif de l'analyste, nous devons les prendre en compte même si elles nous pénalisent du point de vue de la fonction objectif : nous ne cherchons pas de bons bi-clusters grâce aux contraintes mais plutôt de bons bi-clusters malgré les contraintes.

Notre contribution [PRB06a, PRB06c] concerne donc d'abord de nouveaux types de contraintes. A coté des extensions des contraintes "must-mink" et "cannot-link" pour la co-calssification, nous considérons le cas où l'une ou même les deux dimensions sont ordonnées. Dans le contexte de l'analyse de données d'expression booléennes, c'est par exemple le cas des données qui enregistrent l'évolution de l'expression des gènes au cours du temps (cf. Figure 4.1). Nous proposons de spécifier si une collection de bi-clusters doit ou ne doit pas être cohérente vis-à-vis de tels ordres, i.e., les contraintes "intervalle" et "non-intervalle". Notre seconde contribution concerne le cadre algorithmique pour calculer des bi-partitions satisfaisant les contraintes spécifiées. Nous montrons qu'il est possible d'étendre le cadre "L2G" vers la classification sous contraintes, autrement dit qu'il est possible d'exploiter les contraintes fixées au niveau global (sur les bi-clusters) pour en dériver des contraintes exploitables au niveau local (sur les bi-ensembles).

4.2 Contraintes en co-classification

Par la suite, $\mathcal{D}$ représente soit $\mathcal{T}$ soit $\mathcal{G}$. Définissons maintenant quelques contraintes primitives qui nous paraissent intéressantes pour la co-classification.

– **optimisation** Soit $f(\mathbf{r}, P)$ une fonction objectif, soit

$$\mathcal{C}_{opt}(\mathbf{r}, f, P) \text{ est satisfaite ssi } P = \underset{\phi \in \mathcal{L}_{\mathcal{P}}}{\operatorname{argmin}} f(\mathbf{r}, \phi)$$

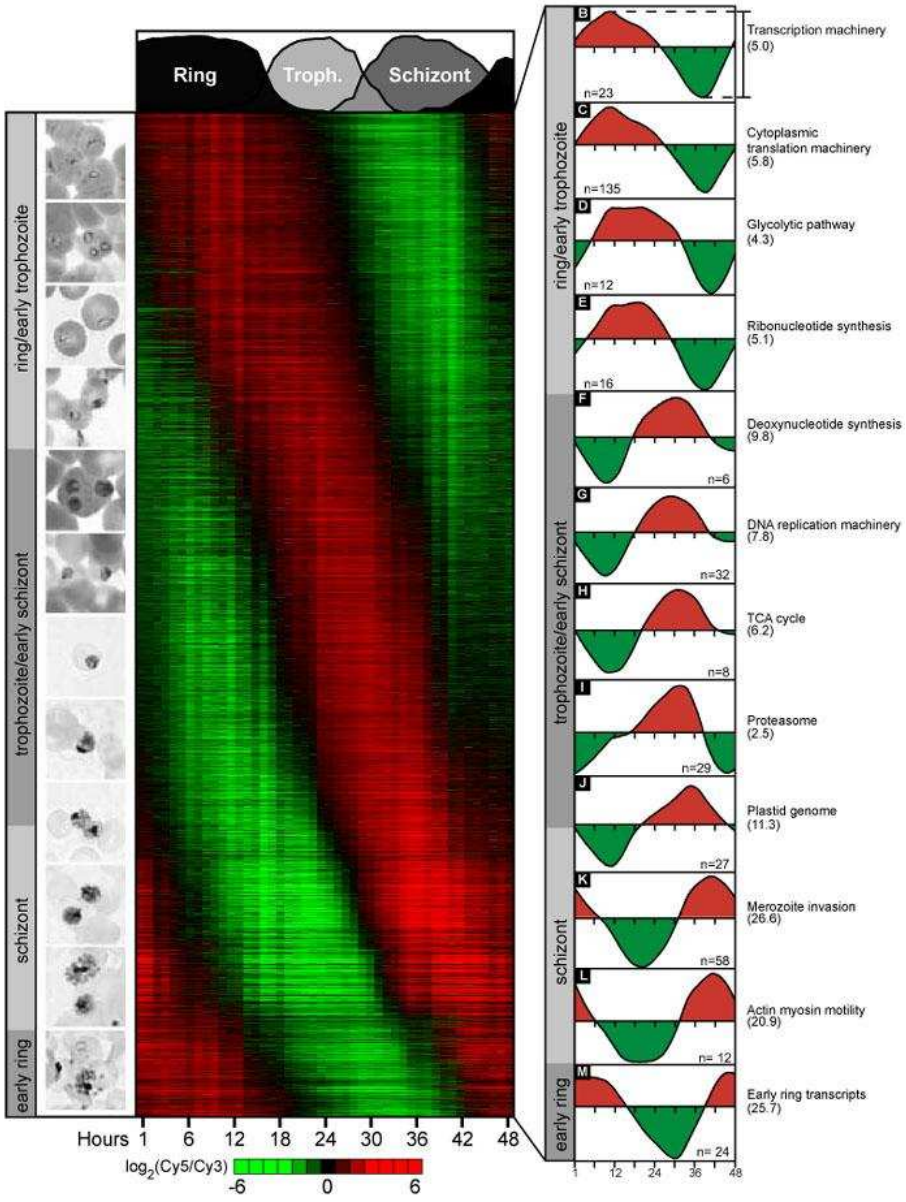


FIG. 4.1 – Un exemple de données d'expression temporelles [BLP+03]

une contrainte d'optimisation, où $\mathcal{L}_{\mathcal{P}}$ est le langage des bi-partitions. Chaque algorithme de co-classification possède sa propre fonction objectif et une approche heuristique locale pour aborder l'optimisation.

- **must-link étendue** Si deux points x_i et x_j sont impliqués dans une contrainte "must-link" étendue, notée $\mathcal{C}_{eml}(x_i, x_j, \mathcal{P})$, ils doivent être dans le même bi-cluster de $\mathcal{P}$.
- **cannot-link étendue** Si deux points x_i, x_j sont impliqués dans une contrainte "cannot-link" étendue, notée $\mathcal{C}_{ecl}(x_i, x_j, \mathcal{P})$, ils ne peuvent pas être dans le même bi-cluster de $\mathcal{P}$.

Nous faisons maintenant l'hypothèse qu'une valeur réelle $s(x_i)$ est associée à chaque élément $x_i \in \mathcal{D}$, d'où

$$s : \mathcal{D} \rightarrow \mathbb{R}$$

Par exemple, $s(x_i)$ pourrait être une mesure temporelle ou spatiale liée à x_i. Dans des données microarrays, où $\mathcal{T}$ est un ensemble de puces à ADN, et $\mathcal{G}$ est un ensemble de gènes, $s(t_i)$ pourrait être le temps d'échantillonnage liée à la puce ADN t_i. D'un autre coté, $s(g_i)$ pourrait mesurer la position absolue dans la séquence d'ADN à part entière (si connue).

La fonction s permet de définir un ordre $\preceq$ sur la dimension $\mathcal{D}$. On dit alors que $x_i \preceq x_j$ si et seulement si $s(x_i) \leq s(x_j)$. Dans suite du chapitre, on dira que, si une fonction s existe sur la dimension $\mathcal{D}$, alors tous les éléments x_i sont ordonnés, i.e., $\forall i, j$ tels que $i < j$, $s(x_i) \leq s(x_j)$. Nous pouvons maintenant introduire deux nouvelles contraintes qui exploitent cette information d'ordre sur la dimension $\mathcal{D}$

- **intervalle** Si un ordre ($\preceq$) est défini sur $\mathcal{D}$, une contrainte "intervalle" sur cette dimension, notée $\mathcal{C}_{int}(\mathcal{D}, \mathcal{P})$, exige que chaque groupe sur $\mathcal{D}$ soit un intervalle : $\forall k = 1 \ldots K$, si $x_i, x_j \in P_k^{\mathcal{D}}$ alors $\forall x_l$ tel que $x_i \preceq x_l \preceq x_j$, $x_l \in P_k^{\mathcal{D}}$.
- **non-intervalle** Si un ordre ($\preceq$) est défini sur $\mathcal{D}$, une contrainte "non-intervalle", notée $\mathcal{C}_{non-int}(\mathcal{D}, \mathcal{P})$ spécifie que les groupes sur $\mathcal{D}$ ne doivent pas être des intervalles : $\forall k = 1 \ldots K$, $\exists x_i, x_j \in P_k^{\mathcal{D}}$, $\exists x_l \in \mathcal{D}$ tel que $x_i \preceq x_l \preceq x_j$, $x_l \notin P_k^{\mathcal{D}}$.

Dans les approches existantes, les contraintes "must-link" et "cannot-link" s'appliquent à l'une des dimensions. Pour une co-classification, il est naturel d'autoriser de telles contraintes sur les deux dimensions, éventuellement simultanément.

Les contraintes "intervalle" et "non-intervalle" sont utiles lorsque l'une ou les deux dimensions est ordonnée (e.g., dans le temps ou dans l'espace). Par exemple, dans l'analyse de données d'expression de gènes, les conditions biologiques peuvent être ordonnées dans le temps (données cinétiques pour suivre l'évolution de l'expression des gènes au cours du temps). Une contrainte "intervalle" peut être utilisée pour trouver des groupes qui ne concernent que des instants adjacents (i.e., qui constituent des intervalles de temps continus), tandis qu'une contrainte "non-intervalle" peut être utilisée pour trouver des groupes qui ne sont pas des intervalles. Dans le premier cas, nous pouvons capturer des associations qui caractérisent chaque stade de la

période d'échantillonnage. Dans le second cas, nous pouvons mettre en évidence des interactions qui sont en quelque sorte non dépendantes du temps.

4.3 Intégration des contraintes dans notre méthode

Nous proposons une extension significative du cadre L2G quand des contraintes définies par l'utilisateur sont spécifiées. L'idée centrale est que, pour calculer une bi-partition qui satisfait une contrainte globale, nous pouvons partir d'une collection de motifs locaux qui ne violent pas une représentation locale de cette contrainte. En utilisant cette contrainte au niveau local (éventuellement associée à une stratégie de propagation), l'idée est qu'il doit être possible d'obtenir une bi-partition qui va satisfaire la contrainte au niveau global. Il faut remarquer que, étant donné l'état de l'art en extraction de bi-ensembles sous contraintes, il existe des algorithmes efficaces pour une grande classe de contraintes. Par exemple, dans nos applications, nous utilisons D-Miner [BRBR05] pour calculer des collections complètes de concepts formels satisfaisant des contraintes définies par l'utilisateur, e.g., des contraintes de taille minimale.

4.3.1 Propagation des contraintes

Discutons maintenant de la réutilisation du principe de CDK-Means et donc des possibilités de traduction des contraintes globales en contraintes locales, et, si nécessaire, comment propager l'information capturée au niveau local jusqu'au niveau global.

Pour traiter des contraintes "must-link", il est possible de forcer cette contrainte dans la collection de bi-ensembles utilisée. En particulier, étant donnée une contrainte "must-link" étendue entre un objet/propriété x_i et un objet/propriété x_j, une collection de bi-ensembles $\mathcal{B}$ satisfait la contrainte ssi $\forall b = (T, G) \in \mathcal{B}$, si $x_i \in T$ (ou G) alors $x_j \in T$ (or G), et vice versa. Comme le coefficient de chaque objet/propriété dans chaque centroïde dépend du nombre de bi-ensembles qui contiennent cet objet/propriété, si x_i et x_j sont impliqués dans une contrainte "must-link" étendue, alors leurs coefficients seront les mêmes dans chaque centroïde. Par suite, les deux seront associés automatiquement au même groupe.

Une condition nécessaire pour une contrainte "cannot-link" étendue est l'exclusion dans $\mathcal{B}$ des bi-ensembles qui violent la contrainte. Puis, étant donnée une contrainte "cannot-link" étendue entre un objet/propriété x_i et un objet/propriété x_j, une collection de bi-ensembles $\mathcal{B}$ satisfait potentiellement la contrainte ssi $\forall b = (T, G) \in \mathcal{B}$, si $x_i \in T$ (ou G) alors $x_j \notin T$ (or G), et vice versa. Cette condition ne peut pas assurer la satisfaction de la contrainte dans la bi-partition finale, et un contrôle ultérieur est

nécessaire. En particulier, à l'étape 2a de CDK-Means (cf. Table 3.4), avant d'ajouter le bi-ensemble contenant x_i (resp. x_j) à un cluster, nous devons nous assurer qu'aucun bi-ensemble contenant x_j (resp. x_i) n'a pas été affecté à ce cluster.

Pour traiter des contraintes "intervalle" et "non-intervalle", il est possible de forcer la même contrainte dans la collection de bi-ensembles utilisée. Toutefois, dans le cas de la contrainte "intervalle", elle pourrait être trop sélective en pratique. A contrario, dans le cas d'une contrainte "non-intervalle", elle pourrait ne pas l'être suffisamment. Pour cette raison, nous proposons de travailler au niveau local sur une contrainte "intervalle" relaxée et une contrainte "non-intervalle" renforcée.

- **max-gap** Étant donné un ordre sur $\mathcal{D}$, une contrainte "max-gap" sur cette dimension, notée $\mathcal{C}_{maxgap}(\mathcal{D}, l, b)$, est satisfaite ssi, pour chaque paire d'éléments consécutifs $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \leq l$.
- **min-gap** Étant donné un ordre sur $\mathcal{D}$, une contrainte "min-gap" sur $\mathcal{D}$, notée $\mathcal{C}_{mingap}(\mathcal{D}, l, b)$, est satisfaite ssi, pour chaque paire d'éléments consécutifs $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \geq l$.

La première est utilisée pour le traitement local de "intervalle" et la seconde aide au traitement de "non-intervalle". Clairement, la satisfaction des contraintes sur la bi-partition résultat n'est pas assurée mais le comportement en phase de calcul est satisfaisant. Il ne s'agit ici que d'une première étape et les stratégies de propagation doivent être étudiées.

4.3.2 Problèmes liés à l'utilisation des contraintes

Nous avons déjà discuté, dans le chapitre précédent, des inconvénients qui peuvent se produire lorsque les contraintes sur les bi-ensembles sont particulièrement fortes.

La solution d'un cluster "poubelle" pour les objets et les propriétés non classés, n'est pas toujours proposable surtout lorsqu'on est dans une situation avec des contraintes de type "cannot-link". En effet, si une contrainte "cannot-link" concernant les éléments x_i et x_j n'est satisfaite par aucun bi-ensemble (i.e., chaque fois que dans un bi-ensemble il y a l'élément x_i on y trouve aussi l'élément x_j), les deux éléments ne seront pas contenus dans la bi-partition finale. Mais, ils ne pourront pas être placés dans une classe poubelle non plus, car une contrainte "cannot-link" empêche cette solution.

Si une telle situation se produit, notre approche ne renvoie pas de solution. Nous dirons que la co-classification est infaisable. On introduit alors deux conditions nécessaires pour que la co-classification sous contraintes ait une solution.

Propriété 4.1 (Condition nécessaire pour la contrainte "must-link") *Soit $\mathcal{C}_{eml}(x_i, x_j)$ une contrainte "must-link" étendue, soit $\mathcal{C}_{all}$ la conjonction de toutes les autres*

contraintes sur les bi-ensembles, $\mathcal{C}_{eml}(x_i, x_j)$ est satisfaisable si $\exists b = (T, G) \in \mathcal{B}$ tel que $x_i \in T$ (ou $x_i \in G$) et $x_j \in T$ (ou $x_j \in G$), et $\mathcal{C}_{all}(b)$ est vraie.

Propriété 4.2 (Condition nécessaire pour la contrainte "cannot-link") *Soit $\mathcal{C}_{ecl}(x_i, x_j)$ une contrainte "cannot-link" étendue, soit $\mathcal{C}_{all}$ la conjonction de toutes les autres contraintes sur les bi-ensembles, $\mathcal{C}_{ecl}(x_i, x_j)$ est satisfaisable si $\exists b_1 = (T_1, G_1) \in \mathcal{B}$, $b_2 = (T_2, G_2) \in \mathcal{B}$, $b_1 \neq b_2$, tels que $x_i \in T_1$ (ou $x_i \in G_1$), $x_j \in T_2$ (ou $x_j \in G_2$), $x_j \notin T_1$ (ou $x_j \notin G_1$), $x_i \notin T_2$ (ou $x_i \notin G_2$), et $\mathcal{C}_{all}(b)$ est vraie.*

La démonstration des deux propriétés est triviale.

Outre les conditions de faisabilité que l'on a vu dans la Section 1.2.2, nous avons aussi ces deux conditions comme directe conséquence de l'utilisation des bi-ensembles. D'ailleurs, le fait d'exiger la satisfaction des contraintes au niveau local, est d'un coté une condition forte, et d'un autre coté est une assurance sur la pertinence de la bi-partition finale (si elle est faisable). Une des perspectives de ce travail concerne l'élargissement des possibilités de satisfaction des contraintes globales.

4.3.3 Propriétés des contraintes

Lorsque l'on dispose d'une collection de bi-ensembles extraite dans les données, il est toujours possible de sélectionner par post-traitement une sous-collection de ces motifs pour exploiter des contraintes au niveau local. On peut aussi considérer le calcul des motifs locaux utiles à une classification sous contraintes donnée, et ainsi exploiter les techniques efficaces d'extraction de motifs sous contraintes (cf. définitions de la Section 1.3.2). En effet, les propriétés duales bien connues de monotonicité et d'anti-monotonicité vis-à-vis des relations de spécialisation permettent d'exploiter de nombreuses contraintes, notamment sur les bi-ensembles (voir, e.g., [BRBR05]). Par suite, si les contraintes au niveau local sont des conjonctions/disjonctions de contraintes (anti-)monotones, elles peuvent être poussées jusque dans la phase d'extraction des bi-ensembles.

Considérons d'abord la contrainte d'inclusion.

Définition 4.1 (contrainte d'inclusion) *Soit $x_i \in \mathcal{D}$, un ensemble $X \subseteq \mathcal{D}$ satisfait une contrainte d'inclusion $\mathcal{C}_{incl}(x_i, X)$ ssi $x \in X$.*

Cette contrainte est monotone et, par suite, la version locale des contraintes "must-link" et "cannot-link" peut être représentée par une conjonction et/ou une disjonction de contraintes (anti-)monotones. En effet, une version locale pour "must-link" $\mathcal{C}_{eml}(x_i, x_j)$ peut être réécrite comme $(\mathcal{C}_{incl}(x_i) \wedge \mathcal{C}_{incl}(x_j)) \vee (\neg\mathcal{C}_{incl}(x_i) \wedge \neg\mathcal{C}_{incl}(x_j))$. Pour la contrainte "cannot-link" $\mathcal{C}_{ecl}(x_i, x_j)$, elle peut être réécrite comme $\neg\mathcal{C}_{incl}(x_i) \vee$

$\neg\mathcal{C}_{incl}(x_j)$. Cette contrainte est anti-monotone puisqu'elle est la disjonction de deux contraintes anti-monotones.

La contrainte "min-gap" est anti-monotone. En effet, soit $b_1 = (X_1, Y_1)$ et $b_2 = (X_2, Y_2)$, tels que $X_1 \subseteq X_2$. On a $S_2 = \{x_h \notin X_2 | x_i \prec x_h \prec x_j\} \subseteq S_1 = \{x_h \notin X_1 | x_i \prec x_h \prec x_j\}$. Par suite, si $|S_2| \geq l$, alors $|S_1| \geq l$. Par contre, la contrainte "max-gap" n'a pas de propriété de monotonicité par rapport à l'inclusion ensembliste. En effet, pour une dimension $D = \{x_1, x_2, \ldots, x_n\}$, une contrainte "max-gap" $\mathcal{C}_{maxgap}(D, 1)$, n'est pas satisfaite par l'ensemble $X_1 = \{x_2, x_3, x_7\}$. Cependant, elle l'est par son sur-ensemble $X_2 = \{x_2, x_3, x_5, x_7\}$, et par son sous-ensemble $X_0 = \{x_2, x_3\}$. Elle n'est donc ni monotone ni anti-monotone[1].

4.4 Exemples de requêtes

Nous sommes maintenant en mesure de fournir des outils pour l'écriture de requêtes concernant les bi-partitions. Dans le chapitre précédent, les contraintes se limitaient aux seuls nombre de bi-clusters et niveau de chevauchement. Dans ce chapitre nous avons défini deux nouvelles contraintes et nous avons étendu le domaine des deux contraintes classiques du "must-link" et "cannot-link". Nous avons donc des nouvelles possibilités d'écriture de requêtes pour le bi-partitionnement :

- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{T}})$ optimisant la distance entre bi-ensembles avec K bi-clusters, où les clusters sur la dimension $\mathcal{T}$ sont des intervalles.
- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{T}})$ optimisant la distance entre bi-ensembles avec K bi-clusters, où l'objet t et la propriété g sont dans le même bi-cluster.
- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{T}})$ optimisant la distance entre bi-ensembles avec K bi-clusters, où les clusters sur la dimension $\mathcal{T}$ ne sont pas des intervalles, et où l'objet t et la propriété g sont dans le même bi-cluster.
- Sélectionner une bi-partition $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{T}})$ optimisant la distance entre bi-ensembles avec K bi-clusters, et différente d'une bi-partition $(\mathcal{P}_0^{\mathcal{T}}, \mathcal{P}_0^{\mathcal{T}})$ précédemment calculée.

La dernière requête est en effet très intéressante. Souvent les attentes des analystes ne trouvent pas leur contrepartie dans la bi-partition calculée sans aucune contrainte. Un moyen pour trouver une bi-partition différente d'une bi-partition $\mathcal{P}_0$ donnée, pourrait être, par exemple, d'établir alétoirement un certain nombre de contraintes "must-link" entre éléments de deux bi-cluster différents de $\mathcal{P}_0$, et de contraintes "cannot-link" entre élément d'un même bi-cluster de $\mathcal{P}_0$.

[1] Elle peut cependant être exploitée dans la phase de génération des candidats.

4.5 Validation

Nous avons étudié l'impact de la contrainte "intervalle" dans deux jeux de données Puces ADN, malaria (précédemment décrit) et drosophila, qui est décrit dans [AFI+02]. Il concerne l'expression des gènes de la Drosophile melanogaster durant son cycle de vie. Les niveaux d'expression de 3 944 gènes sont mesurés pour 57 périodes séquentielles de temps divisées en stade embryonnaire, larvaire et pupaire. Les données d'expression numériques présentées dans [AFI+02] ont été discretisées en utilisant une des méthodes de codage des propriétés décrites dans [BBJ+02] : pour chaque gène g, nous avons affecté la valeur booléenne 1 aux échantillons dont le niveau d'expression était supérieur à X% de la valeur maximale. Nous avons choisi X=35% pour drosophila. Pour les motifs locaux, nous avons utilisé D-Miner [BRBR05] pour extraire les concepts formels dans les deux matrices booléennes dérivées.

Pour évaluer la valeur ajoutée de la contrainte "intervalle", nous mesurons le nombre de sauts à l'intérieur d'une partition.

Définition 4.2 (nombre de sauts) *Soient $\mathcal{D} = \{x_1, \ldots, x_n\}$ un ensemble ordonné de points et $P_k^{\mathcal{D}}$ un cluster sur ces points, il y a un saut lorsque pour un nombre $l > 1$, si $x_i \in C$, $x_{i+l} \in P_k^{\mathcal{D}}$ et $\forall h$ tel que $i < h < i + l$, $x_h \notin P_k^{\mathcal{D}}$. Soit J_k le nombre de sauts dans un cluster $P_k^{\mathcal{D}}$. Etant donnée une partition $\mathcal{P}^{\mathcal{D}} = \{P_1^{\mathcal{D}}, \ldots, P_K^{\mathcal{D}}\}$, le nombre de sauts noté N_J est donc*

$$N_J = \sum_{\forall P_k^{\mathcal{D}} \in \mathcal{P}^{\mathcal{D}}} J_k$$

.

Lorsque $N_J = 0$, les clusters sont exactement des intervalles. Comme la contrainte "intervalle" est une contrainte souple, nous calculons la moyenne des N_J sur un ensemble d'instances de classifications (avec une initialisation aléatoire) pour mesurer l'efficacité de l'approche.

Nous avons d'abord appliqué l'algorithme Cocluster [DMM03] et la version sans contraintes de CDK-Means avec $K = 3$ (i.e., avec l'idée d'identifier les trois stades du développement). L'initialisation des deux algorithmes étant aléatoire, nous avons calculé la moyenne de toutes les mesures sur 100 exécutions. Nous avons mesuré le coefficient N_J, l'indice de Rand par rapport au partitionnement réel disponible dans la littérature, et le coefficient de Goodman-Kruskal pour évaluer la qualité intrinsèque de la bi-partition. Les résultats sont dans la Table. 4.1.

Il y a une différence significative entre les deux jeux de données. Dans malaria, le nombre moyen de sauts (N_J) est déjà petit avec les deux algorithmes. En particulier, si Cocluster obtient un bon coefficient de Goodman-Kruskal, les bi-clusters obtenus avec CDK-Means sont plus cohérents avec la connaissance biologique disponible

	Cocluster			CDK-Means			
Dataset	N_J	*Rand*	τ_S	N_J	*Rand*	τ_S	*CC*
malaria	0.85	0.761	0.494	0.3	0.877	0.438	3.063M
drosophila	6.39	0.692	0.513	4.29	0.601	0.424	1.652M

Tab. 4.1 – Résultats d'une co-classification sans contrainte.

(i.e., la partition à un indice de Rand élevé). D'un autre coté, le nombre de comparaisons est plutôt élevé. Ce que nous attendons dans un tel cas, c'est qu'une approche basée sur les contraintes utilise moins de ressources pour des résultats similaires.

Au contraire, pour Drosophila, les deux algorithmes échouent dans la découverte du partitionnement correct au regard de la connaissance disponible. Le nombre de sauts est dans les deux cas élevé alors que l'indice de Rand est relativement petit. Dans un tel contexte, nous souhaitons obtenir de meilleurs résultats avec une approche basée sur les contraintes.

Nous avons utilisé la contrainte "intervalle" sur les conditions expérimentales. Nous avons appliqué la contrainte "max-gap" pour différentes valeurs du paramètre de cette contrainte, et nous avons étudié l'impact sur la partition finale en mesurant le coefficient N_J, l'indice de Rand, le coefficient de Goodman-Kruskal, et le nombre moyen de comparaisons. Les résultats sont présentés en Figure 4.2 et Figure 4.2 (resp. pour malaria et drosophila).

Pour malaria, nous observons de meilleurs résultats en terme de nombre de sauts (cf. Fig. 4.2a) pour une contrainte "max-gap" de 1 et 2. Dans le second cas (max-gap=2), l'indice de Rand (Fig. 4.2b) est plus élevé, et le coefficient de Goodman-Kruskal (Fig. 4.2c) est maximal (et similaire à celui obtenu sans contraintes, cf Tab. 4.1). Les nombres moyens des comparaisons pour ces valeurs de "max-gap" sont réduits sensiblement (d'un facteur 8, pour max-gap=3, jusqu'à 28 pour max-gap=2). Quand la valeur du max-gap est 1, le nombre moyen de comparaisons est environ 1/1000 de celui obtenu sans la spécification d'aucune contrainte. Pour max-gap=5, nous avons obtenu un coefficient N_J plutôt élevé, mais l'indice de Rand est maximum et similaire à celui que nous obtenions sans contraintes. Dans ce cas, un choix optimal semble être max-gap=2. Il réduit sensiblement le temps de calcul, et produit de bons résultats de classification.

Notons également que notre définition de contrainte "max-gap" fonctionne pour des intervalles de temps ouverts. Cependant, le cycle de développement cellulaire du plasmodium est circulaire. En sélectionnant une contrainte d'intervalle ouvert, nous sommes toujours en mesure d'obtenir une séquence circulaire d'intervalles.

Pour drosophila, l'amélioration est plus évidente. Les résultats d'une co-classification sans contrainte montrent qu'une bonne partition (avec un coefficient de Goodman-

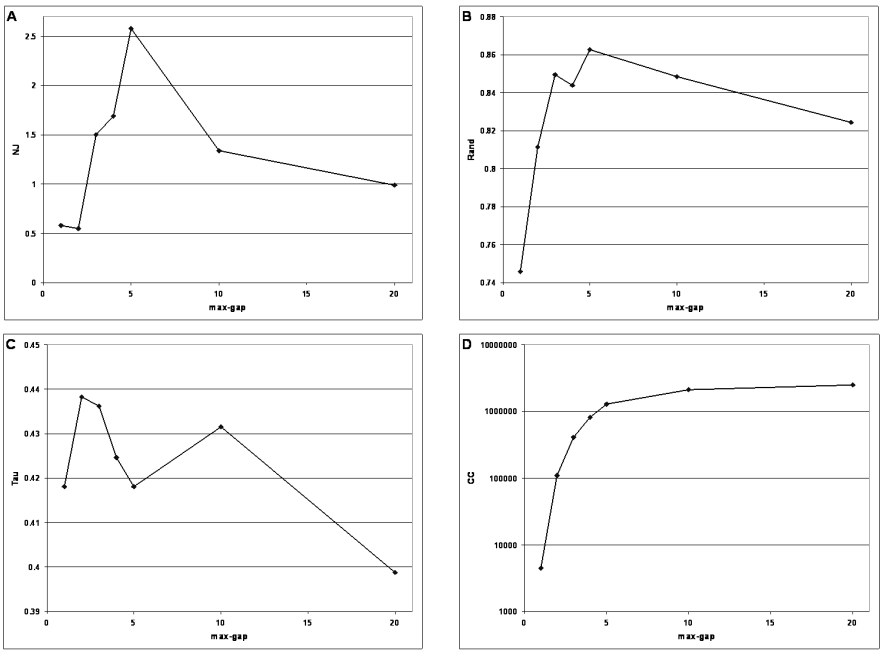


FIG. 4.2 – Résultats pour malaria.

Kruskal élevé) contient beaucoup de sauts. Avec une contrainte max-gap de 2 ou 3, nous pouvons réduire significativement le nombre de sauts (Fig. 4.2a) et augmenter la qualité de la partition (Fig. 4.2b) par rapport à la connaissance biologique disponible. Le fait que, pour ces valeurs de max-gap, le coefficient de Goodman-Kruskal soit minimum, indique que la partition qui satisfait au mieux les contraintes n'est pas forcément la "meilleure". De plus, le nombre moyen de comparaisons est réduit de 60 (max-gap=2) et de 30 (max-gap=3).

4.5.1 Bi-partitions instables et leur caractérisation

Nous avons montré comment la contrainte "intervalle" pouvait aider à la découverte d'intervalles temporels. Pour certains jeux de données (e.g., malaria), une approche sans contrainte produit déjà des intervalles corrects. La question devient : est-il possible de découvrir des associations de gènes différentes qui se produisent entre des points appartenant à des intervalles différents ? Dans quelle mesure les contraintes "intervalle" et "non-intervalle" peuvent-elles être utilisées pour guider la tâche de co-classification quand les algorithmes classiques renvoient des résultats instables ?

Pour répondre à ces questions, nous avons appliqué la contrainte "non-intervalle" aux données concernant les échantillons de la phase adulte du cycle de développement

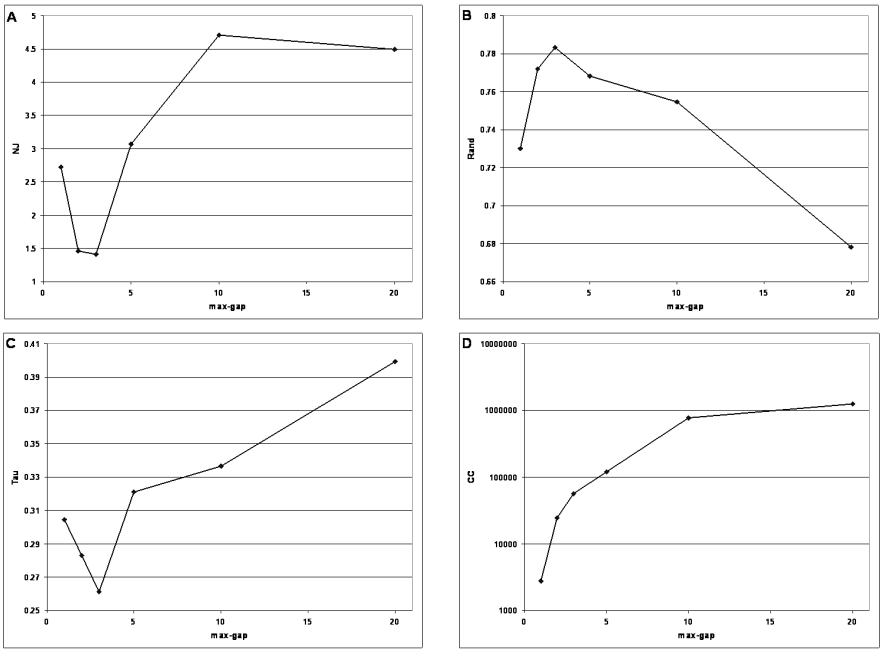


FIG. 4.3 – Résultats pour drosophila.

de la drosophile. Les échantillons de temps t_1 à t_{10} concernent les premiers jours du cycle de vie des individus mâles. Les échantillons de t_{11} à t_{20} concernent les individus femelles. Les résultats sont dans la Table 4.2. Nous avons appliqué les contraintes "intervalle" et "non-intervalle" avec respectivement max-gap=5 et min-gap=5.

Quand nous appliquons CDK-MEANS (avec $k = 2$) sans spécifier aucune contrainte sur ce jeu de données, les deux intervalles $t_1, \ldots, t_{10}$ et $t_{11}, \ldots, t_{20}$ sont bien identifiés pour toutes les 100 exécutions de l'algorithme (la valeur moyenne de N_J est 1). COCLUSTER semble être plus instable. Il renvoie parfois un cluster de mâles et un cluster de femelles, parfois les deux sexes sont mélangés. La valeur moyenne de N_J est 3.45, et l'écart type est 113% de la moyenne. Quand nous imposons la contrainte "non-intervalle", la valeur moyenne de N_J est élevée (environ 6.94), tandis que l'écart type est plus petit (28% de la moyenne) par rapport aux résultats de COCLUSTER. Quand on impose la contrainte "Intervalle", CDK-MEANS renvoie toujours des bi-clusters mâle et femelle parfaits ($N_J = 0$ et $Rand = 1$).

Les résultats montrent qu'au moyen des contraintes "intervalle" ou "non-intervalle", l'utilisateur obtient une forme de contrôle sur la forme de la bi-partition. Sur l'analyse des données concernant les individus drosophiles adultes, un algorithme comme COCLUSTER a parfois trouvé des bi-clusters où le sexe était le paramètre majoritairement discriminant. Parfois il a capturé des interactions entre mâles et femelles. Ce

		τ		$Rand$		N_J	
bi-part.	**inst.**	**mean**	**std.dev**	**mean**	**std.dev**	**mean**	**std.dev**
co :MF	56	0.5605	0.0381	0.82	0.06	0.25	0.61
co :mixed	44	0.1156	0.0166	0.51	0.02	7.52	2.07
co :overall	100	0.3648	0.2240	0.69	0.16	3.45	3.90
cdk :unconst	100	0.4819	0.0594	0.88	0.04	1.00	0.20
cdk :int	100	0.4609	0.0347	1.00	0.00	0.00	0.00
cdk :nonint	100	0.1262	0.0761	0.53	0.04	6.94	1.93

TAB. 4.2 – Résultats pour les individus adultes de la drosophile.

bi-partition	$\lvert\mathcal{B}_1\rvert/\lvert\mathcal{B}_2\rvert$	$\mathcal{P}$**-freq**	$\mathcal{P}$**-conf**	$\mathcal{O}$**-freq**	$\mathcal{O}$**-conf**
males vs. females	0.87	0.6%	91%	6%	78%
mixed	0.97	0.4%	73%	3%	47%

TAB. 4.3 – Mesures d'intérêt pour la caractérisation des données des adultes de la drosophile.

que nous voulons, c'est permettre une supervision de ce processus de classification par la spécification de contraintes.

On se pose maintenant d'autre type de questions. Comment notre technique de caractérisation marche-t-elle dans cette situation particulière ? Comment la caractérisation change-t-elle entre deux bi-partitions différentes ? La qualité de la bi-partition influence-t-elle la pertinence des motifs de caractérisation ?

Nous avons donc utilisé notre technique pour post-traiter la collection de tous les concepts formels contenus dans la matrice. Comme l'on s'y attendait, la caractérisation change, mais nous voulons aussi évaluer ce changement dans l'interprétation de la co-classification. Pour cela, nous avons calculé la moyenne de toutes les mesures d'intérêt (fréquence et confiance), et nous avons sélectionné une instance pour chacun des groupes de solutions de COCLUSTER. Les deux instances ont été choisies en considérant celles avec l'écart à la moyenne minimum. Les mesures d'intérêt ont été calculées pour tous les 5 936 concepts formels, sans contrainte de fréquence ou de confiance. Les résultats sont dans la Table 4.3.

Dans la première bi-partition, la fréquence et la confiance moyenne des règles de caractérisation, sont plus élevées que dans la seconde. Cela est vrai soit pour les règles calculées sur les objets, soit pour celles calculées sur les propriétés. Cela signifie que

les motifs locaux (les concepts formels) reflètent plus la première bi-partition que la deuxième. En d'autres termes, la consistance du premier modèle globale est validée par les associations locales dans la matrice. Le fait que soit la mesure de Goodman-Kruskal, soit la perte d'information mutuelle soient meilleures dans le premier groupe de solutions, atteste le fait que la consistance globale et la consistance locale sont liées. Par conséquent, la caractérisation d'une bi-partition globale à travers des motifs locaux a du sens, et pourrait être utilisée comme un nouveau moyen pour évaluer la qualité d'une bi-partition.

4.6 Conclusion

La co-classification est une approche intéressante en classification conceptuelle. Dans les données catégorielles, elle fournit des bi-partitions qui optimisent, au moins localement, des mesures objectives de la qualité des groupements. L'amélioration de la qualité des groupements reste une tâche difficile dans les processus d'analyse exploratoire des données réelles. Premièrement, il est difficile de capturer les aspects d'intérêt subjectif, e.g., les attentes de l'analyste à partir de sa connaissance du domaine. Puis, quand ces attentes peuvent être spécifiées de façon déclarative, les utiliser durant le processus de calcul est un défi. Pour calculer des bi-partitions qui satisfont des contraintes définies par l'utilisateur, nous avons montré qu'il était possible utiliser un cadre générique de co-classification basé sur les motifs locaux, une approche simple mais puissante. Des nouveaux types de contraintes pour le bi-clusters ont été considérées, e.g., les contraintes "intervalle" et "non-intervalle" pour des données ordonnées. Une perspective à court terme pour cette recherche, est de formaliser les propriétés des contraintes globales (i.e., les contraintes pour le bi-partitions) pouvant être transformées, de façon plus ou moins automatique, dans des contraintes à niveau local. Il faut également étudier des stratégies de propagation de contraintes depuis le niveau local jusqu'au niveau global et ainsi garantir la satisfaction des contraintes fixées par l'analyste (hors contrainte d'optimisation de la fonction objectif) dans les bi-partitions calculées.

Conclusion

Dans cette partie du mémoire nous avons montré la validité du cadre L2G à travers des essais réalisés avec des données benchmark et des données d'expression bien documentées. La caractérisation des bi-clusters à l'aide de motifs locaux, a permis d'identifier des groupes de propriétés intéressantes qui peuvent expliquer les groupes d'objets. La connaissance disponible nous a permis de valider cette approche et a montré, de façon plus générale, qu'une bi-partition peut guider un processus d'extraction de motifs locaux pour en améliorer la pertinence.

En ce qui concerne notre approche de co-classification, les expériences conduites sur les données benchmark ont montré que cette méthode est compétitive avec les autres techniques existantes de co-classification. En revanche, l'application au jeu de données d'expression a montré des différences significatives par rapport à Coclustер.

Premièrement, en considérant des associations (e.g., concepts formels, δ-bi-ensembles) entre objets et propriétés en tant qu'éléments à traiter, nous pouvons capturer des structures significativement différentes par rapport à Cocluster. En effet, ces interactions localement fortes jouent un rôle critique dans notre technique de co-classification (l'assignation d'une classe pour chaque objet/propriété dépend du nombre de bi-ensembles qui la concernent dans chaque centroïde).

D'un point de vue biologique, si on considère les bi-ensembles comme des modules de transcription putatifs (ou des associations fortes de gènes co-régulés et des conditions expérimentales qui ont entraîné cette sur-expression), alors on peut voir un bi-cluster comme le résultat de la somme d'un groupe d'associations locales fermées au sens des connections de Galois. Cela peut expliquer pourquoi les trois étapes du cycle de vie du Plasmodium Falciparum, sont mieux identifiées dans nos résultats plutôt que dans ceux de Cocluster.

Un autre avantage est que CDK-Means peut calculer facilement des bi-partitions avec des classes chevauchantes. C'est une nouvelle possibilité pour les algorithmes de bi-partitionnement. Même si certains algorithmes (e.g., [CC00]) permettent l'extraction de bi-clusters chevauchants, ils ne s'agit pas d'approches de bi-partitionnement. Il est clair que, dans des applications comme l'analyse des puces à ADN, c'est une

valeur ajoutée importante.

Nous avons enfin montré une certaine valeur ajoutée dans l'utilisation des contraintes pour des analyses de données d'expression temporelles. Les contraintes ont été appliquées sur la dimension des conditions expérimentales (objets). On peut considérer d'autres applications pour l'analyse de données biologiques et, e.g., considérer un ordre sur la dimension des gènes (ordre spatial des gènes sur les séquences ADN). Beaucoup d'autres applications reposent sur des données ordonnées, par exemple l'analyse de données géospatiales ou encore la fouille de données textuelles.

Troisième partie

Contribution à l'analyse du transcriptome

Chapitre 5

Contribution à l'analyse de données d'expression

5.1 Introduction

Dans le cadre de la contribution au développement d'outils pour l'analyse du transcriptome, nous allons présenter deux travaux qui, dans un processus d'extraction de connaissances, se situent au début et à la fin de l'enchaînement des différentes étapes. L'étape d'extraction de connaissances pert toute son utilité si elle est exécutée sur un contexte booléen dont on ne connaît pas les propriétés ou les caractéristiques par rapport à la matrice réelle. De plus, même la technique de fouille la plus avancée n'est d'aucune utilité si ses résultats ne sont pas faciles à interpréter.

Nous introduirons donc notre méthode d'évaluation de la discrétisation, que nous avons présentée dans [PLBB04, PB05a], et qui propose un processus guidé pour le choix des méthodes correctes et des paramètres de discrétisation. Nous présenterons également une méthode de visualisation de motifs locaux (e.g., concepts formels) basée sur le post-traitement par classification hiérarchique d'une collection de bi-ensembles. L'utilisation d'un paradigme de visualisation habituellement employé par les biologistes facilite l'analyse des résultats issus de l'étape d'extraction.

Nous détaillerons les deux contributions, et nous terminerons le chapitre par une conclusion.

5.2 Pré-traitement de données d'expressions numériques

L'étape de discrétisation d'un jeu de données d'expression de gènes est cruciale. Le cas le plus simple concerne le calcul d'une matrice booléenne $\mathbf{r} \subset \mathcal{T} \times \mathcal{G}$ qui code une

propriété simple d'expression pour chaque gène dans chaque situation, par exemple la sur-expression. On peut appliquer différents algorithmes et pour chacun on a le choix parmi différentes valeurs de paramètres. Par exemple [BBJ+02] introduit trois techniques pour coder la sur-expression.

- "Mid-Ranged". On identifie la valeur la plus élevée et la valeur la plus petite pour chaque gène, et on considère la moyenne de ces deux valeurs. Puis, pour un gène donné, toutes les valeurs d'expressions qui sont strictement au dessus de cette moyenne, sont codées avec 1 (0 sinon).
- "Max - X% Max". Le seuil est fixé par rapport à la valeur maximale d'expression observée pour chaque gène. On soustrait une pourcentage X de cette valeur et toutes les valeurs d'expression qui sont supérieures à (100 - X)% de la valeur maximum, sont codées avec 1 (0 sinon).
- "X% Max". Pour chaque gène, on considère les situations biologiques dans lesquelles la valeur d'expression est parmi les X% valeurs les plus élevées. Pour ces valeurs d'expression on affecte la valeur 1, et on affecte 0 aux autres.

Ces techniques donnent des points de vue différents du phénomène biologique de sur-expression aucune n'étant supérieure aux autres d'un point de vue biologique. L'impact de la technique choisie et les paramètres utilisés est crucial à la fois pour la qualité et la pertinence des motifs extraits. Par exemple, la densité des données discrétisées, dépend des paramètres de discrétisation, et les cardinalités des ensembles résultants (collection d'itemsets fréquents, règles d'association ou concepts formels) peuvent être très différentes. Nous avons donc besoin d'une méthode pour évaluer les différents codages booléens (techniques et/ou paramètres différents) des mêmes données brutes, et donc d'un cadre pour supporter la décision de l'utilisateur concernant la discrétisation à partir de laquelle le processus d'extraction peut démarrer. Notre thèse est qu'une bonne discrétisation devrait préserver des propriétés qui peuvent être observées dans les données brutes.

Soit $\mathbf{e}$ une matrice de données d'expression. Soit $\{Bin_i,\ i = 1..b\}$ un ensemble d'opérateurs de discrétisation et $\{\mathbf{r}_i,\ i = 1..b\}$ un ensemble de contextes booléens obtenus en appliquant ces opérateurs, $\forall i = 1..b,\ \mathbf{r}_i = Bin_i(\mathbf{e})$. Soit $S : \mathbb{R}^{m,n} \longmapsto \mathbb{R}$ une fonction d'évaluation qui mesure la qualité de la discrétisation d'une matrice d'expression de gènes. On dira qu'un contexte booléen $\mathbf{r}_i$ est plus valide qu'un autre contexte $\mathbf{r}_j$ par rapport à la mesure S si $S(\mathbf{r}_i) > S(\mathbf{r}_j)$. Dans [PLBB04], nous avons étudié une méthode originale pour une telle évaluation. Nous avons suggéré de comparer la similarité entre le dendrogramme généré par un algorithme de classification hiérarchique (e.g., [ESBB98]) appliqué aux données d'expression brutes, et le dendrogramme généré par le même algorithme appliqué à chaque matrice booléenne dérivée de celle-ci. Pour une matrice d'expression $\mathbf{e}$ et deux contextes booléens dérivés $\mathbf{r}_i$ et $\mathbf{r}_j$, on peut choisir la discrétisation qui conduit à un dendrogramme qui soit le plus similaire possible à celui construit sur $\mathbf{e}$. L'idée est qu'une discrétisation qui préserve les similarités dans les profils d'expression peut être considérée plus pertinente. Nous

avons étudié une simple mesure de similarité entre dendrogrammes et nous l'avons validée expérimentalement sur plusieurs jeux de données d'expression [PLBB04].

Soit T un arbre binaire construit sur $\mathcal{T}$. Soit $\mathcal{L} = \{l_1, \ldots, l_n\}$ l'ensemble des m feuilles de T associées à $\mathcal{T}$ tel que, $\forall i \in \{1 \ldots m\}$, $l_i \equiv t_i$ (chaque feuille contient un seul élément, et à chaque élément correspond une feuille). Soit $\mathcal{B} = \{b_1 \ldots b_{m-1}\}$ l'ensemble des $m-1$ nœuds internes de T générés par un algorithme de classification hiérarchique appliqué sur $\mathcal{T}$. On note $r = b_{m-1}$ la racine de l'arbre. On définit les deux ensembles :

$$\begin{aligned} \beta(b_i) &= \{b_j \in \mathcal{B} \mid b_j \text{ est un descendant de } b_i\} \\ \lambda(b_i) &= \{l_j \in \mathcal{L} \mid l_j \text{ est une feuille appartenant au sous} - \text{arbre enraciné en } b_i\}. \end{aligned}$$

On veut mesurer la similarité entre un arbre T et un arbre de référence T_{ref} construit sur le même ensemble d'objets $\mathcal{T}$. Pour chaque nœud b_i de T, on définit le score suivant (noté S_B et appelé **BScore**) :

$$\begin{aligned} S_B(b_i, T_{ref}) &= \sum_{b_j \in \beta(b_i)} a_j \\ a_j &= \begin{cases} \frac{1}{|\lambda(b_j)|}, & si \quad \exists b_k \in T_{ref} \mid \lambda(b_j) = \lambda(b_k) \\ 0, & sinon \end{cases} \end{aligned} \tag{5.1}$$

En d'autres termes, pour un nœud b dans T, son score dépend du nombre de ses nœuds descendants qui correspondent à des nœuds dans T_{ref} (un nœud $b_k \in T_{ref}$ correspond à un nœud b_i si $\lambda(b) = \lambda(b_k)$) et de la valeur de $|\lambda(b)|$. On considère l'inverse de $|\lambda(b)|$ pour donner un poids majeur aux nœuds en bas de la hiérarchie. En effet, si deux feuilles sont associées dans un nœud de l'arbre issu de la matrice discrétisée, et que ces mêmes feuilles sont regroupées dans un nœud de l'arbre de référence, on peut supposer que l'opérateur de discrétisation en a préservé la similarité.

Pour obtenir le score de similarité de T par rapport à T_{ref} (noté S_T et appelé **TScore**), on considère la valeur du **BScore** calculée pour la racine, i.e. :

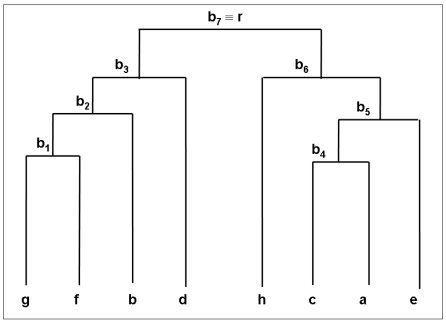
$$S_T(T, T_{ref}) = S_B(r, T_{ref}) \tag{5.2}$$

Il est clairement plus intéressant de normaliser la mesure pour obtenir un score entre 0 (pour un arbre qui est totalement différent de la référence) et 1 (pour un arbre égal à la référence). Comme la valeur maximum du **TScore** dépend de la morphologie de l'arbre, on peut normaliser par $S_T(T_{ref}, T_{ref})$:

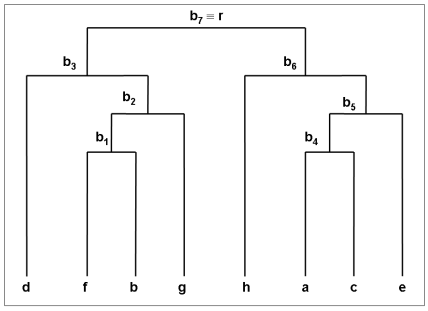
$$\overline{S_T}(T, T_{ref}) = \frac{S_T(T, T_{ref})}{S_T(T_{ref}, T_{ref})} \tag{5.3}$$

$\overline{S_T}(T, T_{ref}) = 0$ signifie que T est totalement différent de T_{ref}, i.e., il n'y a pas de nœuds qui correspondent entre T et T_{ref}. De la même manière, $\overline{S_T}(T, T_{ref}) = 1$

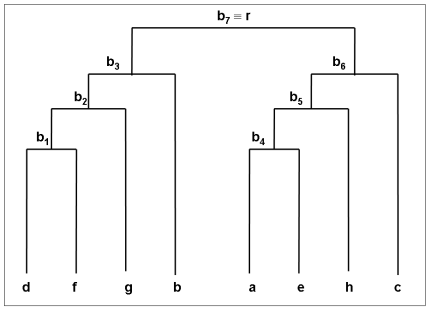
signifie que T est totalement similaire à T_{ref}, i.e., chaque nœud dans T correspond à un nœud dans T_{ref}. Pour deux arbres T_1 et T_2 et une référence T_{ref}, si $\overline{S_T}(T_1, T_{ref}) < \overline{S_T}(T_2, T_{ref})$, alors on dit que T_2 est plus similaire à T_{ref} que T_1 selon le **TScore**.



a)



b)



c)

FIG. 5.1 – Arbre de référence (a) et deux arbres (b et c) construits à partir de deux matrices binaires.

Exemple. Si l'on considère l'arbre de référence T_{ref} dans la Figure 5.1a, et deux arbres T_a et T_b issus de deux matrices binaires différentes (resp. Figure 5.1b et Figure 5.1c), et si on utilise l'Équation 5.3, on obtient :

$$\overline{S_T}(T_a, T_{ref}) = 0.77$$

$$\overline{S_T}(T_b, T_{ref}) = 0.23$$

Comme $\overline{S_T}(T_a, T_{ref}) > \overline{S_T}(T_b, T_{ref})$, la première méthode de discrétisation peut être considérée meilleure pour le jeu de données. En effet, dans T_a, seul le nœud b_1 ne correspond (i.e., il ne partage pas le même ensemble de feuilles) à aucun nœud de T_{ref}, tandis que dans T_b, seuls deux nœuds (b_3 and b_6) correspondent à des nœuds dans T_{ref}.

Dans une matrice de données d'expression on peut appliquer le même processus au dendrogramme des situations et à celui des gènes. Donc on obtient deux scores différents, et si on veut considérer un seul **TScore**, on peut calculer la moyenne entre les deux scores. Cependant, pour forcer le score à être égal à 0 si au moins un des deux scores est égal à 0, on préfère utiliser la racine carrée du produit des deux scores de similaritée :

$$\overline{S_{AT}}\left(T_g, T_s, T_{gref}, T_{sref}\right) = \sqrt{\overline{S_T}\left(T_g, T_{gref}\right) \cdot \overline{S_T}\left(T_s, T_{sref}\right)}$$

où T_g et T_{gref} désignent les arbres sur les gènes, et T_s et T_{sref} ceux sur les situations biologiques.

Dans la suite, on indiquera avec

$$TScore(\mathbf{r}, \mathbf{e})$$

le score de similarité

$$\overline{S_{AT}}\left(T_g, T_s, T_{gref}, T_{sref}\right)$$

où T_g et T_s sont les arbres calculés sur $\mathbf{r}$, et T_{gref} et T_{sref} sont les arbres calculés sur $\mathbf{e}$.

5.3 Visualisation par post-traitement

Un autre problème important, concerne le post-traitement des collections de bi-ensembles. Nous avons besoin de techniques efficaces pour aider la recherche de motifs intéressants. Dans [RPBB04], nous avons introduit une technique de visualisation très familière aux biologistes, car elle reprend la technique de visualisation utilisée dans des outils de classification hiérarchique de données d'expression telles que celle proposée par Eisen [ESBB98]. L'objectif de notre technique est de regrouper des concepts formels similaires en utilisant un algorithme agglomératif de classification hiérarchique (cf. Chapitre 1). Nous avons alors besoin de définir une distance entre deux bi-ensembles, et ensuite une autre distance entre deux classes de concepts formels. Pour la première étape nous utilisons la différence ensembliste symétrique Δ entre deux ensembles S_i et S_j : $S_i \Delta S_j = (S_i \cup S_j) \setminus (S_i \cap S_j)$.

Définition 5.1 *(Distance entre deux bi-ensembles) Soient $b_i = (T_i, G_i)$ and $b_j = (T_j, G_j)$ deux bi-ensembles, la distance d entre b_i et b_j est définie par*

$$d\left(b_i, b_j\right) = \frac{1}{2}\frac{|T_i \Delta\, T_j|}{|T_i \cup T_j|} + \frac{1}{2}\frac{|G_i \Delta\, G_j|}{|G_i \cup G_j|} \tag{5.4}$$

où $|S|$ est le cardinal de S.

Pour calculer la distance entre deux classes de concepts formels, nous pouvons associer un pseudo-bi-ensemble à chaque classe. Un pseudo-bi-ensemble est une représentation unique pour tous les bi-ensembles à l'intérieur d'une classe. Il est composé de deux ensembles flous, un ensemble pour les gènes, et un ensemble pour les situations biologiques : on associe un degré d'appartenance α_i (un nombre réel entre 0 et 1) à chaque élément t_i (ou g_i) de l'ensemble de référence (c'est-à-dire, $\mathcal{T}$ ou $\mathcal{G}$). La valeur 0 indique que l'élément n'appartient pas à l'ensemble. Une valeur 1 indique l'appartenance de l'élément à l'ensemble.

Définition 5.2 *(Pseudo-bi-ensemble) Un pseudo-bi-ensemble est noté* $(T', G', N) \subseteq \mathcal{T}' \times \mathcal{G}' \times \mathbb{N}$ *avec* $\mathcal{T}' = \mathcal{T} \times [0;1]$ *et* $\mathcal{G}' = \mathcal{G} \times [0;1]$. *Le poids* N *indique le nombre de bi-ensembles représentés par le pseudo-bi-ensemble.*

Un pseudo-bi-ensemble (T', G', N) d'un bi-ensemble (T, G) est défini par :

$$\begin{cases} T' = \{(t, \alpha) \mid t \in \mathcal{T},\ \alpha = 1 \text{ si } t \in T,\ \alpha = 0 \text{ sinon}\} \\ G' = \{(g, \alpha) \mid g \in \mathcal{G},\ \alpha = 1 \text{ si } g \in G,\ \alpha = 0 \text{ sinon}\} \end{cases}$$

Le pseudo-bi-ensemble (T', G', N) pour deux bi-ensembles (T'_1, G'_1, N_1) et (T'_2, G'_2, N_2) est calculé de la manière suivante :

$$\begin{cases} T' = \left\{ \left(t, \frac{N_1 \times \alpha_1 + N_2 \times \alpha_2}{N_1 + N_2} \right) \mid \forall t \in \mathcal{T} \text{ et } (t, \alpha_1) \in T'_1 \text{ et } (t, \alpha_2) \in T'_2 \right\} \\ G' = \left\{ \left(g, \frac{N_1 \times \alpha_1 + N_2 \times \alpha_2}{N_1 + N_2} \right) \mid \forall g \in \mathcal{G} \text{ et } (g, \alpha_1) \in G'_1 \text{ et } (g, \alpha_2) \in G'_2 \right\} \\ N = N_1 + N_2 \end{cases}$$

Il est possible de généraliser la distance d pour mesurer la similarité entre pseudo-bi-ensembles. On pourra alors utiliser les opérateurs ensemblistes classiques de la logique floue (indexées par f) :

$$\begin{aligned} S_1 \cup_f S_2 &= \{(t, \max\{\alpha_1, \alpha_2\}) \mid t \in \mathcal{T},\ (t, \alpha_1) \in S_1 \text{ et } (t, \alpha_2) \in S_2\} \\ S_1 \cap_f S_2 &= \{(t, \min\{\alpha_1, \alpha_2\}) \mid t \in \mathcal{T},\ (t, \alpha_1) \in S_1 \text{ et } (t, \alpha_2) \in S_2\} \\ S_1 \setminus_f S_2 &= \{(t, |\alpha_1 - \alpha_2|) \mid t \in \mathcal{T},\ (t, \alpha_1) \in S_1 \text{ et } (t, \alpha_2) \in S_2\} \\ |S_1|_f &= \sum_{t \in \mathcal{T}} \alpha,\ (t, \alpha) \in S_1 \end{aligned}$$

Grâce à cette approche, nous pouvons réduire l'impact de la multiplication des bi-ensembles (tels que les concepts formels) dans des contextes booléens bruités, et nous pouvons faciliter le post-traitement de dizaines de milliers de bi-ensembles.

Exemple. Dans le contexte booléen de la Table 5.1, on peut extraire douze concepts formels (avec au moins un gène et une situation biologique) :

$\mathcal{T}\|\mathcal{G}$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
h_1	0	1	0	1	1	0	1	0
h_2	0	1	0	1	1	0	1	1
h_3	0	0	1	0	0	1	1	0
h_4	1	1	0	1	1	0	1	0
d_1	0	1	0	1	0	1	0	1
d_2	1	0	0	1	1	0	1	0
d_3	1	1	0	1	1	0	1	0
d_4	1	0	0	1	1	0	1	0

TAB. 5.1 – Un contexte booléen

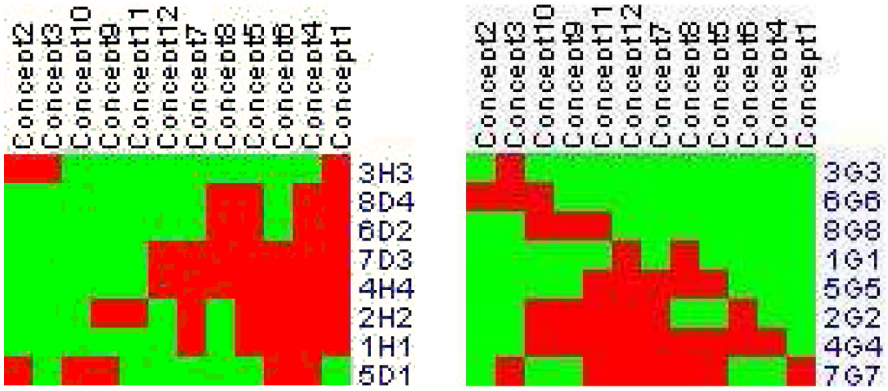
Concept1 : $(\{h_1, h_2, h_3, h_4, d_2, d_3, d_4\}, \{g_7\})$
Concept2 : $(\{h_3, d_1\}, \{g_8\})$
Concept3 : $(\{h_3\}, \{g_3, g_6, g_}7)$
Concept4 : $(\{h_1, h_2, h_4, d_1, d_2, d_3, d_4\}, \{g_4\})$
Concept5 : $(\{h_1, h_2, h_4, d_2, d_3, d_4\}, \{g_4, g_5, g_7\})$
Concept6 : $(\{h_1, h_2, h_4, d_1, d_3\}, \{g_2, g_4\})$
Concept7 : $(\{h_1, h_2, h_4, d_3\}, \{g_2, g_4, g_5, g_7\})$
Concept8 : $(\{h_4, d_2, d_3, d_4\}, \{g_1, g_4, g_5, g_7\})$
Concept9 : $(\{h_2, d_1\}, \{g_2, g_4, g_8\})$
Concept10 : $(\{d_1\}, \{g_2, g_4, g_6, g_8\})$
Concept11 : $(\{h_2\}, \{g_2, g_4, g_5, g_7, g_8\})$
Concept12 : $(\{h_4, d_3\}, \{g_1, g_2, g_4, g_5, g_7\})$

En appliquant un algorithme de classification hiérarchique associé à une technique de visualisation simple[1], on obtient les images (les rectangles) de la Figure 5.2. Les cellules foncées dans le rectangle signifient que le gène (ou la situation biologique) relatif est présent dans le concept formel correspondant. Il faut noter que les groupes de concepts formels similaires peuvent être identifiés en cherchant des zones foncées relativement dense soit dans le rectangle des situations, soit dans le rectangle des gènes.

5.4 Conclusion

Nous avons présenté dans ce chapitre nos deux contributions au prétraitement des données et à la visualisation des motifs locaux. Il s'agit de deux contributions liées directement à l'analyse des données d'expression, et développées dans le cadre du contrat Européen cInQ (IST-2000-26469). Grâce à une méthode basée sur la compa-

[1]Dans ce cas nous avons utilisé l'outil Treeview de [ESBB98].



a) b)

FIG. 5.2 – Rectangles des situations (a) et des gènes (b) résultants d'un algorithme de classification hiérarchique de bi-ensembles

raison des dendrogrammes obtenus par classification hiérarchique des données d'expression réelles et binaires, nous sommes maintenant en mesure de pouvoir mieux choisir nos méthodes de discrétisation ainsi que leurs paramètres. La méthode de visualisation a été utilisée par des biologistes dans le cadre de l'analyse des données SAGE humaines [BPB+06]. Elle a permis d'identifier des regroupements de gènes et de situations biologiques intéressants et de formuler de nouvelles hypothèses biologiques. L'outil a été intégré dans le prototype développé par Besson [BPB+04] qui réunit un ensemble de techniques de fouille de données transcriptomiques dans une seule interface utilisateur.

Chapitre 6

Un scénario d'extraction

6.1 Introduction

Dans ce chapitre nous allons présenter un cas d'utilisation de l'ensemble de nos méthodes présentées dans les chapitres précédents. Nous avons déjà validé expérimentalement la valeur ajoutée de notre approche de co-classification et de caractérisation dans l'application à des jeux de données d'expression. Ici, nous allons considérer un seul jeu de données que l'on va utiliser tout au long d'un processus d'extraction de connaissances, à partir de la phase de pré-traitement des données jusqu'à la phase d'analyse des résultats. Le schéma complet du processus est présenté dans la Figure 6.1.

Le jeu de données que l'on va utiliser a été présenté dans le chapitre précédent. Il s'agit du jeu de données malaria concernant l'expression des gènes du cycle de développement Plasmodium Falciparum. Dans cette application nous allons réaliser un vrai processus d'extraction non supervisé. Nous faisons l'hypothèse que les étapes du développement sont inconnues, mais que nous connaissons leur nombre. Le but de cette application est d'arriver à interpréter les classes d'échantillons biologiques à l'aide des ensembles de gènes qui les caractérisent. Pour cela, le jeu de données d'expression à valeurs réelles, doit d'abord être discrétisé. Le contexte booléen résultant peut être ainsi traité avec un algorithme d'extraction de motifs locaux. 'A partir de ces motifs on peut appliquer notre méthode de co-classification basée sur les contraintes. Pour pouvoir faciliter la tâche d'analyse, on applique enfin notre technique de caractérisation des bi-partitions.

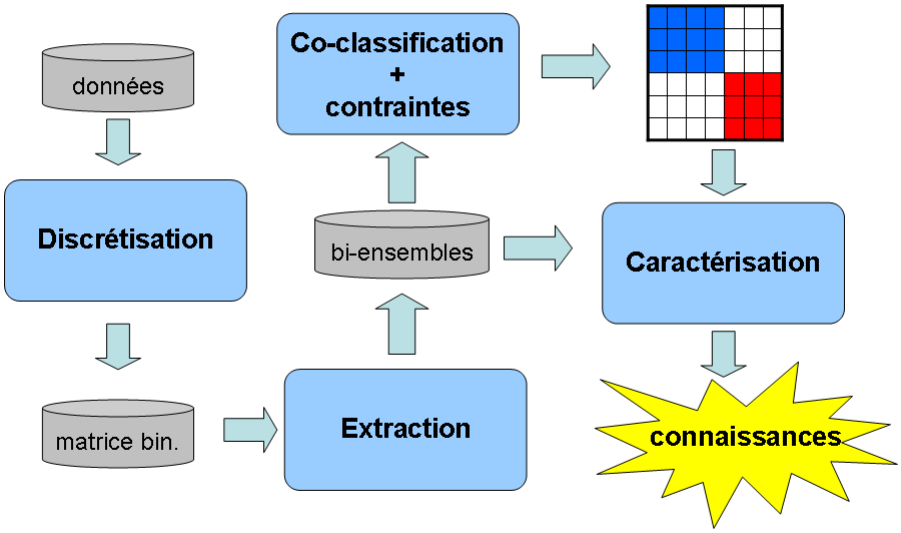


FIG. 6.1 – Schéma complet du processus d'extraction de connaissances

6.2 Pré-traitement de données d'expressions numériques

La première étape consiste à coder les propriétés d'expression de la matrice réelle pour construire un contexte Booléen. On décide de s'intéresser à la sur-expression. Pour notre application, on va considérer uniquement la méthode de discrétisation nommée "Max - X% Max", en faisant varier X entre 1 et 90. Notre ensemble d'opérateurs de discrétisation est donc

$$\{Bin_t,\ t = 1..90\}$$

Soit $\mathbf{e}$ notre matrice d'expression, soit $\{\mathbf{r}_t\}$ l'ensemble des contextes booléens résultant de l'application des opérateurs $\{Bin_t\}$, on a $\mathbf{r}_t = Bin_t(\mathbf{e})$ avec Bin_t défini de la manière suivante :

$$r_{ij} = \begin{cases} 1, & si \quad e_{ij} > \frac{(100-t)\cdot \max_i e_{ij}}{100} \\ 0, & sinon \end{cases} \tag{6.1}$$

La première phase consiste donc à sélectionner le contexte booléen $\mathbf{r}_0$ qui comporte le meilleur score.

$$\mathbf{r}_0 = \underset{\mathbf{r}_t}{\operatorname{argmax}} \{TScore(\mathbf{r}_t, \mathbf{e})\}$$

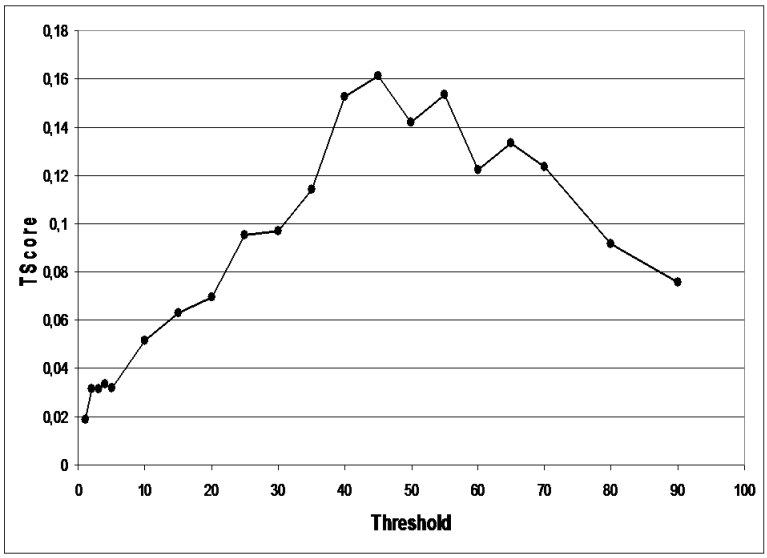


FIG. 6.2 – Valeurs du **TScore** pour différents paramètres X de la méthode "Max - X% Max"

On peut observer le comportement du **TScore** pour des valeurs du seuil t compris entre 1 et 90 (cf. Figure 6.2). La valeur retenue est donc $t = 45$, et $\mathbf{r}_0 = \mathbf{r}_{45}$

On peut donc passer à la deuxième phase de notre processus.

6.3 Extraction de motifs locaux

La deuxième phase de notre processus d'extraction de connaissances consiste à extraire une collection de motifs locaux dans le contexte booléen que l'on vient de matérialiser. Dans ce mémoire nous avons présenté essentiellement deux types de motifs locaux, les concepts formels et les δ-bi-ensembles, mais naturellement, on peut s'intéresser à d'autres types de motifs. Pour chacun des types de motifs d'intérêt on définit une nouvelle contrainte de la manière suivante :

Définition 6.1 (contraintes de type) *Soit* $\mathbf{r}$ *un contexte booléen et* $b = (T, G)$ *un bi-ensemble dans* $\mathbf{r}$*. La contrainte* $\mathcal{C}_{concept}(b, \mathbf{r})$ *est satisfaite si* b *est un concept formel dans* $\mathbf{r}$*. La contrainte* $\mathcal{C}_{\delta-bi-ensemble}$ *est satisfaite si, pour un entier* $\delta \geq 0$ *donné,* b *est un* δ*-bi-ensemble dans* $\mathbf{r}$*.*

On décide de s'intéresser aux concepts formels contenus dans le contexte $\mathbf{r}_0$. Une collection de concepts formels peut être regardée comme l'ensemble $\mathcal{B}$ des motifs du

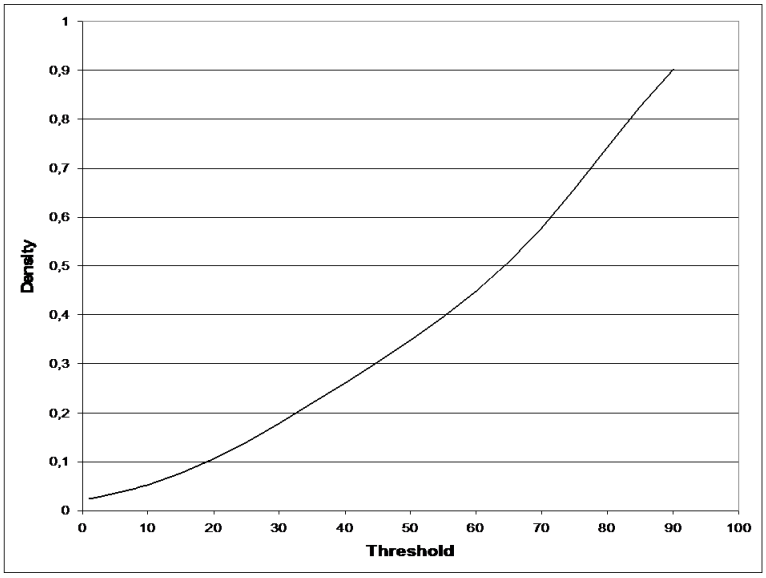


FIG. 6.3 – Valeurs de densité pour différents paramètres X de la méthode "Max - X% Max"

langage $\mathcal{L} = 2^{\mathcal{T}} \times 2^{\mathcal{G}}$ qui satisfont la contrainte $\mathcal{C}_{concept}$. Si l'on n'est pas intéressé par d'autres contraintes on peut chercher à extraire la collection de tous les concepts formels dans $\mathbf{r}_0$ à travers la requête suivante :

$$\mathcal{B}_0 = \{b = (T, G) \in \mathcal{L} \mid \mathcal{C}_{concept}(b, \mathbf{r}_0)\}$$

Lorsque l'on exécute cette requête on tombe sur une très grosse collection de motifs (plus d'un million). En effet, si on regarde la densité de la matrice pour le seuil de 45% (Figure 6.3), elle est de plus de 30%. On peut donc essayer une autre discrétisation qui diminue considérablement la densité, mais qui préserve de manière suffisante la structure interne de la matrice d'expression.

On peut donc utiliser (par exemple) une valeur de $t = 25$. Le nouveau contexte booléen est alors :

$$\mathbf{r}_1 = Bin_{25}(\mathbf{e})$$

On exécute à nouveau la requête pour l'extraction de tous les concepts formels :

$$\mathcal{B}_1 = \{b = (T, G) \in \mathcal{L} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1)\}$$

L'extraction est faisable est donne lieu à une collection de 59 011 concepts formels que l'on va utiliser dans l'étape suivante de notre processus d'analyse.

6.4 Co-classification

L'étape suivante, dans notre processus d'extraction, consiste à déterminer une bi-partition dans les données. Dans cette étape on peut faire intervenir plusieurs types de contraintes. Tout d'abord il y a la contrainte d'optimisation :

$$\mathcal{C}_{opt}(\mathbf{r}, f, \mathcal{P}) \text{ est satisfaite ssi } \mathcal{P} = \underset{\phi \in \mathcal{L}_{\mathcal{P}}}{\operatorname{argmin}} f(\mathbf{r}, \phi)$$

Comme il peut y avoir différentes fonctions objectif à optimiser, on définit des exemples de contraintes d'optimisation.

Définition 6.2 (contraintes d'optimisation) *Soit* $\mathbf{r}$ *un contexte booléen et* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$ *une bi-partition dans* $\mathbf{r}$*. La contrainte* $\mathcal{C}_{\tau}((\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}}), \mathbf{r})$ *est satisfaite si le coefficient* τ *de Goodman-Kruskal est maximisé par* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$*. La contrainte* $\mathcal{C}_{I}((\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}}), \mathbf{r})$ *est satisfaite si la perte d'information mutuelle est minimisée par* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$*.*

Dans le cadre de notre approche CDK-Means, même si nous n'avons pas défini explicitement une fonction objectif, nous pouvons la définir à travers la somme des distances entre tous les éléments d'une classe de bi-ensembles et son centroïde :

$$f_{cdk} = \sum_{k}^{K} \sum_{i}^{N} d(b_i, \mu_k)$$

Cette fonction doit être minimisée par l'algorithme. On peut donc définir la contrainte relative à cette fonction objectif.

Définition 6.3 (contrainte d'optimisation pour CDK-Means) *Soit* $\mathbf{r}$ *un contexte booléen. Soit* $\mathcal{B}$ *une collection de bi-ensembles et* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$ *une bi-partition dans* $\mathbf{r}$*. La contrainte* $\mathcal{C}_{cdk}((\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}}), \mathcal{B}, \mathbf{r})$ *est satisfaite si la bi-partition* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$ *est telle que la fonction objectif* f_{cdk} *est minimisée.*

Nous avons aussi (dans le cas de certains algorithmes) une contrainte portant sur le nombre de bi-clusters cherchés.

Définition 6.4 (contrainte de cardinalité de la bi-partition) *Soit* $\mathbf{r}$ *un contexte booléen. Soit* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$ *une bi-partition dans* $\mathbf{r}$*. La contrainte* $\mathcal{C}_{card}((\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}}), K, \mathbf{r})$ *est satisfaite si la bi-partition* $(\mathcal{P}^{\mathcal{I}}, \mathcal{P}^{\mathcal{G}})$ *contient* K *bi-clusters*

Comme l'initialisation de l'algorithme est aléatoire, on peut imaginer de sélectionner une bi-partition parmi un ensemble de 100 exécutions. Le critère de sélection que nous avons choisi est que la valeur du coefficient de Goodman-Kruskal doit être

maximale. Dans cette application on s'intéresse à des bi-clusters qui impliquent les trois étapes du développement du plasmodium. Pour cela, pour chaque instance, on peut utiliser la contrainte *Intervalle* et la contrainte de cardinalité de la manière suivante :

$$\begin{aligned}(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}}) \quad = \quad & \{(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}) \in \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{cdk}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), \mathcal{B}_1, \mathbf{r}_1) \\ & \wedge \mathcal{C}_{int}(\mathcal{T}, \mathcal{P}^{CT}) \wedge \mathcal{C}_{card}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), 3, \mathbf{r}_1)\}\end{aligned} \tag{6.2}$$

Cette requête se traduit en deux sous-requêtes. La première sert à extraire tous les motifs satisfaisant la version locale de la contrainte *Intervalle*. La deuxième applique CDK-Means sur la collection résultante de la première sous-requête.

1. $\mathcal{B}_2 = \{b = (T, G) \in \mathcal{L} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1) \wedge \mathcal{C}_{maxgap}(\mathcal{T}, l, b)\}$
2. $(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}}) = \{(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}) \in \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{cdk}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), \mathcal{B}_2, \mathbf{r}_1)$

Dans cette application nous avons pris une valeur de $l = 1$. La bi-partition retenue est donc celle qui maximise le coefficient de Goodman-Kruskal

$$(\mathcal{P}_1^{\mathcal{T}}, \mathcal{P}_1^{\mathcal{G}}) = \underset{(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}})}{\operatorname{argmax}}\{\tau_i\}$$

où τ_i est le coefficient de Goodman-Kruskal associé à la bi-partition $(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}})$.

La valeur maximale du coefficient τ est 0.515, tandis que sa moyenne est 0.432 et son écart-type est 0.068. La partition résultante est la suivante :

bi-cluster	$\lvert\boldsymbol{\tau}\rvert$	ring	troph	schiz.	$\lvert\boldsymbol{\gamma}\rvert$
$(P_1^{\mathcal{T}}, P_1^{\mathcal{G}})$	22	21	1	0	927
$(P_2^{\mathcal{T}}, P_2^{\mathcal{G}})$	9	0	9	0	1480
$(P_3^{\mathcal{T}}, P_3^{\mathcal{G}})$	15	0	0	15	1312
total	46	21	10	15	3719

Naturellement, dans un processus complètement non supervisé, on peut imaginer que l'étape du développement de chaque échantillon biologique est une variable inconnue. Si c'était le cas dans notre application, l'imposition d'une contrainte *Intervalle* nous aurait permis de détecter les trois étapes de manière presque parfaite. L'étape suivante consiste à interpréter les résultats via notre méthode de caractérisation des bi-clusters.

6.5 Caractérisation et interprétation des résultats

Dans cette phase du processus d'extraction, nous allons caractériser un des bi-clusters obtenus à l'étape précédente, le but étant d'arriver à fournir une interprétation

qui explique l'association entre les gènes et les échantillons faisant partie de ce bi-cluster. Pour cela, nous introduisons la contrainte de caractérisation.

Définition 6.5 (contrainte de caractérisation) *Soit* $\mathbf{r}$ *un contexte booléen. Soit* $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$ *une collection de bi-clusters et* $b = (T, G)$ *un bi-ensemble. Un bi-cluster* $(P^{\mathcal{T}}, P^{\mathcal{G}}) \in (\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$ *satisfait une contrainte* $\mathcal{C}_{char}\left(b, (P^{\mathcal{T}}, P^{\mathcal{G}}), \mathbf{r}\right)$ *si*

$$(P^{\mathcal{T}}, P^{\mathcal{G}}) = \underset{(P_k^{\mathcal{T}}, P_k^{\mathcal{G}})}{\operatorname{argmax}}\{sim\left(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}})\right)\}$$

Pour pouvoir sélectionner les bi-ensembles selon leurs mesures d'intérêt, nous définissons les contraintes suivantes :

Définition 6.6 (contraintes d'intérêt) *Soit* $\mathbf{r}$ *un contexte booléen. Soit* $(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$ *une collection de bi-clusters et* $b = (T, G)$ *un bi-ensemble. Un bi-cluster* $(P^{\mathcal{T}}, P^{\mathcal{G}}) \in (\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}})$ *satisfait une contrainte* $\mathcal{C}_{max\epsilon_t}\left(b, (P^{\mathcal{T}}, P^{\mathcal{G}}), \varepsilon_t, \mathbf{r}\right)$ *si* $\epsilon_t(T, P^{\mathcal{T}}) \leq \varepsilon_t$. *Il satisfait une contrainte* $\mathcal{C}_{min|T|}\left(b, l, \mathbf{r}\right)$ *si* $|T| \geq l$. *Les contraintes* $\mathcal{C}_{max\epsilon_g}$ *et* $\mathcal{C}_{min|G|}$ *sont définie de manière similaire.*

Nous pouvons maintenant chercher à caractériser, par exemple, le premier bi-cluster. Pour cela on sélectionne tous les bi-ensembles avec $|T| \geq 10$ et $|G| \geq 5$, et qui ne comportent pas d'exception (i.e, avec $\epsilon_t \leq 0$ et $\epsilon_g \leq 0$), via la requête suivante :

$$\begin{array}{rcl} \mathcal{B}_3 & = & \{b = (T, G) \in \mathcal{L} = 2^{\mathcal{T} \times \mathcal{G}} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1) \\ & & \wedge \mathcal{C}_{char}\left(b, (P_1^{\mathcal{T}}, P_1^{\mathcal{G}}), \mathbf{r}_1\right) \\ & & \wedge \mathcal{C}_{min|T|}\left(b, 10, \mathbf{r}_1\right) \\ & & \wedge \mathcal{C}_{min|G|}\left(b, 5, \mathbf{r}_1\right) \\ & & \wedge \mathcal{C}_{max\epsilon_t}\left(b, (P_1^{\mathcal{T}}, P_1^{\mathcal{G}}), 0, \mathbf{r}_1\right) \\ & & \wedge \mathcal{C}_{max\epsilon_g}\left(b, (P_1^{\mathcal{T}}, P_1^{\mathcal{G}}), 0, \mathbf{r}_1\right)\} \end{array} \tag{6.3}$$

Nous obtenons une collection de 89 concepts formels que l'on peut réduire d'avantage en prenant en compte uniquement les bi-ensembles tels que $|T|$ est maximum.

$$\mathcal{B}_4 = \{b = (T, G) \in \mathcal{B}_3 \mid |T| = \max_{b_i \in \mathcal{B}_3} |T_i|\}$$

Cette requête nous permet d'avoir une collection assez petite (8 concepts formels) à analyser. Ils contiennent tous 11 échantillons biologiques et entre 5 et 8 gènes. On peut se demander donc quelle est la fonction des gènes impliqués et on entre dans la phase de vraie interprétation biologique. Si on étudie le gènes impliqués dans ces concepts formels, on s'aperçoit que 7 concepts sur 8 contiennent en moyenne deux gènes appartenant au groupe nommé "cytoplasmic translation machinery", et qui est actif dans l'étape anneau du développement qui est majoritaire dans le bi-cluster que

l'on a décidé de caractériser. En particulier, nous considérons le concept $b_7 = (T_7, G_7)$ où :

$$G_7 = \{b547, opfblob0096, opfb0679, n128_85, j132_9, c242\}$$

et

$$T_7 = \{TP3, TP4, TP5, TP6, TP8, TP9, TP10, TP11, TP12, TP13, TP17\}$$

Les quatre premiers gènes contenus dans G_7 font tous partie du groupe fonctionnel cité auparavant [BLP+03]. Dans un cas réel, l'étude de 4 gènes sur 6 aurait conduit à des résultats biologiquement pertinent et utiles. Nous avons donc montré l'intérêt de notre méthode de caractérisation liée à l'utilisation des contraintes en co-classification, et en général, nous avons montré que l'approche basée sur les requêtes inductives et portant sur la spécification des besoins de l'utilisateur à travers les contraintes, est une approche prometteuse et qui pourrait se révéler utile non seulement dans l'analyse du transciptome, mais également dans d'autres applications.

6.6 Scénarios prototypiques et conclusion

Nous avons présenté dans ce chapitre, une application complète d'un processus d'extraction de connaissances. Toutes les étapes ont été décrites à l'aide de contraintes, et de manière formelle. Ce que l'on vient de décrire est, en effet, un exemple de scénario d'extraction. Si on reprend maintenant la séquence de toutes les requêtes en les paramétrant, (c'est-à-dire, on remplace tous les paramètres fixés par des variables), ce que l'on obtient est un scénario prototypique, qui est un moyen pour transmettre un savoir faire (dans ce cas, l'extraction de bi-ensembles d'intérêt dans un jeu de données d'expression), à l'aide d'une suite de requêtes inductives :

$$\begin{array}{ll}
\text{q1}: & \mathbf{r}_1 = \text{argmax}_{\mathbf{r}_t} \{TScore(\mathbf{r}_t = Bin_t(\mathbf{e}), \mathbf{e})\} \\
\text{q2}: & \mathcal{B}_1 = \{b = (T, G) \in \mathcal{L} = 2^{\mathcal{T}\times\mathcal{G}} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1)\} \\
\text{q3}: & (\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}}) = \{(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}) \in \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{cdk}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), \mathcal{B}_1, \mathbf{r}_1) \\
& \wedge \mathcal{C}_{int}(\mathcal{T}, \mathcal{P}^{CT}) \wedge \mathcal{C}_{card}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), K, \mathbf{r}_1)\} \\
\text{q4}: & (\mathcal{P}_1^{\mathcal{T}}, \mathcal{P}_1^{\mathcal{G}}) = \text{argmax}_{(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}})} \{\tau_i\} \\
\text{q5}: & \mathcal{B}_3 = \{b = (T, G) \in \mathcal{L} = 2^{\mathcal{T}\times\mathcal{G}} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1) \\
& \wedge \mathcal{C}_{char}\left(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}}), \mathbf{r}_1\right) \\
& \wedge \mathcal{C}_{min|T|}\left(b, \sigma_t, \mathbf{r}_1\right) \\
& \wedge \mathcal{C}_{min|G|}\left(b, \sigma_g, \mathbf{r}_1\right) \\
& \wedge \mathcal{C}_{max\epsilon_t}\left(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}}), \varepsilon_t, \mathbf{r}_1\right) \\
& \wedge \mathcal{C}_{max\epsilon_g}\left(b, (P_k^{\mathcal{T}}, P_k^{\mathcal{G}}), \varepsilon_g, \mathbf{r}_1\right)\} \\
\text{q6}: & \mathcal{B}_4 = \{b = (T, G) \in \mathcal{B}_3 \mid |T| = \max_{b_i \in \mathcal{B}_3} |T_i|\}
\end{array}$$

Lorsque un SGBD inductif doit évaluer cette séquence de requêtes, il peut prendre des décisions. Il peut décider, par exemple, d'exécuter la troisième requête comme suit :

1. $\mathcal{B}_2 = \{b = (T, G) \in \mathcal{L} = 2^{\mathcal{T} \times \mathcal{G}} \mid \mathcal{C}_{concept}(b, \mathbf{r}_1) \wedge \mathcal{C}_{maxgap}(\mathcal{T}, l, b)\}$
2. $(\mathcal{P}_i^{\mathcal{T}}, \mathcal{P}_i^{\mathcal{G}}) = \{(\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}) \in \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{cdk}((\mathcal{P}^{\mathcal{T}}, \mathcal{P}^{\mathcal{G}}), \mathcal{B}_2, \mathbf{r}_1)$

ou d'utiliser une autre approche, selon les méta-données dont il dispose. Il est important de noter que, le processus d'extraction de connaissances étant itératif, l'utilisateur peut, à tout moment, revenir à l'arrière, en sélectionnant des paramètres différents. Il s'agit, bien évidemment, d'une première étape vers la définition d'un vrai scénario prototypique, pour lequel on aura besoin de définir un langage de requêtes et donc un ensemble de primitives.

Conclusion et perspectives

Nous avons travaillé au développement de nouvelles méthodes de fouille de données booléennes avec des applications privilégiées dans l'analyse de données d'expression de gènes, i.e., l'analyse du transcriptome. Notre principale contribution réside dans l'exploitation de motifs locaux ensemblistes pour la co-classification sous contraintes d'une part, et la caractérisation de bi-partitions d'autre part. Lorsque nous avons débuté ce travail, l'état de l'art consistait essentiellement en :

- Des techniques de bi-partitionnement sans possibilité de chevauchement entre les bi-clusters calculés et, plus généralement, sans la possibilité de satisfaction de contraintes fixées par l'analyste ;
- De nombreux algorithmes de classification uni-dimensionnelle, avec de premières approches pour une exploitation limitée de contraintes (contraintes "cannot-link" et "must-link") ;
- De multiples techniques d'extraction de motifs ensemblistes locaux (extractions correctes et complètes de tous les (bi-)ensembles satisfaisant certaines contraintes). Le problème est alors de pouvoir exploiter ces collections qui sont généralement de très grande taille.

Nous avons étudié de façon approfondie certains processus d'extraction de connaissances basés sur le bi-partitionnement de (très grandes) matrices de données booléennes. Le domaine d'application qui a motivé notre travail, et qui a été utilisé pour valider la pertinence de nos propositions, est l'analyse du transcriptome, plus particulièrement la fouille de matrices codant des propriétés booléennes d'expression de gènes dans un certain nombre de conditions expérimentales. Dans tous nos développements, nous avons cherché à préserver la généricité des méthodes et algorithmes proposés (e.g., notre approche du bi-partitionnement peut s'appuyer sur différents types de motifs locaux). Au regard de la nature heuristique des algorithmes de classification, nous avons particulièrement soigné l'aspect expérimental en multipliant les mises en œuvre sur différents jeux de données "benchmark" mais aussi sur des données réelles. Même si nous avons mis l'accent sur les applications aux données transcriptomiques, nous avons souvent mentionné la généricité de nos approches et leurs possibles applications dans les nombreux domaines s'appuyant sur la fouille de contextes booléens.

Nos résultats concernent donc principalement deux axes de recherche :

– L'utilisation de motifs locaux pour la construction et l'interprétation de modèles globaux.
  Nous avons travaillé à la définition de cadres pour l'interprétation des bi-partitions au moyen de motifs locaux, mais aussi pour la co-classification à partir de tels motifs. Pour la première contribution, l'idée était de faciliter l'analyse des bi-partitions grâce à des motifs locaux comme les concepts formels : les motifs sont associés aux bi-clusters avec certaines valeurs de mesures d'intérêt qui vont permettre d'en faciliter la caractérisation, et donc l'interprétation. Notre seconde contribution a concerné la construction de bi-partitions à partir d'une collection de bi-ensembles déjà extraite. Notre cadre générique a été expérimenté grâce à une instance basée sur un algorithme de type *k-means*, fonctionnant avec différents types de motifs, plus précisément des concepts formels et une forme d'extension des concepts formels vers la tolérance aux exceptions. Nous avons montré la pertinence du chevauchement entre bi-clusters et sa valeur ajoutée pour l'analyse de données d'expression de gènes. Nous avons ensuite abordé la co-classification sous contraintes. Ainsi, nous proposons de nouvelles contraintes utiles lorsqu'une relation d'ordre (spatial ou temporel) existe sur les dimensions de la matrice à traiter (e.g., les objets correspondent à des situations ordonnées dans le temps). Nous avons initié l'étude des possibilités d'exploitation de contraintes locales (i.e. contraintes sur des motifs locaux) pour la prise en compte de contraintes globales posées sur les bi-partitions. Ce travail doit se poursuivre, typiquement pour mettre en oeuvre des stratégies de propagation des niveaux locaux aux niveaux globaux.
– La définition de nouvelles méthodes d'analyse pour l'extraction de connaissances à partir des données transcriptomiques.
  Faire du sens à partir de grands volumes de données expérimentales est devenu un enjeu crucial pour la science en général, et la biologie moléculaire en particulier. Ainsi, en analyse du transcriptome, il faut s'intéresser à des scénarios d'extraction de connaissances qui permettent de travailler sur des données d'expression de gènes produites par des techniques à haut débit (Puces ADN, technlogie SAGE), et donc à l'échelle de génomes entiers. Les approches basées sur la fouille de données transcriptomiques booléennes sont de plus en plus crédibles car elles ont déjà été utilisées avec succès sur des problématiques biologiques précises (voir, par exemple, le doctorat de J. Besson [Bes05]). Nous avons contribué à fiabiliser l'étape délicate de codage de propriétés booléennes à partir de données numériques. Les mérites et les défauts des approches basées sur les motifs locaux (vs. motifs globaux) sont mieux compris, et de multiples extracteurs de motifs sont aujourd'hui intégrés dans une plate-forme logicielle utilisable par les biologistes avec lesquels nous coopérons. La complémentarité des motifs globaux et locaux a été montrée dans une application à des données d'expression réelles. Cela nous a permis de décrire un scénario prototypique pour l'extraction de connaissances depuis le pré-traitement des données d'ex-

pression de gènes jusqu'à la mise à disposition de connaissances (après post-traitement et interprétation des motifs).

En reprenant ces deux principaux axes de recherche, nous considérons maintenant les perspectives ouvertes par notre travail.

Perspectives sur la complémentarité des types de motifs

Les relations qui existent entre des motifs locaux (e.g. des concepts formels) et des motifs globaux (e.g. des bi-partitions) ne sont pas suffisamment étudiées. Nous avons ouvert de nouvelles pistes de recherche qui exploitent la complémentarité entre certains types de motifs locaux et globaux. Ainsi, nous avons montré que les motifs locaux qui désignent des "interactions" localement fortes pouvaient être utilisés pour conduire à des bi-partitions équivalentes ou meilleures que celles qui sont obtenues par les méthodes heuristiques n'exploitant pas de tels motifs. Bien évidemment, de nombreuses alternatives à notre instance CDK-Means méritent d'être étudiées. En premier lieu, le choix d'un algorithme de type *k-means* peut être discuté. On peut envisager l'adaptation d'autres types d'algorithmes utilisant des approches différentes.

La complexité de CDK-Means est liée à la taille des collections de motifs locaux utilisées. Pour limiter les impossibilités de calcul, une piste de travail possible serait à identifier les bonnes contraintes (les contraintes de taille des collections ne sont probablement pas suffisantes), qui permettraient de pouvoir garantir la faisabilité d'un calcul CDK-Means. Il serait aussi intéressant d'étudier la possibilité d'un bi-partitionnement pendant la phase d'extraction des bi-ensembles. L'une de difficultés majeures est que les bonnes propriétés que nous savons exploiter pour optimiser le calcul de motifs locaux (monotonicité des contraintes de taille ou de nombreuses contraintes syntaxiques) sont intrinsèquement locales, et donc non suffisantes pour répondre aux besoins d'optimisation des fonctions objectifs en classification. Nous pouvons imaginer que des contraintes qui seraient utiles pour une co-classification basée sur des motifs locaux vont devoir prendre en compte, à la fois, la couverture de la matrice, et une notion de distance entre motifs. Ceci ne pourra d'ailleurs être réalisé qu'au moyen de solveurs heuristiques.

Nos résultats sur la gestion de contraintes spécifiées par l'analyste qui veut effectuer une co-classification sont très préliminaires. Cependant, nous avons montré comment l'information contenue dans les motifs locaux pouvait être utilisée pour résoudre de manière simple des problèmes plutôt compliqués lorsqu'ils sont considérés au niveau global (e.g. l'exploitation de contraintes d'intervalle). Ceci soulève néanmoins de nombreuses questions ouvertes. Ainsi, nous avons vu que les contraintes d'intervalle se propageaient souvent efficacement du niveau local au niveau global. Des stratégies de propagation paraissent donc envisageables pour ce type de contrainte. D'une façon plus générale, la propagation des contraintes du niveau local au niveau global mérite d'être analysée et formalisée. Intuitivement, il semble naturel de vouloir faire un

rapprochement entre les propriétés de propagation des contraintes et les propriétés de monotonicité et d'anti-monotonicité bien connues pour les niveaux locaux. Dans l'optique des systèmes de gestion de bases de données inductives, pouvoir planifier l'exécution des solveurs selon les propriétés des contraintes qui apparaissent dans une requête donnée est un point crucial. Les recherches en ce sens se poursuivent pour le domaine de motifs locaux. Nous suggérons d'adopter ce type de démarche au cours des processus de construction de bi-partitions. Une autre perspective sur la complémentarité entre types de motifs consiste à améliorer des méthodes prédictives par le calcul de meilleurs descripteurs. Dans cette optique, nous venons de débuter une collaboration avec l'Institut Jozef Stefan de Ljubljana (Slovenie) pour l'utilisation de notre technique de caractérisation, non pas pour interpréter des bi-clusters, mais pour construire des descripteurs (projet Européen IQ IST-FET FP6-516169).

Scénarios de découverte de connaissances

Dans le chapitre 6, nous avons détaillé une application de l'ensemble des méthodes décrites dans ce mémoire à un jeu de données d'expression. Il s'agissait, dans ce cas, d'une application de redécouverte afin de valider la pertinence de l'approche L2G. Nous sommes également impliqué dans des scénarios réels comme, par exemple, l'analyse de données SAGE humaines en collaboration avec le laboratoire CGMC (UMR 5534 CNRS et Université Lyon 1, équipe Dr. O. Gandrillon). Ce type d'applications pilotes est nécessaire à la validation qualitative des progrès obtenus (voir, par exemple, les plans de travail de l'ACI MD46 BINGO 2004-2007 et du contrat Européen IQ IST-FET FP6-516169 2005-2008). Au delà de ces applications en analyse du transcriptome, nous avons débuté l'expérimentation de nos approches sur deux applications en gestion de documents (base de brevets français des dernières décennies et base bibliographique DBLP). De telles applications sont indispensables pour éprouver la qualité de nos propositions, et notamment la généricité des concepts proposés. Elles devraient être développées plus avant.

Le but ultime de la communauté émergente travaillant sur les bases de données inductives est d'identifier les primitives (disons le langage) qui permettent de décrire des processus de découverte de connaissances comme des successions de requêtes. Aujourd'hui, nous savons formuler des requêtes inductives relativement complexes portant sur des motifs locaux (e.g. des ensembles, des règles d'association ou des épisodes) et donc décrire certaines étapes du processus d'extraction. Cependant, nous restons plutôt démunis pour la rédaction de requêtes portant simultanément sur des motifs et les données, ou encore des requêtes portant sur des motifs globaux ou modèles. Notre travail montre la possibilité d'enrichir notre potentiel d'interrogation par des requêtes de caractérisation. Ainsi, si nous imaginons une syntaxe "à la SQL", nous pourrions suggérer d'écrire une requête comme :

```
SELECT       Bi-set.t, Bi-set.g, conf(Bi-set.t,Bi-cluster.t) AS c
FROM         Bi-set CHAR_JOIN Bi-cluster, Bi-partition
WHERE        Bi-cluster.id_part=Bi-partition.id
             AND Bi-set.type="formal_concept"
             AND interval(Bi-cluster.t)="TRUE"
             AND Bi-partition.K=3
             AND conf(Bi-set.t,Bi-cluster.t)>0.9
OPTIMIZE BY  Bi-partition.Tau
SORT BY      c DESC ;
```

Cette requête est supposée fournir tous les concepts formels qui caractérisent une bi-partition à trois bi-clusters, dont les ensembles d'objets forment des intervalles, et ayant une confiance supérieure à 90%. Dans cet exemple, la caractérisation est considérée elle-même comme un opérateur. `OPTIMIZE` est une primitive qui, parmi toutes les bi-partitions qui satisfont les contraintes, sélectionne la meilleure par rapport à une mesure donnée. Nous ne voulons pas ici nous poser la question de la matérialisation (i.e. du stockage) a priori des bi-ensembles et des bi-partitions présentes dans les données. Il s'agit d'un problème d'optimisation de la requête en question, une optimisation confiée à un SGBD (inductif) devant étudier l'ensemble des contraintes avant de sélectionner les algorithmes de calcul appropriés. Le choix qui a été fait dans cet exemple est de considérer la caractérisation comme un opérateur, et l'optimisation de la bi-partition comme une primitive. D'autres approches sont naturellement possibles. Ces questions sont au cœur d'une des principales tâches du contrat IQ FP6-516169.

Annexe A

Validation des δ-bi-ensembles

Dans le cadre de l'extraction de motifs locaux tolérants au bruit, plusieurs travaux ont été réalisés au sein de l'équipe. Le but de cette annexe est de présenter une évaluation expérimentale des différentes approches pour permettre au lecteur de pouvoir situer les δ-bi-ensembles dans le panorama des bi-ensembles résistants au bruit. Nous avons analysé le comportement de quatre classes de bi-ensembles dans des données synthétiques et des données réelles. Outre les δ-bi-ensembles (ici FBS), nous avons considéré aussi des collections de concepts formels, de α-β-bi-ensembles [BRB04b] (ici CBS) et de DR-bi-ensembles [BRB06] (ici DRBS). Les α-β-bi-ensembles sont des bi-ensembles maximaux avec un nombre borné d'exceptions par ligne et par colonne. En plus de cette caractéristique, il faut qu'aucune ligne (resp. colonne) extérieure au bi-ensemble soit identique, sur les colonnes (resp. lignes) du bi-ensemble, à une des lignes (resp. colonnes) contenues dans le bi-ensemble. Dans [BRB04b], les auteurs proposent un algorithme pour post-traiter une collection de concepts formels préalablement extraite afin de fusionner les concepts les plus proches et obtenir ainsi une collection de α-β-bi-ensembles. Les DR-bi-ensembles sont des bi-ensembles maximaux avec un nombre borné de lignes et de colonnes et tels que chaque ligne (resp. colonne) externe contienne au minimum ϵ exceptions en plus du nombre maximum d'exceptions contenues dans le bi-ensemble. Pour les extraire les auteurs ont proposé une adaptation de l'algorithme Dual-Miner [BGKW03], qui utilise une stratégie d'énumération des bi-ensembles qui permet un élagage efficace suivant les contraintes monotones et anti-monotones.

D'abord, nous présenterons les résultats des extractions dans des jeux de données bruités artificiellement. Ensuite, nous analyserons les résultats sur un jeu de données médicales. Nous avons évalué soit les performances et la taille des collections, soit la pertinence des bi-ensembles extraits.

A.1 Expériances sur des données artificielles

Nous présentons d'abord la méthode d'évaluation. On nomme **r** un jeu de données de référence, c'est-à-dire, un jeu de données qui est supposé être sans bruit. Ensuite nous dérivons différents jeux de données en ajoutant une quantité de bruit uniforme (i.e., pour un niveau de X% de bruit, chaque valeur dans la matrice est inversé aléatoirement avec une probabilité de X%). Notre objectif est de comparer la collection de concepts formels contenue dans le jeu de données de référence avec des collections de bi-ensembles résistants au bruit, extraites dans les matrices bruitées.

Pour mesurer la pertinence de chaque collection extraite ($\mathcal{B}_e$) par rapport à la collection de référence ($\mathcal{B}_r$), nous avons analysé la présence d'un sous-ensemble de la collection de référence dans chacun d'entre eux. Pour cela, nous pouvons identifier les bi-ensembles qui ont la surface la plus grande en commun avec la référence. Notre mesure, nommé σ prend en compte la surface commune et est définie de la manière suivante :

$$\sigma(\mathcal{B}_r, \mathcal{B}_e) = \frac{\rho(\mathcal{B}_r, \mathcal{B}_e) + \rho(\mathcal{B}_e, \mathcal{B}_r)}{2}$$

où ρ est calculé de la façon suivante :

$$\rho(\mathcal{B}_1, \mathcal{B}_2) = \frac{1}{\sharp\mathcal{B}_1} \sum_{(X_i,Y_i)\in\mathcal{B}_1} \max_{(X_j,Y_j)\in\mathcal{B}_2} \frac{\sharp(X_i \cap X_j) \cdot \sharp(Y_i \cap Y_j)}{\sharp(X_i \cup X_j) \cdot \sharp(Y_i \cup Y_j)}$$

Ici, $\mathcal{B}_r$ est la collection des concepts formels calculée dans la matrice de référence, $\mathcal{B}_e$ est une collection de motifs dans une matrice bruitée. Lorsque $\rho(\mathcal{B}_r, \mathcal{B}_e) = 1$, tous les bi-ensembles appartenant à $\mathcal{B}_r$ ont des instances identiques dans la collection $\mathcal{B}_e$. De la même manière, lorsque $\rho(\mathcal{B}_e, \mathcal{B}_r) = 1$, tous les bi-ensembles appartenant à $\mathcal{B}_e$ ont des instances identiques dans la collection $\mathcal{B}_r$. Par conséquent, lorsque $\sigma = 1$, les deux collections sont idéntiques. Des valeurs élevées de σ, ne signifient pas seulement que nous pouvons trouver tous les concepts formels de la collection de référence dans la matrice bruitée, mais aussi que la collection bruitée ne contient pas beaucoup de bi-ensembles très différents de ceux de référence.

Dans cette expérience, **r** concerne 30 objets (lignes) et 15 propriétés (colonnes) et il contient 3 concept formels de la même taille et disjoint. En d'autres termes, les concepts formels dans **r** sont $(\{t_1, \ldots, t_{10}\}, \{g_1, \ldots, g_5\})$, $(\{t_{11}, \ldots, t_{20}\}, \{g_6, \ldots, g_{10}\})$, et $(\{t_{21}, \ldots, t_{30}\}, \{g_{11}, \ldots, g_{15}\})$. Ensuite, nous avons généré 40 jeux de données différents en ajoutant à **r** des quantités croissantes de bruit uniforme (1% à 40% de la matrice). Une technique robuste devrait être capable de capturer les trois concepts formels même en présence de bruit. Ainsi, pour chaque jeu de données, nous avons extrait une collection de concepts formels et des collections différentes de motifs tolérant au bruit avec des paramètres différents. Pour la collection de FBS, nous avons considéré des valeurs de δ entre 1 et 6. Ensuite nous avons extrait deux groupes de collections de CBS suivant le paramètre δ (resp. δ') pour le nombre maximum

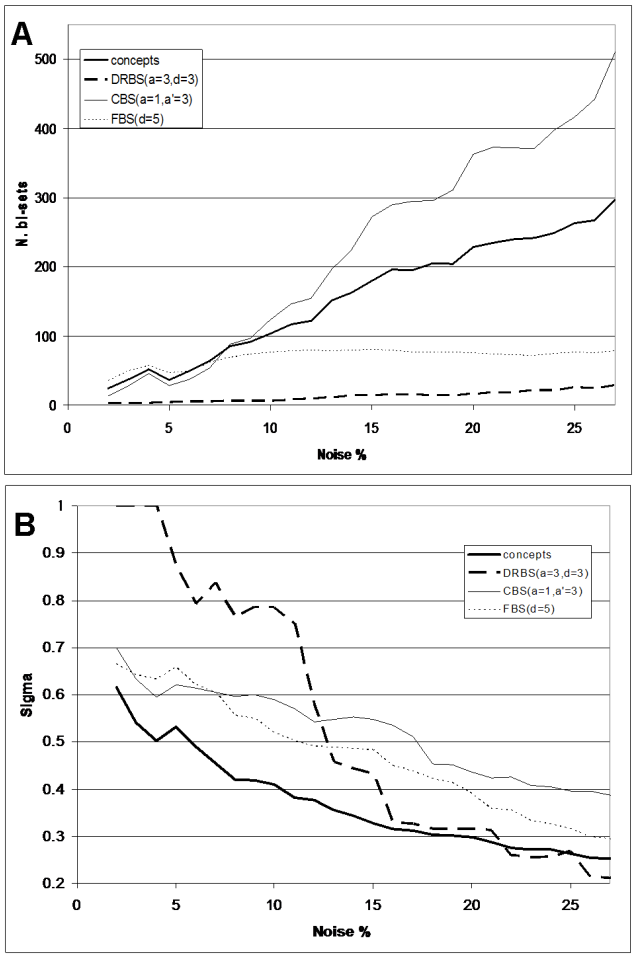


FIG. A.1 – Tailles des différentes collections de bi-ensembles et valeurs respectives de σ par rapport au niveau de bruit pour tous les types de bi-ensemble

de zéros par ligne (resp. par colonne) : un groupe avec $\delta = 1$ et $\delta' = 1 \ldots 3$ et un deuxième groupe avec $\delta' = 1$ et $\delta = 1 \ldots 3$. Enfin, nous avons extrait des collections de DRBS pour chaque combinaison de $\delta = 1 \ldots 3$ et $\epsilon = 1 \ldots 3$.

Dans la Figure A.1, nous montrons seulement les meilleurs résultats par rapport à σ pour chaque classe de motif. La Figure A.1A fournit le nombre de motifs extraits dans chaque collection. Les collections de bi-ensembles tolérant au bruit contiennent presque toujours moins de motifs que les collections des concepts formels. La seule exception est la classe des CBS lorsque $\delta = 1$. La classe des DRBS se comporte mieux que les autres. La taille de ses collections est presque constante, même pour des niveaux de bruit relativement élevés. Le paramètre discriminant est ϵ. Dans la

Formal Concepts						
size	354 366					
time	5s					
FBS						
δ	1	2	3	4	5	6
size	141 983	67 898	39 536	25 851	18 035	13 382
time	19s	10s	6s	4s	3s	2s
DRBS (δ=1)						
ϵ	1	2	3	4	5	6
size	-	75 378	22 882	8 810	4 164	2 021
time	-	1507s	857s	424s	233s	140s

TAB. A.1 – Taille et temps d'extraction pour FBS et DRBS dans meningitis.

Figure A.1B, les valeurs de la mesure σ pour les collections de DRBS naturellement diminuent lorsque le bruit augmente. En général, chaque classe de bi-ensemble tolérant au bruit se comporte mieux que les concepts formels. En termes de pertinence, la classe de motifs DRBS donne aussi les meilleurs résultats. Il est important de remarquer que les résultats pour la classe FBS et CBS ne sont pas significativement différents lorsque leurs paramètres changent. Le paramètre qui semble avoir le plus fort impact sur la valeur de σ pour les motifs DRBS est ϵ. Pour des niveaux raisonnables de bruit ($< 15\%$), l'utilisation des DRBS a du sens. Pour des niveaux plus élevés, les classes CBS et FBS se comportent mieux.

A.2 Expérience sur un jeu de données médicales

Il est important d'obtenir un retour qualitatif sur les motifs résistant au bruit dans un jeu de données réel. Nous avons, donc, considéré le jeu de données médicales décrit dans le Chapitre 2. Notre idée est que, si on cherche des bi-ensembles résistant au bruit plutôt larges, les algorithmes devraient produire des nouvelles associations entre les pairs attribut-valeur (les propriétés booléennes) et les objets. Si l'ensemble des objets et des propriétés à l'intérieur d'un bi-ensembles sont compatibles (e.g., tous les objets sont de type bactérien, et toutes les propriétés sont compatibles avec la méningite de type bactérien), alors on pourrait affirmer que nous avons obtenu des nouvelles informations pertinentes.

Une approche simple pour éviter des motifs non pertinents et pour réduire la taille de la collection est d'utiliser les contraintes de taille sur les composantes des bi-ensembles. Pour cette expérience, nous avons imposé une taille minimale de 10 pour les ensembles d'objets et de 5 pour les ensembles de propriétés. En utilisant D-MINER [BRBR05], nous avons calculé la collection de ces concepts formels, et nous avons obtenu plus de 300 000 motifs dans un temps relativement court (cf.

Table A.1). Naturellement, il est très difficile d'exploiter une telle quantité de motifs. Par exemple, nous ne sommes pas en mesure de post-traiter pour produire les motifs CBS comme décrit dans [BRB04b].

Ensuite, nous avons essayé d'extraire plusieurs collections de motifs FBS et DRBS. Pour les motifs FBS, avec $\delta = 1$ (au plus une exception par colonne) nous avons obtenu une réduction de la taille de la collection de l'ordre de 60%. En utilisant des valeurs de δ entre 2 et 6, la taille est réduite à chaque pas d'un coefficient entre 0.5 et 0.3. Enfin, nous avons utilisé DR-Miner pour extraire des collections différentes de DRBS. Nous avons choisi $\delta = 1$ (au plus une exception par ligne et par colonne) et nous avons utilisé le paramètre ϵ pour réduire ultérieurement la taille de la collection. Le choix $\epsilon = 1$ conduit à une extraction infaisable mais, avec $\epsilon = 2$, la collection résultante est le 80% plus petite que la respective collection de concepts formels. De plus, avec $\delta = 1$ et $\epsilon = 2$ la taille de la collection de DRBS et beaucoup plus réduite que la collection de FBS avec la même contrainte (i.e., $\delta = 1$). D'autre côté, les temps de calcul sont sensiblement plus élevés.

Nous considérons maintenant la pertinence des motifs extraits. Nous avons cherché des bi-ensembles contenant la propriété "presence of bacteria detected in C.S.F. bacteriological analysis" avec au moins une exception. Cette propriété est typiquement vraie dans la méningite de type bactérien [FRC+90, RCB02]. En cherchant des bi-ensembles qui satisfont cette contrainte, nous devrons obtenir des associations entre la méningite bactérienne et des propriétés qui caractérisent cette classe de la pathologie. Nous avons analysé la collection des motifs FBS avec $\delta = 1$. Nous avons obtenu 763 motifs FBS satisfaisant la contrainte choisie. Parmi eux, 124 FBS contiennent seulement un objet de type viral. Nous n'avons pas obtenu de motifs FBS qui contenant plus d'un objet de type viral. Les propriétés qui appartiennent à ces motifs FBS sont soit caractéristiques des cas bactérien, voir des propriétés non discriminants (mais compatibles) telles que l'âge et le sexe du patient. Lorsque $\delta = 2$, le nombre de motifs FBS satisfaisant la contrainte est 925. Parmi eux, 260 contiennent au moins un cas viral de méningite et environ 25 motifs FBS contiennent plus d'un cas viral. Pour $\delta = 5$ les bi-ensembles obtenus ne sont plus pertinents, i.e., les exceptions incluent des propriété booléennes contradictoires (e.g., présence et absence de bactéries).

Nous avons effectué la même analyse sur les motifs DRBS pour $\epsilon = 2$. Nous avons trouvé 24 DRBS plutôt larges. Parmi eux, 2 contiennent aussi un objet bactérien. Un seul motif DRBS semble non pertinent : il contient 3 cas viraux et 8 cas bactériens. En analysant ses propriétés booléennes, nous avons pu constater qu'elles ne sont pas discriminantes par rapport à la méningite bactérienne. Si nous analysons la collection obtenue avec $\epsilon = 3$, il y a un seul motif DRBS qui satisfait la contrainte. Il s'agit d'un bi-ensembles plutôt large impliquant 11 propriétés booléennes et 14 objets. Tous les 14 objets appartiennent à la classe bactérienne, et les 11 propriétés booléennes sont compatibles avec la condition bactérienne de la méningite. D'un côté on pourrait affirmer que les motifs DRBS capturent moins d'associations par rapport aux motifs

FBS, (24 contre 763), mais il faut remarquer aussi que les motifs FBS sont plus redondants par rapport aux DRBS.

Pour résumer, nous avons montré qu'utiliser une contrainte de taille pour réduire la taille de la collection n'est pas toujours suffisant. meningitis est un jeu de données plutôt petit qui conduit à l'extraction de plusieurs centaines de milliers de concepts formels (environ 700 000 si aucune contrainte n'est donnée). En extrayant des bi-ensembles tolérant au bruit, nous pouvons reduire la taille de la collection qui doit être interprétée et cela est crucial pour les processus exploratoires d'extraction de connaissances. D'un point de vue qualitatif, la classe des bi-ensembles DRBS conduit à des résultats plus intéressants. D'un autre côté, les δ-bi-ensembles sont plus faciles à calculer même dans des contextes difficiles, alors que le calcul des collections de motifs DRBS reste infaisable dans beaucoup de cas.

A.3 Discussion

Les deux expériences ont montré les avantages apportés par l'utilisation de techniques d'extraction de bi-ensembles tolérant au bruit dans des données bruitées.

L'utilisation des motifs CBS pourrait être un bon choix lorsque une collection relativement petite de concept formels est déjà matérialisée. Lorsque les données sont denses ou significativement corrélées, comme, par exemple, dans meningitis, l'extraction des CBS échoue même dans des matrices petites. Dans ce cas on pourrait utiliser soit les motifs FBS soit les DRBS. Nos expériences ont montré que la deuxième classe donne des résultats plus pertinent et que la taille des collections est généralement plus petite. Un problème majeur est que cette tâche est infaisable dans des matrices très larges. D'un autre côté, les motifs FBS sont extraits de manière plutôt simple, mais leur sémantique n'est pas symétrique et cela affecte leur pertinence. Une étape de post-traitement pourrait se révéler utile pour éliminer les bi-ensembles qui ne satisfont pas la contrainte d'erreur maximale sur les lignes.

Bibliographie

[AFI+02] M.N. Arbeitman, E.E. Furlong, F. Imam, E. Johnson, B.H. Null, B.S. Baker, M.A. Krasnow, M.P. Scott, R.W. Davis, and K.P. White. Gene expression during the life cycle of drosophila melanogaster. *Science*, 297(5590) :2270–2275, sept. 2002.

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD'93*, pages 207–216, Washington, D.C., USA, 1993.

[BB00] J-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Proceedings PAKDD 2000*, volume 1805 of *LNAI*, pages 62–73, Kyoto, JP, April 2000. Springer-Verlag.

[BBJ+02] C. Becquet, S. Blachon, B. Jeudy, J-F. Boulicaut, and O. Gandrillon. Strong association rule mining for large gene expression data analysis : a case study on human SAGE data. *Genome Biology*, 3(12), Nov. 2002. See http ://genomebiology.com/2002/3/12/research/0067.

[BBM02] S. Basu, A. Banerjee, and R.J. Mooney. Semi-supervised clustering by seeding. In *Proceedings ICML 2002*, pages 27–34, Sydney, Australia, 2002.

[BBM04a] S. Basu, M. Bilenko, and R.J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings ACM SIGKDD 2004*, pages 59–68, Seattle, USA, 2004.

[BBM04b] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings ICML 2004*, pages 81–88, Banff, Canada, 2004.

[BBR00] J-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by mean of free-sets. In *Proceedings PKDD 2000*, volume 1910 of *LNAI*, pages 75–85, Lyon, F, September 2000. Springer-Verlag.

[BBR03] J-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets : a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, 7(1) :5–22, January 2003.

[BDG+04] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D.S. Modha. A generalized maximum entropy approch to bregman co-clustering and matrix approximation. In *Proceedings SIGKDD 2004*, pages 509–514, Seattle, WA, USA, Aug 2004.

[Bes05] J. Besson. *Découvertes de motifs pertinents pour l'analyse du transcriptome : application à l'insulio-résistance*. PhD thesis, Institut National des Sciences Appliquées de Lyon, 2005.

[Bez81] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[BGKW03] C. Bucila, J. E. Gehrke, D. Kifer, and W. White. Dualminer : A dual-pruning algorithm for itemsets with constraints. *Data Mining and Knowledge Discovery Journal*, 7(4) :241–272, Oct. 2003.

[BHHSW03] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of ICML 2003*, pages 11–18, Washington, DC, USA, August 2003.

[BJK02] S. Busygin, G. Jacobsen, and E. Kramer. Double conjugated clustering applied to leukemia microarray data. In *SIAM ICDM Workshop on clustering high dimensional data*, San Diego, CA ,USA, August 2002. SIAM.

[BLP+03] Z. Bozdech, M. Llinás, B. Lee Pulliam, E.D. Wong, J. Zhu, and J.L. DeRisi. The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum. *PLoS Biology*, 1(1) :1–16, October 2003.

[BM98] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Dept. of Information and Computer Sciences, 1998. http ://www.ics.uci.edu/∼mlearn/MLRepository.html. Consulté le 15 juillet 2006.

[Bor86] J-P. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Mathématiques et sciences humaines*, 96 :31–47, 1986.

[BPB+04] J. Besson, R.G. Pensa, S. Blachon, C. Robardet, and J-F. Boulicaut. A simple tool to support gene expression analysis. In *Délivrable D14C - Projet enropéen CinQ (IST-2000-26469)*, avril 2004.

[BPB+06] S. Blachon, R.G. Pensa, J. Besson, C. Robardet, J-F. Boulicaut, and O. Gandrillon. Clustering local patterns to discover biologically relevant knowledge from sage data. Technical report, LIRIS CNRS UMR 5205, July 2006. Submitted to BMC Bioinformatics. 25 pages.

[BPRB06] J. Besson, R.G. Pensa, C. Robardet, and J-F. Boulicaut. Constraint-based mining of fault-tolerant patterns from boolean data. In *Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID 2005, Porto, Portugal, October 3, 2005, Revised Selected and Invited Papers*, volume 3933 of *LNCS*, pages 55–71. Springer-Verlag, 2006.

[BRB04a] J. Besson, C. Robardet, and J-F. Boulicaut. Constraint-based mining of formal concepts in transactional data. In *Proceedings PAKDD 2004*, volume 3056 of *LNAI*, pages 615–624, Sydney (Australia), May 2004. Springer-Verlag.

[BRB04b] J. Besson, C. Robardet, and J-F. Boulicaut. Mining formal concepts with a bounded number of exceptions from transactional data. In *Knowledge Discovery in Inductive Databases 4th International Workshop KDID 2004 Revised Selected and Invited Papers*, volume 3377 of *LNCS*, pages 33–45. Springer-Verlag, 2004.

[BRB05] J. Besson, C. Robardet, and J-F. Boulicaut. Approximation de collections de concepts formels par des bi-ensembles denses et pertinents. In *Proceedings of the French-speaking conference on Machine Learning CAp 2005*, pages 313–328, Nice, France, June 2005.

[BRB06] J. Besson, C. Robardet, and J-F. Boulicaut. Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In *Proceedings of the 14th International Conference on Conceptual Structures ICCS'06*, volume 4068 of *LNAI*, pages 144–157, Aalborg, Denmark, July 2006. Springer-Verlag.

[BRBR05] J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis*, 9(1) :59–82, 2005.

[BS04] A. Berry and A. Sigayret. Representing a concept lattice by a graph. In *Proceedings of the Workshop on Discrete Mathematics and Data Mining (DM&DM)*, volume 144, pages 27–42, November 2004.

[CB02] B. Crémilleux and J-F. Boulicaut. Simplest rules characterizing classes generated by delta-free sets. In *Proceedings ES 2002*, pages 33–46, Cambridge, UK, December 2002. Springer-Verlag.

[CC00] Y. Cheng and G.M. Church. Biclustering of expression data. In *Proceedings ISMB 2000*, pages 93–103, San Diego, USA, 2000. AAAI Press.

[CDG+88] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données*. Dunod, Paris, 1988.

[CST00] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarray classification califano. In *Computational Molecular Biology*, pages 75–85, San Diego, CA, USA, August 2000. AAAI.

[DC02a] N. Durand and B. Crémilleux. Ecclat : a new approach of clusters discovery in categorical data. In *Proceedings ES 2002*, pages 177–190, Cambridge, UK, 2002. Springer-Verlag.

[DC02b] N. Durand and B. Crémilleux. Extraction of a subset of concepts from the frequent closed itemset lattice : A new approach of meaningful clusters discovery. In *Proceedings of the International Workshop on Ad-*

vances in Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD), co-located with the 15th European Conference on Artificial Intelligence (ECAI 2002), pages 24–25, Lyon, France, July 2002.

[DLR77] A. Dempster, N. Laird, and Donald Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.

[DMM03] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic coclustering. In *Proceedings ACM SIGKDD 2003*, pages 89–98, Washington, USA, 2003.

[DR05a] I. Davidson and S.S. Ravi. Agglomerative hierarchical clustering with constraints : Theoretical and empirical results. In *Proceedings PKDD 2005*, volume 3721 of *LNCS*, pages 59–70, Porto, Portugal, 2005. Springer.

[DR05b] I. Davidson and S.S. Ravi. Clustering with constraints : Feasibility issues and the k-means algorithm. In *Proceedings SIAM SDM 2005*, Newport Beach, USA, 2005.

[Dur04] N. Durand. *Extraction de clusters à partir du treillis de concepts : Application à la découverte de communautés d'intérêt pour améliorer l'accès à l'information.* PhD thesis, Université de Caen, november 2004.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[ERR05] M. Elati, C. Rouveirol, and F. Radvany. Apprentissage de signatures de facteurs de transcription à partir de données d'expression. In *EGC 2005*, Paris, France, January 2005.

[ESBB98] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25) :14863–14868, Dec. 1998.

[FRC+90] P. François, C. Robert, B. Cremilleux, C. Bucharles, and J. Demongeot. Variables processing in expert system building : application to the aetiological diagnosis of infantile meningitis. *Med Inform*, 15(2) :115–124, 1990.

[FWE03] B.C.M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *SDM*, San Francisco, CA, USA, May 2003.

[GK54] L.A. Goodman and W.H. Kruskal. Measures of association for cross classification. *Journal of the American Statistical Association*, 49 :732–764, 1954.

[GMS04] A. Gionis, H. Mannila, and J.K. Seppänen. Geometric and combinatorial tiles in 0-1 data. In *Proceedings PKDD 2004*, volume 3202 of *LNCS*, pages 173–184, Pisa, Italy, Sept. 2004. Springer.

[Gov84] G. Govaert. Classification simultanée de tableaux binaires. In E. Diday, M. Jambu, L. Lebart, J. Pages, and R. Tomassone, editors, *Data analysis and informatics III*, pages 233–236. North Holland, 1984.

[Gué90] A. Guénoche. Construction du treillis de galois d'une relation binaire. *Mathématiques, Informatique et Sciences Humaines*, 109 :41–53, 1990.

[Han02] D.J. Hand. Pattern detection and discovery. In *Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 1–12, London, UK, September 2002. Springer.

[Har72] J. Hartigan. Direct clustering of data matrix. *American Statistical Association*, 67(337) :123–129, March 1972.

[HC05] C. Hébert and B. Crémilleux. Mining frequent *delta*-free patterns in large databases. In *Proceedings DS'05*, volume 3735 of *LNCS*, pages 124–136, Singapore, october 2005.

[HKKM97] E-H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering based on association rule hypergraphs. In *DMKD*, 1997.

[JD88] A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.

[Jeu02] B. Jeudy. *Extraction de motifs sous contraintes : application à l'évaluation de requêtes inductives*. PhD thesis, Institut National des Sciences Appliquées de Lyon, December 2002.

[KAKS97] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning : Application in vlsi domain. In *DAC*, pages 526–529, 1997.

[KBCG03] Y. Kluger, R. Basri, J.T. Chang, and M. Gerstein. Spectral biclustering of microarray data : coclustering genes and conditions. *Genome Research*, 13 :703–716, 2003.

[KKM02] D. Klein, S.D. Kamvar, and C.D. Manning. From instance-level constraints to space-level constraints : Making the most of prior knowledge in data clustering. In *Proceedings ICML 2002*, pages 307–314, Sydney, Australia, 2002.

[Koh95] T. Kohonen. Self-organizing maps. In *Springer Series in Information Science*, volume 30. Springer-Verlag, Berlin, Germany, 1995.

[KR90] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data : An Introduction to Cluster Analysis*. John Wiley, 1990.

[LHM98] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings KDD'98*, New York, NY, August 1998.

[LHP01] W. Li, J. Han, and J. Pei. Cmar : Accurate and efficient classification based on multiple class-association rules. In *Proceedings IEEE ICDM 2001*, San Jose, CA, November 2001.

[LO00] L. Lazzeroni and A. Owen. Plaid models for gene expression data. Technical report, Stanford University, 2000.

[LW67] G.N. Lance and W.T. Williams. A general theory of classificatory sorting strategies. *Computer Journal*, 9 :373–380, 1967.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, Berkeley, CA, 1967. University of California Press.

[Man97] H. Mannila. Inductive databases and condensed representations for data mining. In *Proceedings ILPS'97*, pages 21–30, Port Jefferson, USA, October13-16 1997. MIT Press.

[MO04] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis : A survey. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 1(1) :24–45, 2004.

[MT97] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. In *Data Mining and Knowledge Discovery journal*, volume 1(3), pages 241–258. Kluwer Academic Publishers, 1997.

[NH94] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *VLDB'94*, pages 144–155, Santiago de Chile, Chile, September 1994.

[PB05a] R.G. Pensa and J-F. Boulicaut. Boolean property encoding for local set pattern discovery : An application to gene expression data analysis. In *Local Pattern Detection*, volume 3539 of *Lecture Notes in Computer Science*, pages 115–134. Springer, 2005.

[PB05b] R.G. Pensa and J-F. Boulicaut. From local pattern mining to relevant bi-cluster characterization. In *Proceedings IDA'05*, volume 3646 of *LNCS*, pages 293–304, Madrid, Spain, november 2005. Springer-Verlag.

[PB05c] R.G. Pensa and J-F. Boulicaut. Towards fault-tolerant formal concept analysis. In *AI*IA 2005 : Advances in Artificial Intelligence, 9th Congress of the Italian Association for Artificial Intelligence, Milan, Italy, September 21-23, 2005, Proceedings*, volume 3673 of *LNCS*, pages 212–223. Springer, 2005.

[PBB04] R.G. Pensa, J. Besson, and J-F. Boulicaut. A methodology for biologically relevant pattern discovery from gene expression data. In *International Converence on Discovery Science (DS)*, volume 3245, pages 230–241, Padova, Italy, October 2004. Springer-Verlag.

[PBRB05] R.G. Pensa, J. Besson, C. Robardet, and J-F. Boulicaut. Contribution to gene expression data analysis by means of set pattern mining. In *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNCS*, pages 328–347. Springer, 2005.

[Pen03] R.G. Pensa. Algorithmes de clustering et de caractérisation de clusters : application à des données d'expression de gènes. Master report, Politecnico di Torino, November 2003. Edited in French.

[PHM00] J. Pei, J. Han, and R. Mao. CLOSET an efficient algorithm for mining frequent closed itemsets. In *Proceedings ACM SIGMOD Workshop DMKD'00*, pages 21–30, Dallas, USA, May 2000.

[PLBB04] R.G. Pensa, C. Leschi, J. Besson, and J-F. Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data. In *Proceedings ACM BIOKDD'04*, pages 24–30, Seattle, USA, August 2004.

[PRB05] R.G. Pensa, C. Robardet, and J.-F. Boulicaut. A bi-clustering framework for categorical data. In *Proceedings PKDD 2005*, volume 3721 of *LNAI*, pages 643–650, Porto, Portugal, October 2005. Springer-Verlag.

[PRB06a] R.G Pensa, C. Robardet, and J-F. Boulicaut. Co-classification sous contraintes. In *Proceedings of the French-speaking conference on Machine Learning CAp 2006*, pages 155–170, Tr'egastel, France, Mai 2006. Presses Universitaires de Grenoble.

[PRB06b] R.G. Pensa, C. Robardet, and J-F. Boulicaut. Supporting bi-cluster interpretation in 0/1 data by means of local patterns. *Intelligent Data Analysis*, 2006. 20 pages, accepted for publication in June 2006. To appear.

[PRB06c] R.G. Pensa, C. Robardet, and J-F. Boulicaut. Towards constrained co-clustering in ordered 0/1 data sets. In *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems ISMIS 2006*, Bari, Italy, September 2006. 10 pages. To appear as a Springer-Verlag LNAI volume.

[RCB02] C. Robardet, B. Crémilleux, and J-F. Boulicaut. Characterization of unsupervized clusters by means of the simplest association rules : an application for child's meningitis. In *Proceedings of the Seventh International Workshop on Intelligent Data Analysis in Biomedicine and Pharmacology (IDAMAP'02) co-located with the Fifteen European Conference on Artificial Intelligence ECAI'02*, pages 61–66, Lyon, France, July 2002.

[RF01a] C. Robardet and F. Feschet. Comparison of three objective functions for conceptual clustering. In *Proceedings PKDD 2001*, volume 2168 of *LNAI*, pages 399–410. Springer-Verlag, September 2001.

[RF01b] C. Robardet and F. Feschet. Efficient local search in conceptual clustering. In *Proceedings DS'01*, volume 2226 of *LNCS*, pages 323–335, Washington, USA, november 2001. Springer-Verlag.

[RK89] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4), 1989.

[Rob02] C. Robardet. *Contribution à la classification non supervisée : proposition d'une methode de bi-partitionnement.* PhD thesis, Université Claude Bernard - Lyon 1, juliet 2002.

[RPBB04] C. Robardet, R.G. Pensa, J. Besson, and J-F. Boulicaut. Using classification and visualization on pattern databases for gene expression data analysis. In *Proceedings PaRMa'04 co-located with EDBT 2004*, volume 96 of *CEUR Proceedings*, pages 107–118, Heraclion-Crete, Greece, March 2004.

[RR02] C. Robardet and C. Rigotti. Etude de méthodes de recherche locale pour la construction de bi partitions. *RSTI-RIA-ECA*, 16 :705–728, 2002.

[RR05] C. Rouveirol and F. Radvanyi. Local pattern discovery in array-cgh data. In *Local Pattern Detection*, volume 3539 of *LNCS*, pages 135–152. Springer, 2005.

[SE97] E. Schikuta and M. Erhart. The bang-clustering system : Grid-based data analysis. In *Proceedings IDA'97*, volume 1280 of *LNCS*, pages 513–524, London, UK, August 1997. Springer.

[STB+02] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with TITANIC. *Data & Knowledge Engineering*, 42 :189–222, 2002.

[VSL04] V. Ventos, H. Soldano, and T. Lamadon. Alpha galois lattices. In *Proceedings IEEE ICDM 2004*, pages 555–558, Brighton, UK, November 2004. IEEE Computer Society.

[WB93] A. Weil-Barais. *L'homme cognitif.* Presse Universitaire de France, 1993.

[WC00] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings ICML 2000*, pages 1103–1110, Standford, USA, 2000.

[WCRS01] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings ICML 2001*, pages 577–584, Williamstown, USA, 2001.

[WF99] I.H. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, October 1999.

[Wil82] R. Wille. Restructuring lattice theory : an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470. Reidel, Dordrecht, 1982.

[Wil89] R. Wille. Knowledge acquisition by methods of formal concept analysis. In E. Diday, editor, *Data analysis, learning symbolic and numeric knowledge*, pages 365–380. Nova Sciences, 1989.

[Wol70] J.H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioural Research*, 5 :329–350, 1970.

[WWYY02] H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets. In *Proceedings ACM SIGMOD'02*, pages 394–405, Madison, USA, 2002.

[WXL99] K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *CIKM*, pages 483–490, Kansas City, Missouri, USA, November 1999.

[WYM97] W. Wang, J. Yang, and R.R. Muntz. Sting : A statistical information grid approach to spatial data mining. In *VLDB*, pages 186–195, Athens, Greece, August 1997.

[XNJR02] E.P. Xing, A.Y. Ng, M.I. Jordan, and S.J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512, Vancouver, British Columbia, Canada, December 2002.

[YWWY02] J. Yang, W. Wang, H. Wang, and P. Yu. Delta-cluster : capturing subspace correlation in a large data set. In *Data Engineering*, pages 517–528, San Jose, CA, february 2002. IEEE.

[ZH02] M.J. Zaki and C.J. Hsiao. CHARM : An efficient algorithm for closed itemset mining. In *Proceedings SIAM DM'02*, Arlington, USA, Avril 2002. SIAM.