Extraction et usages de motifs minimaux en fouille de données, contribution au domaine des hypergraphes

Céline Hébert

UNIVERSITÉ de CAEN/BASSE-NORMANDIE

U.F.R. de Sciences

École doctorale SIMEM

THÈSE

présentée par

Céline Hébert

et soutenue

le 11 septembre 2007

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

Spécialité Informatique

Arrêté du 07 août 2006

Extraction et usages de motifs minimaux en fouille de données, contribution au domaine des hypergraphes

MEMBRES du JURY

Christophe Rigotti	**Maître de Conférences HDR**	**INSA de Lyon**	**(Rapporteur)**
Marc Sebban	**Professeur**	**Université de St-Étienne**	**(Rapporteur)**
Alain Bretto	**Professeur**	**Université de Caen**	
Bart Goethals	**Senior Researcher**	**Université d'Antwerp Belgique**	
Jean-Daniel Zucker	**Professeur**	**Université de Paris XIII**	
Bruno Crémilleux	**Professeur**	**Université de Caen**	**(Directeur)**

Mis en page avec la classe thloria.

Remerciements

Je tiens tout d'abord à remercier Bruno Crémilleux d'avoir encadré mes travaux de recherche durant les trois dernières années. Ses conseils avisés ont guidé mes premiers pas dans le monde de la recherche et ce travail lui doit beaucoup.

Une partie de ces travaux a été réalisée sous la direction de Alain Bretto. Je le remercie pour m'avoir initiée au domaine des hypergraphes. Il m'a témoigné beaucoup de confiance en me laissant une grande autonomie dans mes travaux.

Je suis particulièrement heureuse que Marc Sebban et Christophe Rigotti aient accepté de rapporter ma thèse. Leur lecture minutieuse et leurs remarques m'ont été précieuses. Plus particulièrement, j'ai beaucoup apprécié les discussions informelles que j'ai eues avec Christophe par le passé, ainsi que son ouverture d'esprit et son enthousiasme.

Je remercie également Bart Goethals et Jean-Daniel Zucker de me faire l'honneur de participer au jury.

Une partie de ce travail est le résultat d'une collaboration avec Sylvain Blachon du laboratoire CGMC de Lyon. Je le remercie pour la patience dont il a fait preuve lors de mon initiation à la génomique.

Un grand merci à mes collègues de bureau et amis François Rioult et Pierre Renaux pour leur enthousiasme, leur soutien et le plaisir qu'ils prennent à partager leur culture scientifique. Un grand nombre de personnes ont également contribué à une ambiance de travail décontractée et agréable durant cette thèse : Thibault, Loïck, Jin, Nicolas, Florent, Cyril, Céline, Guillaume, Antonio, Matthieu, Simon, Léo, Michèle, Françoise, Virginie et j'en oublie sans doute...

Je tiens tout spécialement à remercier les amis qui se sont toujours intéressés, de près ou de loin, à l'avancement de mes travaux et qui n'ont pas manqué de me distraire lorsque le besoin s'en est fait sentir : Eric, Stéphanie, Antoine, Virginie, Olivia, Thomas, Evelyne, Aurélien, JD, Sylvie, Yveline, Sophie, Yannick... Je regrette que Jérôme nous ait quittés trop tôt pour voir ce manuscrit achevé, sa présence ainsi que nos échanges sur nos expériences de thèse m'ont beaucoup manqué ces sept derniers mois.

Merci également aux membres de ma famille pour m'avoir donné l'opportunité de faire de longues études universitaires et pour s'être intéressés à mon domaine de recherche.

Enfin, Hugo m'a patiemment encouragé tout au long de la rédaction de ce manuscrit. Il m'a procuré un soutien sans faille pendant un an et m'a permis de me consacrer à mon travail. Merci Hugo.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Introduction

Contexte

L'extraction de connaissances dans les bases de données (ECBD) est un domaine dont l'essor va de pair avec la multiplication des collectes d'information et l'augmentation des capacités de stockage de données. L'ECBD tire son origine dans la volonté d'appréhender de manière rigoureuse des phénomènes complexes et a pour objectif de découvrir des informations pertinentes à partir de données brutes. Cette discipline se situe à la croisée des bases de données, des statistiques, de l'intelligence artificielle et de l'interface homme-machine. C'est un processus subtil composé de plusieurs phases [FPSS96] : la préparation des données, l'extraction et l'évaluation de connaissances - cette étape est aussi appelée *fouille de données* ou *data mining* en anglais, l'interprétation des résultats.

Ce travail contribue plus particulièrement à la fouille de données, une étape centrale dans un processus de découverte d'information. Il est bien connu que l'exploration de données décrites par un grand nombre d'attributs est un problème algorithmiquement ardu. Dans un premier temps, nous nous intéressons à l'extraction de motifs, un point clé en data mining, dans ce type de données. D'autre part, pour faire face à la quantité de motifs produits par un processus de fouille, nous nous intéressons à la qualité des motifs extraits et des règles qui en découlent. Nous concentrons nos efforts sur les deux aspects suivants : l'extraction d'une couverture de règles qui permet d'éliminer de nombreuses règles redondantes, et l'utilisation de mesures d'intérêt qui constituent des critères fins pour évaluer la qualité des règles. Mais ce travail dépasse le strict domaine de la fouille de données. En effet, nous verrons que les idées que nous avons développées pour la fouille de données orientée motifs sont exploitables dans d'autres domaines comme celui des hypergraphes. Plus précisément, nous montrons comment nos méthodes d'extraction de motifs apportent une solution originale au problème classique du calcul des traverses minimales d'un hypergraphe.

Dans ce mémoire, nous étudions plus particulièrement les motifs libres et δ-libres (voir leur définition aux pages 18 et 20) du point de vue de leur extraction et leurs usages, aussi bien en fouille de données que dans le domaine des hypergraphes. Un motif libre est composé de sous-motifs n'ayant pas de relation entre eux dans les données et constitue la conjonction minimale de facteurs permettant d'expliquer un phénomène. L'obtention de la plupart de nos résultats repose sur la propriété de minimalité que vérifient ces motifs et qui constitue le pivot de nos travaux.

La problématique des données larges

L'extraction de motifs a fait l'objet de nombreux travaux et est aujourd'hui une tâche bien maîtrisée, au moins sur des données du type « caddie de supermarché » [AIS93]. Cependant, il existe des applications qui nécessitent l'exploration de données au format inhabituel et qui restent des contextes d'extraction difficiles comme par exemple l'analyse du transcriptome [PCT+03] ou la toxicité de molécules chimiques [HK03]. Au chapitre 6, nous verrons que l'étude du niveau d'expression des gènes est essentielle pour la compréhension des mécanismes moléculaires contrôlant l'auto-renouvellement des cellules. Or, la taille importante du génome implique que les données d'expression de gènes comportent un grand nombre de descripteurs (ou attributs). Dans de telles données, que nous appellerons *données larges*, les algorithmes usuels d'extraction de motifs échouent [RBCB03]. En revanche, le nombre d'objets étudiés dans les données, qui correspondent à des expériences biologiques coûteuses, est généralement assez faible. À titre d'illustration, le tableau 1 donne un exemple « jouet » de données d'expression de gènes où l'expression de huit gènes $a_1, \ldots, a_8$ est étudiée chez six patients $o_1, \ldots, o_6$ (ce tableau est donné séparément en

annexe A pour faciliter la lecture du mémoire). De manière plus générale, nous appellerons les gènes des *attributs* et les patients des *objets*.

Patients (Objets) \ Gènes (Attributs)	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
o_1	1	0	1	0	1	0	1	0
o_2	0	1	1	0	1	0	1	0
o_3	1	0	1	0	1	0	0	1
o_4	1	0	0	1	0	1	0	1
o_5	0	1	1	0	0	1	0	1
o_6	0	1	1	0	1	1	0	1

TAB. 1 – Un exemple $\mathcal{D}$ de données larges.

Dans la réalité, les données à fouiller ont des dimensions beaucoup plus imposantes. Notre démarche s'appuie sur un cas d'étude pratique que nous avons rencontré dans le cadre du projet BINGO (Bases de données INductives et GénOmique) [1] : les données SAGE (Serial Analysis of Gene Expression) qui sont composées de 90 situations biologiques décrites par l'expression de 27 679 gènes. Ces données sont obtenues expérimentalement par un protocole décrit dans [VZVK95]. Nous verrons que les biologistes sont fortement intéressés par des groupes de synexpression (des ensembles de gènes ayant des profils d'expression similaires) ou des règles permettant d'aboutir à une classification fine des différents types de cancer. Dans le domaine de la fouille de données, les ensembles de gènes sont appelés des motifs [2] et les règles des règles d'association. Par exemple, $a_1a_4a_6$ et a_3a_5 sont des motifs et $a_2a_3 \rightarrow a_6$ une règle d'association du jeu de données du tableau 1. La fréquence d'un motif X est égale à son nombre d'occurrences dans les données, elle sera notée $\mathcal{F}(X)$. La fréquence du motif a_3a_5 est égale à 4 puisqu'il apparaît dans 4 objets de $\mathcal{D}$ (o_1, o_2, o_3, et o_6).

L'extraction de motifs fréquents (c'est-à-dire de motifs dont la fréquence dépasse un certain seuil) - et plus généralement de motifs contraints - est une tâche algorithmiquement ardue car l'espace de recherche des motifs est gigantesque. En effet, celui-ci augmente exponentiellement avec le nombre d'attributs. Dans le cas des données SAGE, l'espace de recherche peut compter jusqu'à $2^{27\,679}$ motifs et il est vain de chercher à le parcourir naïvement. Actuellement, cette tâche est bien maîtrisée pour certaines contraintes dont la classique contrainte de fréquence. Malheureusement, si de nombreux progrès ont été faits dans le domaine de l'extraction sous contraintes [Sou06], ceux-ci restent insuffisants lorsque le nombre d'attributs est élevé notamment pour des contraintes comme la δ-liberté. Or, les motifs libres et δ-libres possèdent une propriété de minimalité qui les place au centre de nombreuses problématiques aussi bien en fouille de données (concision de l'information, extraction facilitée, diminution de la redondance dans les règles d'association, etc.) qu'en algorithmique (*vertex cover* pour les graphes ou traverses minimales pour les hypergraphes). En particulier, ce sont des motifs parfaitement adaptés à la génération de règles. En fouille de données, une règle est une assertion de type « si X est vrai alors Y l'est aussi ». La production de règles et notamment de règles d'association [AIS93] est une utilisation classique des motifs extraits à partir d'une base de données. Plus formellement, une règle d'association est une implication entre deux motifs de la forme $X \rightarrow Y$. X est appelé *prémisse* de la règle et Y sa

[1] Le projet BINGO (2004-2007) est réalisé dans le cadre de l'Action Concertée Incitative Masse de Données.
[2] Ce mémoire traite uniquement des motifs dits *ensemblistes* i.e. d'ensembles d'attributs non ordonnés.

conclusion. Malheureusement, la quantité de règles produites peut à nouveau être exponentielle en le nombre d'attributs et l'utilisateur est alors confronté à un nouveau problème : celui de la fouille de règles. Il existe des indicateurs de la qualité des règles comme le *support* et la *confiance* [AS94]. Le support est la proportion d'objets contenant à la fois la prémisse et la conclusion de la règle dans la base de données. La confiance est la proportion de ces mêmes objets par rapport aux objets contenant la prémisse de la règle. Considérons l'exemple du tableau 1. Le support de la règle $a_2a_3 \rightarrow a_6$ vaut $\frac{1}{3}$ puisque le motif $a_2a_3a_6$ apparaît dans 2 objets de la base de données sur 6. Sa confiance vaut $\frac{2}{3}$ car 3 objets contiennent la prémisse a_2a_3 mais seuls deux d'entre eux contiennent également a_6. Les limites du support et de la confiance sont bien connues [Gui00, Azé03], ce qui a conduit à la définition d'autres indices ou *mesures d'intérêt* pour mesurer la pertinence des règles d'association. Mais, paradoxalement, l'abondance et la diversité de ces mesures rendent le choix d'une mesure et son utilisation compliquée pour un non-spécialiste.

Contributions

Nous présentons maintenant brièvement les principales contributions de notre travail. En ce qui concerne la découverte de motifs, nous offrons une nouvelle vision de l'extraction de connaissances dans les jeux de données décrits par un grand nombre d'attributs. Nous préconisons l'utilisation de l'*extension* des motifs dans de tels contextes d'extraction. En effet, l'extension d'un motif est composée des objets qui contiennent ce motif. L'extension comporte peu d'objets dans les données larges et c'est alors un outil aisément manipulable. Cette démarche débouche notamment sur l'extraction des motifs δ-libres pour lesquels il n'existait pas de solution dans ce type de données.

La propriété de minimalité des motifs δ-libres peut être exploitée pour la construction de couvertures de règles d'association, ce qui permet de réduire sans perte d'information, le nombre de règles produites. Les mesures d'intérêt sont un autre moyen de diminuer le nombre de règles proposées à l'utilisateur. Le choix d'une mesure d'intérêt est difficile. C'est pourquoi nous proposons un cadre générique permettant de mieux comprendre le fonctionnement des mesures d'intérêt et les caractéristiques des règles sélectionnées. De plus, nous fournissons une méthode produisant un ensemble réduit de règles optimisant un grand nombre de mesures d'intérêt. Ces règles, appelées règles optimisées informatives, ont une prémisse minimale et une conclusion maximale. Elles sont construites à partir des motifs libres, ce qui permet notamment de les extraire efficacement. Dans le cas particulier des règles de classification, les règles optimisées informatives constituent une couverture des règles qui optimisent de nombreuses mesures.

Par ailleurs, nous donnons un nouvel éclairage sur l'extraction de motifs. Nous affirmons que des méthodes de fouille de données peuvent contribuer à la résolution de problèmes réputés « formels ». Il s'agit d'une démarche inverse de celles usuellement rencontrées où des outils théoriques comme les connexions de Galois, la structure de treillis ou les traverses minimales d'un hypergraphe sont utilisés pour mettre au point des algorithmes d'extraction de motifs. Plus précisément, nous montrons que notre méthode d'extraction des motifs δ-libres est applicable à d'autres contextes, en particulier nous l'adaptons à la résolution d'un problème sur les hypergraphes, le calcul des traverses minimales. Nous caractérisons les traverses minimales à l'aide de leur extension puis nous exploitons cette caractérisation dans une approche de type Apriori [AS94].

Enfin, les différentes méthodes proposées dans ce mémoire ont été testées dans divers contextes applicatifs. Notre approche concernant les motifs δ-libres a montré son intérêt pour la détermi-

nation de règles de caractérisation sur les données SAGE. Nous validons notre méthode de calcul des traverses minimales pour le calcul de bordures et pour la visualisation de clusters.

En résumé, notre travail est transverse aux domaines de la fouille de données et des hypergraphes. En fouille de données, nous nous intéressons à la fois aux méthodes d'extraction d'information (extraction de motifs δ-libres dans les données larges), à la qualité de l'information extraite (cadre théorique pour les mesures d'intérêt) ainsi qu'aux applications (analyse du transcriptome). La partie algorithmique s'intéresse au problème du calcul des traverses minimales dans un hypergraphe au moyen des algorithmes de fouille de données précédemment cités. Même si les contributions relèvent de plusieurs domaines, nous verrons que celles-ci sont liées aux propriétés d'éléments minimaux dans la structure de treillis.

Organisation du mémoire

La première partie de ce mémoire dresse un état de l'art sur l'extraction de motifs et l'évaluation de la qualité des règles issues de bases de données. Nous présentons plus particulièrement au chapitre 1 la problématique et les méthodes d'extraction de motifs dans les données comportant un grand nombre d'attributs. Les représentations condensées ainsi que la technique de transposition de données, qui apportent des solutions pour certains types de motifs, sont exposées. Nous définissons et soulignons l'intérêt des motifs libres (ou minimaux) et δ-libres tout en montrant qu'il n'existe pas de méthode permettant leur extraction à partir de données larges. Le chapitre 2 montre les difficultés posées par l'évaluation des règles et connaissances extraites et propose une synthèse sur les mesures d'intérêt. Nous discutons l'utilisation de couvertures de règles et de mesures d'intérêt portant sur les règles.

La deuxième partie présente l'ensemble de nos contributions sur l'extraction et l'usage des motifs minimaux. Nous proposons au chapitre 3 une méthode pour extraire les motifs δ-libres dans les larges jeux de données. Celle-ci repose sur l'utilisation de l'extension des motifs et d'un critère d'élagage original. Nous expliquons comment elle peut être étendue à la caractérisation de classes. Au chapitre 4, nous définissons un cadre formel générique pour les mesures d'intérêt. Nous montrons qu'un grand nombre de mesures d'intérêt ont des comportements similaires et nous donnons une méthode pour produire un ensemble réduit de règles informatives qui optimisent de nombreuses mesures d'intérêt, cet ensemble étant construit à partir de motifs libres. Le chapitre 5 propose une ouverture ou un élargissement de la propriété de minimalité à un autre domaine : celui des hypergraphes. Nous prouvons que la méthode d'extraction de motifs δ-libres présentée au chapitre 3 peut être exploitée pour le calcul des traverses minimales d'un hypergraphe. De plus, nos expériences montrent que notre approche est particulièrement efficace dans le cas des hypergraphes constitués d'hyperarêtes de grande taille, une situation pour laquelle il n'existe pas de solution satisfaisante.

La dernière partie de ce mémoire est consacrée aux applications de nos résultats. Dans le chapitre 6, nous présentons l'intérêt des motifs δ-libres et des règles de caractérisation dans le cas des données SAGE. Nous discutons également leur interprétation. Le chapitre 7 montre l'intérêt pratique de notre algorithme de calcul des traverses minimales pour les algorithmes en profondeur d'extraction de motifs fréquents et la construction de clusterings lors d'un processus de visualisation de données.

Le dernier chapitre établit un bilan de notre travail. Après avoir résumé et discuté les principaux résultats, nous donnons quelques perspectives de recherche.

Première partie

Découverte de motifs : état de l'art

Introduction

Cette partie d'état de l'art pointe les difficultés d'extraction de motifs dans les données larges et de sélection de règles pertinentes.

Dans le chapitre 1, le vocabulaire usuel de fouille de données est précisé puis nous rappelons le principe général des algorithmes d'extraction de motifs les plus connus. Ensuite, nous nous concentrons sur les méthodes les plus appropriées dans le contexte des données larges : les représentations condensées de motifs fréquents (plus précisément, les motifs libres et les motifs fermés) et la transposition de données. Ces méthodes utilisent largement la notion de connexion de Galois pour exploiter efficacement la structure de treillis ; c'est pourquoi nous donnons dans les préliminaires la définition et les propriétés d'une telle connexion. Le chapitre 1 introduit notamment les concepts de motifs libres et δ-libres qui sont au cœur de notre travail.

Nous avons vu au chapitre d'introduction qu'une utilisation classique des motifs est la production de règles d'association. De même que les motifs, les règles produites sont très nombreuses. Diverses solutions ont été mises en œuvre pour pallier le problème de la sélection des règles d'association les plus pertinentes. Le chapitre 2 indique les principales approches de sélection qui ne font pas appel à des connaissances expertes. Il détaille deux d'entre elles : les couvertures de règles et les mesures d'intérêt. La première de ces deux méthodes vise à éliminer les règles redondantes i.e. celles qui n'apportent aucune information supplémentaire par rapport à l'ensemble des règles extraites. La deuxième méthode consiste à définir des critères indépendants du domaine d'expertise, appelés mesures d'intérêt objectives, qui permettent de sélectionner les règles les plus pertinentes. Par ailleurs, nous mettons l'accent sur les règles de classification qui constituent un cas particulier de règles très étudié et pour lequel nous obtenons des résultats spécifiques (cf. chapitre 4).

Chapitre 1

Extraction de motifs dans les données comportant un grand nombre d'attributs

Sommaire

Dans ce chapitre, nous décrivons les techniques qui ont été mises en œuvre pour pallier les difficultés d'extraction rencontrées lors de l'exploration de données larges. Dans la section 1.1, nous rappelons quelques définitions inhérentes à la fouille de données orientée motifs [MT97], ainsi que les principales propriétés d'une connexion de Galois. Puis nous évoquons deux approches appropriées au contexte des bases de données larges : l'utilisation de représentations condensées (cf. section 1.2) et la méthode de transposition de données (cf. section 1.3).

1.1 Préliminaires

Nous commençons par rappeler quelques termes issus de la fouille de données et les principes généraux de l'extraction de motifs. Ensuite nous soulignons le lien entre le treillis qui représente une base de données et les connexions de Galois, puis nous donnons les principales propriétés d'une connexion de Galois.

1.1.1 Extraction de motifs

Espace de recherche

La définition 1 donne la définition formelle d'une base de données.

Définition 1 (Contexte formel) *Une base de données $\mathcal{D}$ est décrite par un triplet $(\mathcal{A}, R, \mathcal{O})$ où $\mathcal{A}$ est l'ensemble des attributs et $\mathcal{O}$ est l'ensemble des objets. R est une relation binaire définie sur $(\mathcal{A}, \mathcal{O})$ par « un attribut a est en relation avec un objet o si et seulement si o vérifie la propriété notée a ». On dira dans ce cas que o contient ou supporte a.*

Un contexte formel peut simplement être vu comme une matrice binaire. Un exemple de contexte formel comportant huit attributs et six objets a été donné dans le tableau 1 en introduction (voir page 4). La définition 2 rappelle ce qu'est un motif.

Définition 2 (Motif) *Un motif d'attributs est une partie de $\mathcal{A}$. Un motif d'objets est une partie de $\mathcal{O}$.*

Lorsqu'il n'y aura pas d'ambiguïté, nous désignerons par motif un motif d'attributs. L'espace de recherche des motifs est défini comme le langage $\mathcal{L}_\mathcal{A}$ construit sur l'alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ constitué de tous les attributs de $\mathcal{D}$. C'est aussi l'ensemble des parties de $\mathcal{A}$. Cet espace est généralement représenté sous forme d'un treillis comme sur la figure 1.1. L'inclusion entre deux motifs définit une relation de *spécialisation* [Mit82]. Ainsi, si un motif X est inclus dans le motif Y, on dit que Y est plus spécifique que X ou que X est plus général que Y.

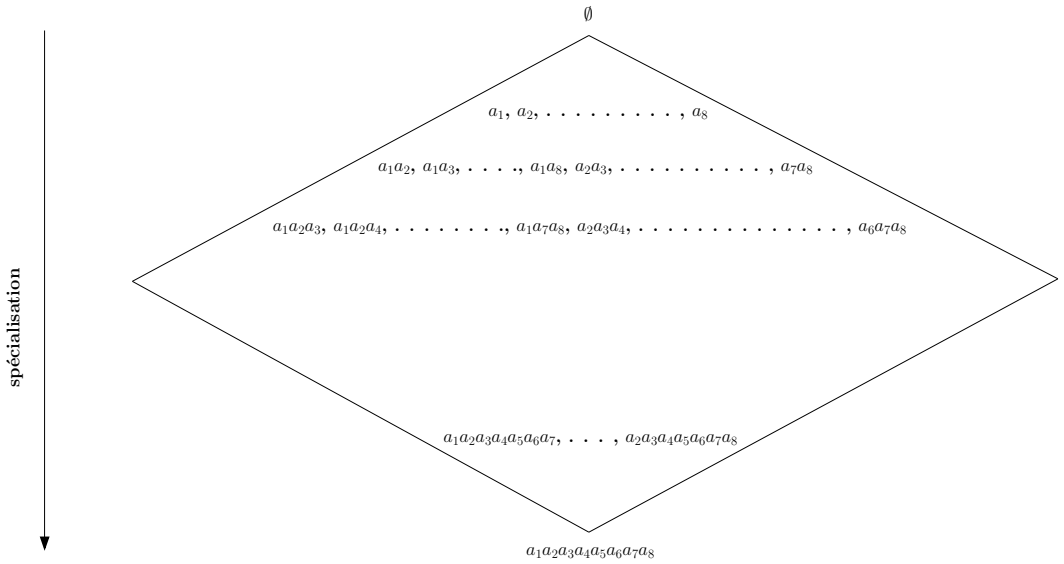

FIG. 1.1 – Treillis représentant le langage $\mathcal{L}_\mathcal{A}$ avec $\mathcal{A} = \{a_1, \ldots, a_8\}$.

L'ensemble des motifs du langage $\mathcal{L}_\mathcal{A}$ qui vérifient une certaine propriété ou *contrainte* q dans la base de données $\mathcal{D}$, est appelé *théorie* [MT97] et est noté $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$. Certaines contraintes vérifient des propriétés qui facilitent le calcul de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$ comme les contraintes anti-monotones [MT97] :

Définition 3 *La contrainte q est anti-monotone si et seulement si pour tout motif d'attributs X, on a :*

$$\forall Y \subset X, \quad q(X) \Rightarrow q(Y).$$

La contrainte de fréquence - qui consiste à sélectionner les motifs dont le nombre d'occurrences dans $\mathcal{D}$ dépasse un seuil donné - est anti-monotone. Pour toute contrainte q anti-monotone, les motifs maximaux de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$ définissent une frontière entre les motifs vérifiant q et ceux qui ne la vérifient pas. On définit ainsi les *bordures* de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$:

Définition 4 *L'ensemble des motifs maximaux (au sens de l'inclusion) qui satisfont la contrainte q dans $\mathcal{D}$ est noté $Bd^+(Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q))$ et constitue la bordure positive de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$. De manière duale, l'ensemble des motifs minimaux qui ne satisfont pas la contrainte q dans $\mathcal{D}$ est noté $Bd^-(Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q))$ et constitue la bordure négative de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$.*

La réunion de la bordure positive et de la bordure négative forment la *bordure* d'une théorie. Remarquons que la recherche de motifs se restreint souvent aux motifs présents. La figure 1.2 représente le treillis des motifs contenus dans la base de données de l'exemple 1. Nous indiquons entre parenthèses sous chaque motif, sa fréquence (la définition de la fréquence d'un motif est donnée à la page 4). En considérant la contrainte q « avoir une fréquence au moins égale à 2 », nous avons dessiné la frontière entre les motifs vérifiant q et ceux qui ne la vérifient pas. Les motifs de la bordure positive de $Th(\mathcal{L}_\mathcal{A}, \mathcal{D}, q)$ sont entourés en bleu (avec des tirets) et ceux de sa bordure négative en rouge.

Algorithmes

L'algorithme d'extraction de motifs fréquents le plus connu est certainement Apriori [AS94]. Son principe est de parcourir en largeur l'espace de recherche en commençant par les attributs puis en visitant les paires d'attributs puis les triplets, etc. Cela revient à parcourir le treillis de la figure 1.1. L'algorithme Apriori étant une instance de l'algorithme générique `Guess & Correct` [MT97], chaque itération se décompose en deux étapes : tout d'abord la *génération* des candidats à une profondeur donnée et ensuite la *vérification* qui consiste à tester si les candidats générés sont fréquents en passant sur la base de données. La génération de candidats de longueur k (la fonction nommée `apriori-gen` dans [AS94]) revient à effectuer l'union de deux motifs fréquents de longueur $k-1$ qui possèdent un préfixe commun de longueur $k-2$. Sur l'exemple de la figure 1.2, les motifs 2-fréquents $a_2a_3a_6$ et $a_2a_3a_8$ qui ont en commun le préfixe a_2a_3 sont fusionnés pour donner le candidat $a_2a_3a_6a_8$ à la profondeur 4.

Apriori exploite largement l'anti-monotonie de la contrainte de fréquence puisqu'il y est fait usage de deux critères d'élagages donnés par les propriétés 1 et 2.

Propriété 1 *Si un motif n'est pas fréquent, il est inutile de tester ses spécialisations dont la fréquence est nécessairement moins élevée.*

Propriété 2 *Un motif dont l'une des généralisations n'est pas fréquente ne peut être fréquent lui-même et ne doit pas être testé.*

Toujours sur le même exemple, les spécialisations de a_4 ne seront pas considérées puisque la fréquence de a_4 est inférieure à 2. De plus, $a_2a_5a_6$ est généré car a_2a_5 et a_2a_6 sont 2-fréquents mais éliminé puisque a_5a_6 ne l'est pas. Nous verrons que le principe de génération de candidats et les critères d'élagages d'Apriori seront largement utilisés dans la suite de ce mémoire (cf. chapitres 3 et 5). Toutefois, la structure en deux phases ne sera pas conservée puisque dans les algorithmes que nous proposerons, la génération et la vérification d'un motif seront presque simultanées. Par ailleurs, nous ajouterons un critère d'élagage fin aux deux critères cités plus haut.

Certains algorithmes [GMS97, Zak00b, SU03] permettent d'extraire directement la bordure positive des motifs fréquents en effectuant un parcours en profondeur de l'espace de recherche. Un motif fréquent est spécialisé jusqu'à ce que plus aucune de ses spécialisations ne soit fréquente. On dispose alors d'un motif appartenant à la bordure positive des fréquents. Zaki propose de réitérer ce procédé pour tous les attributs fréquents dans l'algorithme Eclat [Zak00b]. Au contraire,

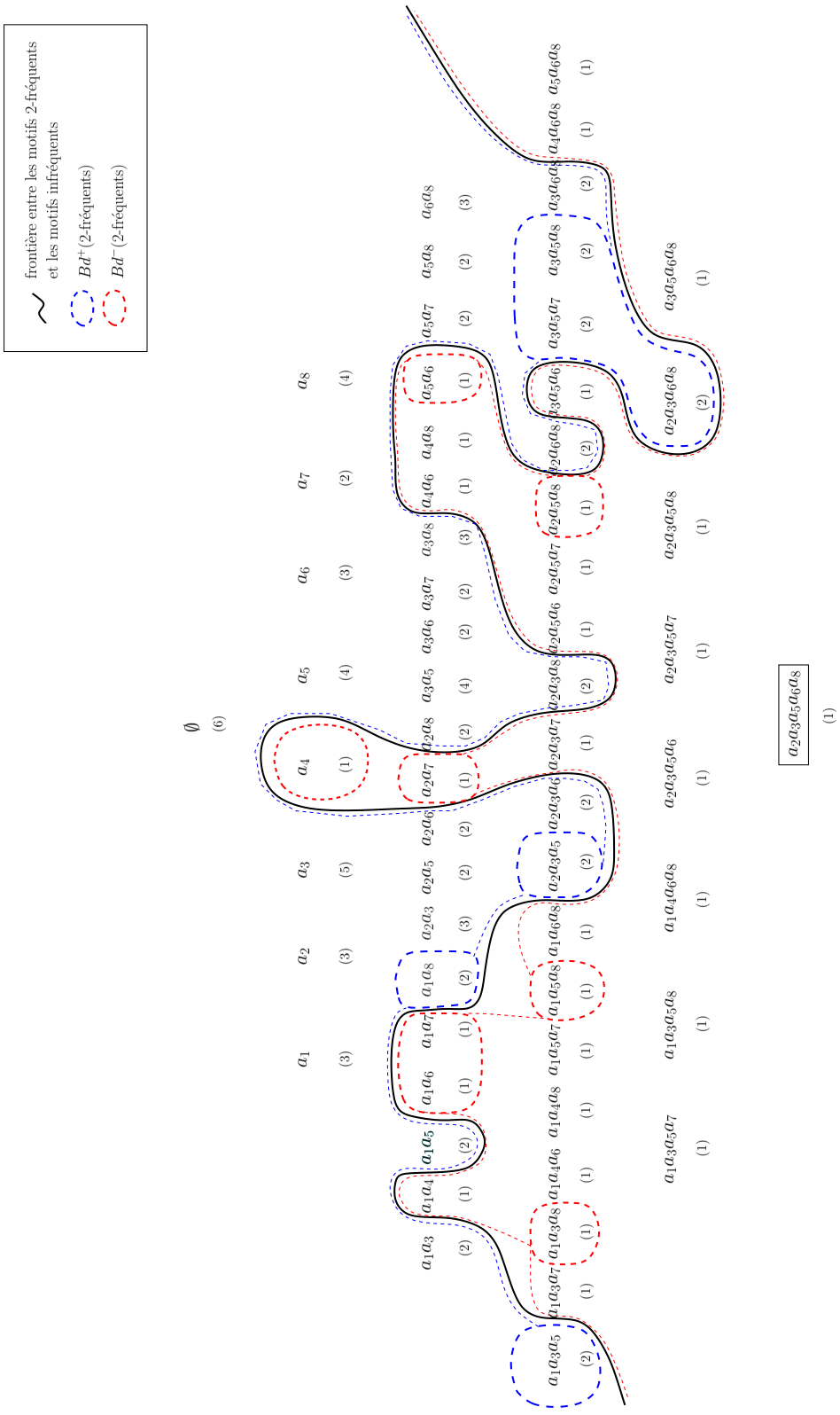

FIG. 1.2 – Bordures de l'ensemble des motifs 2-fréquents dans la base de données du tableau 1.

dans [GMS97, SU03], on teste si la bordure positive est déterminée en totalité pour chaque motif maximal fréquent découvert. Ce test s'appuie sur la notion d'hypergraphe [Ber89] et plus particulièrement, de traverse minimale d'un hypergraphe. Ces notions sont définies au chapitre 5 et nous y présentons aussi nos contributions dans ce domaine.

Les algorithmes que nous venons de citer ne tirent pas pleinement parti de la structure de treillis qui représente l'espace de recherche. Dans le treillis de la figure 1.2, de nombreux motifs partagent la même fréquence. Cela provient du fait que ces motifs apparaissent exactement dans les mêmes objets de $\mathcal{D}$. Par exemple, les motifs a_7, a_3a_7, a_5a_7 et $a_3a_5a_7$ ont tous une fréquence égale à 2 car ils apparaissent exactement dans les objets o_1 et o_2. De telles similitudes peuvent être exploitées de manière très fine pour résumer l'information contenue dans une base de données et éviter une recherche exhaustive i.e. éviter de parcourir tous les motifs du treillis. Les représentations condensées, que nous présenterons dans la section 1.2, sont basées sur ces observations. De manière plus formelle, c'est un opérateur de fermeture sur les motifs défini à partir d'une connexion de Galois qui permet de construire des classes d'équivalence de fréquence c'est-à-dire de regrouper des motifs qui ont la même fréquence. C'est pourquoi la section suivante est consacrée à la définition des connexions de Galois.

1.1.2 Un outil de structuration du treillis : la connexion de Galois

D'une manière plus générale, une connexion de Galois permet d'associer deux ensembles. Il est alors également possible de structurer le treillis à l'aide de classes d'équivalence comme nous le verrons plus loin. Dans le cas particulier de l'ECBD, ce sont les motifs d'attributs et les motifs d'objets d'une base de données qui sont reliés par une connexion de Galois particulière (voir la définition 6 au paragraphe suivant). À la section 1.2, il est expliqué comment les classes d'équivalence sont exploitées par les représentations condensées pour couvrir efficacement tout le treillis. La section 1.3 montre comment la méthode de transposition tire profit du lien induit par la connexion et apporte une solution partielle au problème de la découverte de motifs dans les données larges.

Définitions

Une connexion de Galois [Bir48, page 56] permet de relier deux ensembles partiellement ordonnés[3]. Soient $(A, \leq_1)$ et $(B, \leq_2)$ deux ensembles partiellement ordonnés et 2^A et 2^B les ensembles des parties de A et de B.

Définition 5 (Connexion de Galois) *Une connexion de Galois entre deux ensembles partiellement ordonnés $(A, \leq_1)$ et $(B, \leq_2)$ est composée de deux applications $f : A \to B$ et $g : B \to A$ qui sont décroissantes et telles que $f \circ g$ et $g \circ f$ sont extensives i.e. elles vérifient $\forall X \in B,\ X \leq_2 f \circ g(X)$ et $\forall Y \in A,\ Y \leq_1 g \circ f(Y)$.*

Nous notons (f, g) une connexion de Galois. Les fonctions f et g sont appelées *opérateurs de Galois*. En fouille de données, on considère l'ensemble des motifs d'attributs $2^{\mathcal{A}}$ et l'ensemble des motifs d'objets $2^{\mathcal{O}}$. Ceux-ci sont ordonnés par inclusion, ce qui induit une relation d'ordre partiel [Bir48, page 2]. On définit alors une connexion de Galois notée $(f_{\mathcal{D}}, g_{\mathcal{D}})$ entre $(2^{\mathcal{A}}, \subseteq)$ et $(2^{\mathcal{O}}, \subseteq)$ de la manière suivante :

[3] Un ensemble partiellement ordonné est un ensemble sur lequel est définie une relation binaire réflexive, antisymétrique et transitive.

Définition 6 (Intension et extension)

$$\forall O \subseteq \mathcal{O},\ f_{\mathcal{D}}(O) = \{a \in \mathcal{A} | \forall o \in O, aRo\}$$

$$\forall A \subseteq \mathcal{A},\ g_{\mathcal{D}}(A) = \{o \in \mathcal{O} | \forall a \in A, aRo\}$$

$f_{\mathcal{D}}$ est alors appelée *intension* et $g_{\mathcal{D}}$ prend le nom d'*extension*. Intuitivement, l'intension d'un motif d'objets correspond aux attributs communs à chacun des objets composant ce motif ; l'extension d'un motif d'attributs est composée des objets qui contiennent chacun des attributs du motif. Dans l'exemple du tableau 1, l'extension du motif d'attributs a_3a_7 est o_1o_2 et l'intension du motif d'objets o_3o_4 est a_1a_8.

Propriétés

Les applications $h = f \circ g$ et $h' = g \circ f$ sont des opérateurs de fermeture. Elles vérifient les propriétés d'extensivité ($X \leq_2 h(X)$) ; d'idempotence ($h(h(X)) = h(X)$) et d'isotonie ($X_1 \leq_2 X_2 \Rightarrow h(X_1) \leq_2 h(X_2)$). Un élément de A ou de B est dit *fermé* s'il est égal à sa propre image par h' ou par h. La *fermeture* d'un élément X de B est son image par h. Un motif d'attributs X vérifiant $X = h(X)$ est appelé *motif fermé*. La fermeture d'un motif d'attributs est égale au plus petit motif fermé qui le contient. La fermeture d'un motif X peut aussi être vue comme le plus grand motif commun à tous les objets contenant X. Reprenons l'exemple du tableau 1. Le motif a_3a_7 n'est pas fermé car les objets de son extension o_1 et o_2 ont tous deux en commun l'attribut a_5. Cela signifie que l'attribut a_5 est toujours présent avec le motif a_3a_7. La fermeture de a_3a_7 est donc $a_3a_5a_7$. Sur la figure 1.3, tous les motifs fermés de $\mathcal{D}$ (il y en a 18) ont été encadrés.

La propriété suivante [DW02, page 310] montre comment l'on peut calculer l'extension de la réunion de plusieurs ensembles en intersectant les extensions de chacun de ces ensembles.

Propriété 3 *Soit $(X_1, \ldots, X_n)$ une famille d'éléments de 2^B. On a l'égalité suivante :*

$$g(\bigcup_{i \in \{1 \ldots n\}} X_i) = \bigcap_{i \in \{1 \ldots n\}} g(X_i).$$

Nous utilisons cette propriété dans le chapitre 3 pour calculer l'extension d'un motif candidat à partir des extensions de ses générateurs dans un algorithme par niveaux. Remarquons que la propriété 3 est vraie pour n'importe quelle relation d'ordre, ce qui nous permettra d'en faire également usage au chapitre 5. Nous montrons maintenant comment l'extension permet de structurer le treillis en classes d'équivalence.

Structuration en classes d'équivalence

Le fait d'avoir la même image par g constitue une relation d'équivalence sur B : pour tous $Y, Y' \in B,\ Y \sim Y'$ si et seulement si $g(Y) = g(Y')$. On peut alors en déduire une structuration du treillis sous forme de classes d'équivalence comme indiqué dans la définition 7 :

Définition 7 (Classe d'équivalence) *La classe d'équivalence $\mathcal{R}_g(X)$ d'un élément X de B est définie comme suit :*

$$\mathcal{R}_g(X) = \{X' \in B | g(X') = g(X)\}.$$

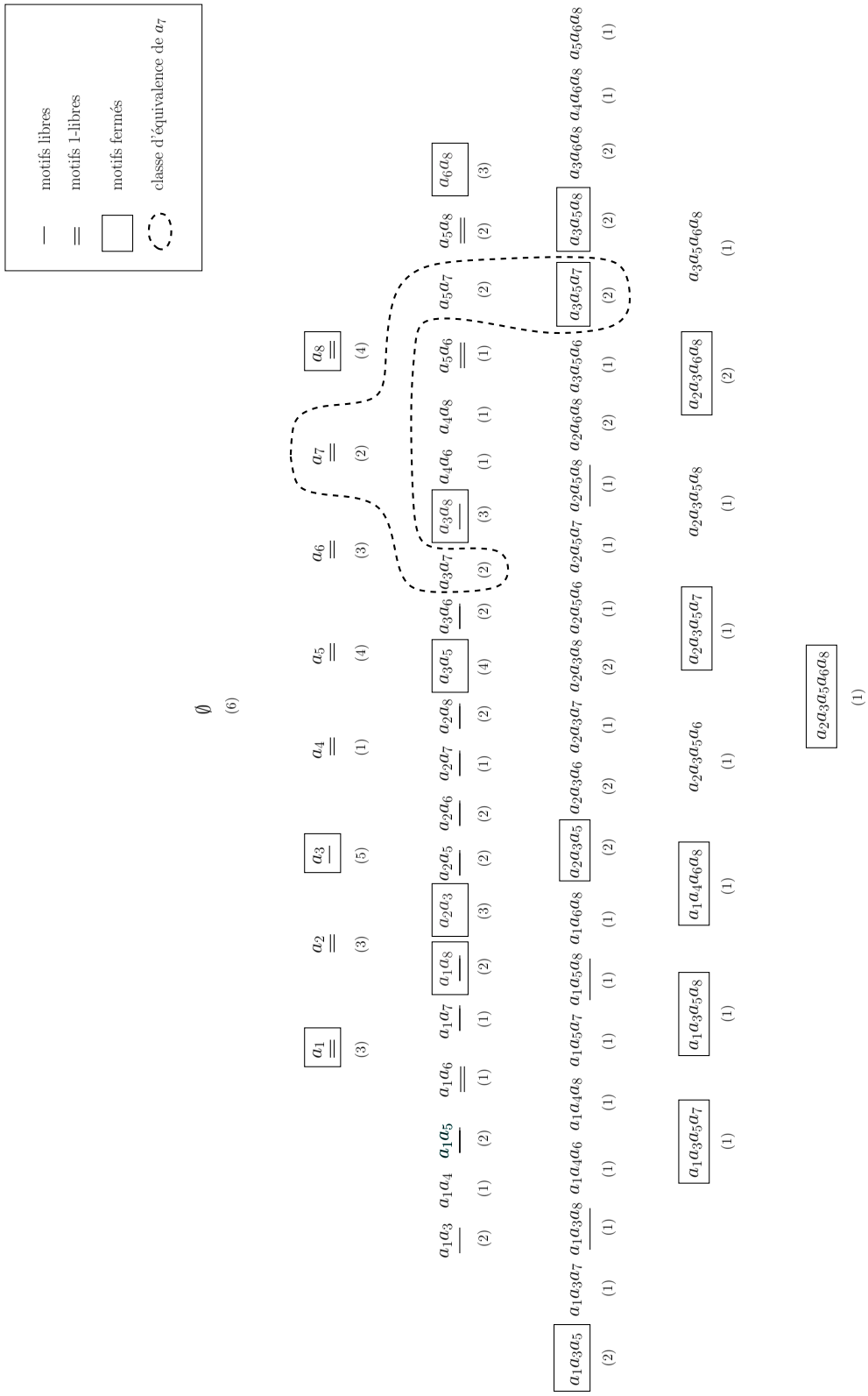

FIG. 1.3 – Treillis des motifs présents dans la base de données du tableau 1.

Remarquons que les éléments d'une classe d'équivalence ont tous la même fermeture. La fermeture de tout élément d'une classe d'équivalence donnée appartient à cette classe d'équivalence et contient tous ses éléments : c'est le plus grand élément relativement à $\leq_2$ dans cette classe d'équivalence.

Propriété 4 *Soit X un élément de B. $h(X)$ est le plus grand élément de $\mathcal{R}_g(X)$.*

Dans le domaine de la fouille de données, les classes ainsi définies sont habituellement appelées classes d'équivalence de fréquence car les motifs appartenant à une même classe d'équivalence ont la même fréquence. Ceci peut prêter à confusion puisque le fait de posséder la même fréquence n'est qu'une conséquence de cette structuration. Considérons à nouveau l'exemple du tableau 1. Les motifs a_7, a_3a_7, a_5a_7 et $a_3a_5a_7$ ont tous la même extension o_1o_2. Ils forment donc une classe d'équivalence du treillis comme l'indique le trait noir en pointillés sur la figure 1.3. On peut vérifier facilement qu'ils ont également la même fréquence qui est égale à 2. Le fait que ces motifs partagent la même fréquence n'est qu'une particularité de la connexion $(f_{\mathcal{D}}, g_{\mathcal{D}})$ utilisée en ECBD et dépend de la définition de la fréquence. Par contre, les motifs $a_2a_3a_7$ et $a_3a_5a_7$ ont tous deux une fréquence égale à 2 mais ne sont pas dans la même classe d'équivalence.

Cette partition du treillis en classes d'équivalence revêt une importance particulière pour la découverte de motifs fréquents. Puisque les motifs d'une classe d'équivalence partagent la même fréquence, il est possible de ne considérer que les motifs minimaux ou le motif maximal de chaque classe d'équivalence au lieu de parcourir tout l'espace de recherche. Dans l'exemple que nous venons de citer, seul a_7 (qui est minimal) ou $a_3a_5a_7$ (qui est maximal) sera considéré. C'est sur ce principe que sont basées les représentations condensées décrites dans la section suivante.

1.2 Les représentations condensées de motifs fréquents

1.2.1 Intuitions

Le partage du treillis selon les classes d'équivalence vues précédemment offre la possibilité de résumer les informations de fréquence contenues dans une base de données. Ce résumé est constitué d'un nombre réduit de motifs bien choisis qui forment une *représentation condensée* des motifs fréquents. Plus précisément, l'idée est de déduire la fréquence de n'importe quel motif à partir d'un ou de plusieurs représentants qui appartiennent à la même classe d'équivalence. Ces représentants sont soit les motifs maximaux soit les motifs minimaux des classes d'équivalence du treillis. La notion de maximalité dans une classe d'équivalence a déjà été définie à la section précédente sous le terme de fermé. Nous nous concentrons à présent sur la notion de minimalité.

1.2.2 Liberté

Les termes de motif *libre* [BBR00, BBR03] ou de motif *clé* [PBTL99b] sont souvent utilisés pour désigner la minimalité dans une classe d'équivalence.

Définition 8 (Liberté 1) *Un motif X est libre si et seulement si il est minimal au sens de l'inclusion dans $\mathcal{R}_g(X)$.*

Dans le treillis représenté sur la figure 1.3, les 24 motifs libres de $\mathcal{D}$ ont été soulignés. Les motifs libres définissent une frontière avec les motifs de fréquence supérieure (leurs généralisations). Ce constat donne lieu à une autre définition de la liberté qui sera aussi utilisée dans la suite de ce document.

Définition 9 (Liberté 2) *Un motif X est libre si et seulement si pour tout motif Y inclus strictement dans X, $\mathcal{F}(Y) > \mathcal{F}(X)$.*

Dans la suite de ce document, nous choisirons la définition la plus appropriée en fonction du contexte. La définition 9 reste vraie si on la restreint aux sous-ensembles Y de X de longueur $X - 1$ au lieu de considérer tous les sous-ensembles propres de X. Elle souligne le fait que la fréquence d'un motif libre est « déconnectée » de celle de ses sous-ensembles. Cela signifie que les motifs libres ne possèdent pas de corrélation intrinsèque et nous verrons dans le chapitre suivant qu'ils ont de bonnes propriétés pour construire des règles d'association. De plus, ce sont des motifs relativement faciles à extraire puisque la liberté est une contrainte anti-monotone par rapport à la spécialisation des attributs [BBR00, BBR03].

Les motifs libres permettent à eux seuls de déterminer la fréquence de chaque motif d'une base de données. Il suffit pour un motif donné de retrouver le plus grand libre qu'il contient, ces deux motifs ayant la même fréquence. Bastide *et al.* utilisent cette propriété dans leur algorithme d'extraction de motifs fréquents nommé Pascal [BTP+02] où le fait de déduire la fréquence d'un motif quelconque à partir de celle d'un motif libre est appelé « comptage par inférence ». Les motifs libres sont une représentation condensée exacte des motifs fréquents : il est possible de retrouver la fréquence **exacte** de n'importe quel motif de la base de données. Lorsque l'on dispose des motifs libres fréquents et non pas de tous les motifs libres, une information supplémentaire (telle que la bordure négative des fréquents par exemple) est nécessaire pour déterminer la totalité des motifs fréquents. Il existe d'autres représentations condensées exactes des motifs fréquents ([CRB04] en donne un large panorama) telles que les motifs fermés [PBTL99a], les *disjunction-free generators* [BR01], les *motifs non dérivables* [CG02], les *motifs k-libres* [CG03] ou encore les *motifs essentiels* [CCL05]. Les efforts algorithmiques ont surtout été concentrés sur les motifs fermés pour lesquels il existe de nombreux algorithmes ; citons Close [PBTL99a], Charm [ZH02] ou encore Closet [PHM00].

Lors d'un parcours par niveaux, on réalise un gain de temps substantiel en évitant de générer tous les motifs d'une classe d'équivalence dès lors qu'on a déterminé les motifs libres de cette classe. Le nombre de motifs à considérer et à stocker en mémoire diminue drastiquement. Les motifs fermés constituent également une représentation condensée exacte des motifs fréquents. La fréquence d'un motif est déterminée à partir du fermé de sa classe d'équivalence. La propriété 4 montre que le motif maximal d'une classe d'équivalence est l'unique fermé qu'elle contient. Remarquons que ce résultat d'unicité est faux pour les libres, une classe d'équivalence peut posséder plusieurs motifs minimaux. En fixant le seuil de fréquence noté γ à 2 dans le jeu de données du tableau 1, on dénombre 13 classes d'équivalence. Ces données contiennent donc 13 motifs fermés 2-fréquents (voir propriété 4) que l'on peut retrouver sur la figure 1.3. Elles contiennent 16 motifs libres 2-fréquents. Or le nombre de motifs qui sont simplement 2-fréquents est de 31. Même pour cet exemple de dimension réduite, on réalise une économie de moitié sur le nombre de motifs à extraire en utilisant des représentations condensées.

1.2.3 Généralisation de la liberté : les motifs δ-libres

Nous verrons dans la section 2.2 du chapitre suivant qu'il est intéressant de considérer des règles d'association (voir la page 4 pour une définition) qui admettent un nombre borné d'exceptions dans des contextes réels. Les prémisses de ces règles sont constituées d'un motif dit δ-libre. Les motifs δ-libres sont une généralisation des motifs libres (un motif libre est un motif δ-libre avec $\delta = 0$). Ils constituent une représentation condensée approximée des motifs fréquents car ils permettent de restituer la fréquence de n'importe quel motif d'une base de données avec

une erreur bornée. Ils sont de plus relativement faciles à extraire. Les motifs δ-libres sont définis comme suit.

Définition 10 (Motif δ-libre) *Soit δ un entier strictement positif. Un motif X est δ-libre si et seulement si pour tout motif Y inclus strictement dans X, $\mathcal{F}(Y) > \mathcal{F}(X) + \delta$.*

Si δ est supérieur ou égal à 1, un motif δ-libre est nécessairement un motif δ'-libre avec $\delta' < \delta$. Un attribut a est δ-libre si et seulement si $\mathcal{F}(\emptyset) > \mathcal{F}(a) + \delta$ i.e. $|\mathcal{O}| > \mathcal{F}(a) + \delta$. Dans l'exemple du tableau 1, le motif a_5a_8 est 1-libre car $\mathcal{F}(a_5a_8) = 2$ et on a donc $\mathcal{F}(a_5a_8) + 1 < \mathcal{F}(a_5) = 4$ et $\mathcal{F}(a_5a_8) + 1 < \mathcal{F}(a_8) = 4$. Le motif a_5a_8 est aussi 0-libre car $\mathcal{F}(a_5a_8) + 1 < \mathcal{F}(a_5)$ et $\mathcal{F}(a_5a_8) + 1 < \mathcal{F}(a_8)$ implique que $\mathcal{F}(a_5a_8) < \mathcal{F}(a_5)$ et $\mathcal{F}(a_5a_8) < \mathcal{F}(a_8)$. Sur la figure 1.3, les motifs 1-libres, au nombre de 10, sont soulignés deux fois. Rappelons qu'on y compte 18 fermés et 24 libres. Même sur cet exemple élémentaire, il y a beaucoup moins de motifs 1-libres que de motifs fermés ou de motifs libres. Plus généralement, les motifs δ-libres sont des motifs assez courts et peu nombreux, même comparés aux motifs fermés. De plus, la δ-liberté étant une contrainte anti-monotone, une simple approche par niveaux se révèle suffisamment efficace pour extraire les représentations condensées basées sur des motifs δ-libres.

Nous verrons au chapitre 3 que la notion de presque-fermeture [BB00] est utile pour construire des règles de caractérisation de classes. La définition 11 précise cette notion.

Définition 11 (Presque-fermeture) *Soit X un motif. La presque-fermeture de X, notée $\mathcal{AC}(X)$ [4], est égale à l'ensemble des attributs contenus dans les mêmes objets que X à δ exceptions près.*

La presque-fermeture permet d'approximer la fréquence des motifs non δ-libres. Par exemple, sur le treillis représenté sur la figure 1.3, le motif a_5a_8 qui est 1-libre a pour presque-fermeture $a_1a_2a_3a_6$. Comme la fréquence de a_5a_8 vaut 2, celles de $a_1a_5a_8$ et de $a_3a_5a_8$ sont égales à 1 ou à 2. En réalité, la fréquence de $a_1a_5a_8$ vaut 1 et celle de $a_3a_5a_8$ vaut 2.

Le seul algorithme d'extraction des motifs δ-libres est MinEx [BBR00, BBR03], et son fonctionnement repose sur le calcul de presque-fermetures. C'est une instance de type Apriori où un test de δ-liberté est ajouté au test de fréquence. À notre connaissance, MinEx n'a donné lieu qu'à deux implémentations : ACMiner [BBR00, BBR03] par Artur Bykowski (LIRIS) et MVMiner [RC03] par François Rioult (GREYC) dont l'utilisation est discutée à la section 3.4.1 (page 43).

1.2.4 Bilan

Les représentations condensées permettent de résumer la fréquence de tous les motifs d'une base de données. Le traitement de données de volume important peut être rendu possible grâce à l'utilisation de ces dernières. Cependant, dans le cas de certaines données larges, la taille de l'espace de recherche et la longueur des motifs à manipuler restent des difficultés insurmontables. C'est pourquoi dans la section 1.3, nous exposons brièvement une méthode qui consiste à transposer les bases de données larges de manière à réaliser l'extraction sur la dimension la plus « faible » de la base. Cette méthode prend appui sur les motifs fermés et les connexions de Galois.

Les représentations condensées ne sont pas seulement à l'origine de méthodes efficaces de découverte de motifs fréquents. Dans le chapitre suivant, nous présentons un usage des motifs

[4] $\mathcal{AC}$ est l'acronyme de « almost-closure », traduction de presque-fermeture en anglais.

δ-libres : les règles de caractérisation δ-fortes qui peuvent être inférées à partir des motifs δ-libres. Plus généralement, nous définirons la notion de couverture d'un ensemble de règles d'association basée sur des représentations condensées.

1.3 Transposer les données larges pour en extraire des motifs

Dans le cas de contraintes particulières telles que la fréquence, il existe une solution pour la découverte de motifs vérifiant ces contraintes dans les données larges [RBCB03]. Cette approche exploite à la fois l'idée de transposer les données et les propriétés de la connexion de Galois de la définition 6. Cette section résume cette méthode dite de transposition [JR04].

Idées clés Si les données larges comportent un grand nombre d'attributs, leur nombre d'objets est en revanche assez peu important. La méthode de transposition des données s'appuie sur cette spécificité. Transposer les données permet d'exploiter ce déséquilibre en travaillant sur la « petite » dimension des données - les objets. Dans le cas des données larges, les motifs d'objets sont courts donc faciles à manipuler et à stocker.

L'idée fondamentale est d'appliquer de manière classique l'algorithme de son choix à la matrice de données transposée, puisque ses dimensions sont plus appropriées aux techniques de fouille, et d'en déduire les informations initialement recherchées. Dans l'exemple du tableau 1, l'espace de recherche des attributs est composé de 61 motifs (voir le treillis de la figure 1.3) alors que l'espace de recherche côté objets n'en contient que 43 ! Cependant, pour pouvoir exploiter les résultats de cette extraction, il faut être en mesure de retrouver les informations sur les motifs d'attributs à partir des motifs d'objets extraits.

C'est l'extension de la connexion de Galois donnée dans la définition 6 à la section 1.1.2 qui permet d'associer un motif d'objets à chaque motif d'attributs d'une base de données et c'est l'intension qui associe un motif d'attributs à chaque motif d'objets. Les motifs fermés bénéficient d'un lien privilégié résumé dans la définition d'un concept [Wil82, Gan84] : un *concept* associe de manière unique un fermé d'attributs à un fermé d'objets (son extension) et réciproquement. On peut passer d'un fermé à l'autre via les opérateurs de Galois. Reprenons l'exemple du tableau 1. La paire $(a_2a_3a_5, o_2o_6)$ forme un concept puisque $g_{\mathcal{D}}(a_2a_3a_5) = o_2o_6$. De plus, on a nécessairement $f_{\mathcal{D}}(o_2o_6) = a_2a_3a_5$.

Relaxation de contraintes Il n'est cependant pas possible d'appliquer la technique de transposition pour n'importe quelle contrainte. Dans le cas particulier des fermés, la connexion de Galois permet de déterminer facilement le fermé d'attributs associé au fermé d'objets obtenu par transposition, comme nous l'avons déjà remarqué. Puisqu'un concept (A, O) appartient à une base de données si et seulement si le concept (O, A) appartient à la base de données transposée [Wil82], la contrainte « être fermé » pour les motifs d'attributs peut être facilement déduite de la contrainte « être fermé » pour les motifs d'objets.

Dans [JR04], les contraintes portant sur des motifs d'attributs que l'on peut aisément traduire en termes d'objets sont listées. Malheureusement, une telle correspondance n'existe pas toujours. Pour les contraintes qui ne sont pas vérifiées par tous les éléments d'une même classe d'équivalence (et pour lesquelles il ne suffit pas d'examiner le fermé de la classe d'équivalence), il est impossible d'établir une forme « transposée » de la contrainte. Une autre solution est de relaxer la contrainte q en extrayant tous les fermés issus d'une classe d'équivalence qui contient potentiellement un motif vérifiant q. Il est ensuite nécessaire de regénérer entièrement certaines classes d'équivalence pour achever le processus et accéder à l'ensemble des motifs contraints. Cependant, cette approche

est inopérante dans le cas des motifs δ-libres. L'examen du fermé d'une classe d'équivalence ne permet pas de déterminer si cette classe contient des motifs δ-libres ou pas et il faudrait regénérer tout le treillis pour déterminer les motifs δ-libres. Il est donc impossible d'appliquer la méthode de transposition des données à la recherche de motifs δ-libres.

1.4 Conclusion

À notre connaissance, il n'existe pas de méthode permettant une extraction efficace des motifs δ-libres dans le contexte difficile des données larges. Nous venons de voir à la section précédente que la méthode de transposition échoue. Pourtant, les motifs δ-libres sont très utiles. Ils peuvent en effet servir à la construction de règles admettant un nombre borné d'exceptions [BBR00, BBR03] ou non redondantes [BPT$^+$00, Zak00b]. De plus, leur capacité à indiquer les propriétés minimales induisant un phénomène est précieuse en caractérisation de classes [CB02] et en classification [Bay04]. C'est pourquoi nous proposons dans le chapitre 3 une nouvelle méthode d'extraction des motifs δ-libres. Cette méthode est basée sur l'extension des motifs δ-libres et permet leur obtention dans les données larges, là où les autres approches échouent.

Après nous être attachés aux difficultés algorithmiques de l'extraction de motifs, nous nous intéressons au chapitre suivant à la qualité de l'information inférée à partir des motifs extraits d'une base de données : nous recensons les travaux dont l'objectif est d'évaluer la pertinence des règles d'association.

Chapitre 2

Évaluer la pertinence des règles d'association

Sommaire

Ce chapitre discute des stratégies élaborées pour faire face au grand nombre de règles d'association qui résultent d'applications pratiques. Certaines sont syntaxiques et portent sur la forme des règles, d'autres sont basées sur des calculs de probabilités qui servent à quantifier l'intérêt d'une règle. Les techniques subjectives sont des procédures de sélection où l'utilisateur intervient alors que les méthodes objectives ne nécessitent aucune connaissance experte. Ce chapitre est consacré aux méthodes objectives qui sont à la source du cadre générique pour les mesures d'intérêt que nous proposons au chapitre 4.

La section 2.1 fait le point sur l'extraction des règles d'association. La section 2.2 définit la notion de *règle informative* et expose le principe des couvertures de règles, qui permettent d'éliminer de nombreuses règles redondantes. La section 2.3 recense les travaux existants sur les mesures d'intérêt. Ces dernières servent à sélectionner les règles d'association les plus intéressantes. Leur utilisation donne également un ordre sur les règles sélectionnées qu'il est possible de ranger des meilleures aux moins pertinentes.

2.1 Extraction des règles d'association

Définition de la tâche Depuis [AIS93], il est souvent implicite que la tâche d'extraction de règles d'association consiste à extraire les règles qui dépassent un seuil de support et de confiance (respectivement notés min_{supp} et min_{conf}) fixés par l'utilisateur. De telle règles sont dites *valides*. La définition 12 rappelle brièvement les notions de support et de confiance.

Définition 12 *Le support d'une règle d'association $X \rightarrow Y$ est égal à $\frac{\mathcal{F}(XY)}{|\mathcal{D}|}$ et sa confiance à $\frac{\mathcal{F}(XY)}{\mathcal{F}(X)}$ où $\mathcal{F}(X)$ dénote la fréquence du motif X.*

Notons qu'une règle dont la confiance vaut 1 est qualifiée d'*exacte*. Le calcul des règles d'association dépend fortement de celui des motifs fréquents. En effet, celui-ci peut être décomposé [AS94] en :

1. l'extraction des motifs fréquents et de leur fréquence ;
2. la génération des règles valides qui découlent de chaque motif fréquent.

La première étape peut être réalisée à l'aide de n'importe quelle méthode d'extraction de motifs fréquents évoquée à la section 1.1.1 (cf. chapitre précédent). L'étape de génération des règles consiste à décliner les sous-ensembles propres X' de chaque motif fréquent X puis à tester si les règles de la forme $X' \rightarrow X \backslash X'$ sont valides. Comme X est fréquent, on est assuré que le support de la règle $X' \rightarrow X \backslash X'$ dépasse min_{supp}. Pour chaque règle générée, il reste à vérifier que sa confiance est supérieure à min_{conf}. Cette vérification est triviale puisqu'il suffit de calculer le ratio $\frac{\mathcal{F}(X)}{\mathcal{F}(X')}$ et que la fréquence de tous les motifs fréquents est connue, donc celles de X et de X' le sont également. Dans certains cas, il est même possible d'éviter cette vérification : quand la règle $X_1 \rightarrow X \backslash X_1$ a une confiance inférieure à min_{conf} alors il en est de même pour toutes les règles $X_2 \rightarrow X \backslash X_2$ avec $X_2 \supset X_1$.

Dans l'exemple du tableau 1 à la page 4, en fixant $min_{supp} = \frac{1}{3}$, le motif fréquent $a_2a_3a_6$ permet de générer les six règles suivantes, toutes de support $\frac{1}{3}$ et dont la confiance est donnée entre parenthèses :

$$a_2 \rightarrow a_3a_6 \quad \left(\tfrac{2}{3}\right) \qquad a_3 \rightarrow a_2a_6 \quad \left(\tfrac{2}{5}\right) \qquad a_6 \rightarrow a_2a_3 \quad \left(\tfrac{2}{3}\right)$$

$$a_2a_3 \rightarrow a_6 \quad \left(\tfrac{2}{3}\right) \qquad a_2a_6 \rightarrow a_3 \quad (1) \qquad a_3a_6 \rightarrow a_2 \quad (1)$$

Si on pose $min_{conf} = \frac{2}{3}$, seule la règle $a_3 \rightarrow a_2a_6$ est éliminée car sa confiance est en-dessous de ce seuil. En considérant tous les motifs 2-fréquents constitués des attributs a_2, a_3, a_6 et a_8 de ce même exemple, 50 règles sont générées et 9 d'entre elles sont supprimées pour cause de confiance trop faible.

Fixer min_{supp} entre 0 et 1 revient à chercher les motifs dont la fréquence est supérieure à $min_{supp} \times |\mathcal{D}|$. Le degré de confiance accordé à une règle peut être représenté par le nombre d'exceptions de cette règle i.e. le nombre d'objets qui vérifient la prémisse de la règle mais pas sa conclusion. Le support et le degré de confiance d'une règle peuvent aussi s'exprimer avec un nombre absolu d'objets. Dans [BBR00], on appelle règle δ-forte une règle dont le nombre d'exceptions est borné.

Définition 13 (Règle δ-forte) *Soit δ un entier positif. Une règle d'association $r : X \rightarrow Y$ admettant moins de δ exceptions i.e. vérifiant $\mathcal{F}(X) - \mathcal{F}(XY) \leq \delta$ est appelée règle δ-forte.*

Ce point de vue permet d'autoriser un nombre d'exceptions « raisonnable » dans les cas pratiques. Dans ce document, nous utilisons plutôt la fréquence (absolue) et le nombre d'exceptions d'une règle car cela facilite l'obtention des résultats au chapitre 4.

Règles à conclusion restreinte Remarquons que les règles d'association dont la conclusion est soit fixée soit limitée à un seul attribut font l'objet d'une attention toute particulière dans la littérature [AIS93]. En effet, ces règles ont de nombreuses applications. Les règles qui concluent sur un attribut de classe sont utilisées par exemple pour la production de motifs émergents [DL99],

de règles de caractérisation et de classification [LHM98, CB02] et sont à la base de la construction de nombreux modèles provenant de motifs locaux (i.e., associations entre les attributs) comme les classifieurs fondés sur les associations [LHM98, BG03]. [BA99] montre qu'elles possèdent des propriétés particulières pour certaines mesures d'intérêt. Par ailleurs, comme nous le verrons au chapitre suivant, des algorithmes plus efficaces peuvent souvent être mis au point puisque la tâche d'extraction se trouve simplifiée.

Limitations des règles d'association Une limitation importante de l'approche des règles d'association provient de la quantité de règles générées. Dans la pratique, le nombre élevé de règles produites rend leur exploitation difficile. Par ailleurs, l'ensemble des règles valides contient un grand nombre de règles sans intérêt. Différentes approches tentent de pallier ce problème en sélectionnant des règles particulières afin de cerner l'information la plus pertinente. Citons l'utilisation de templates pour donner des critères de filtrage sur les règles [KMR+94], la spécification de contraintes portant sur les prémisses ou conclusions des règles [NLHP98], le regroupement de règles basées sur leur similarité afin de structurer leur présentation à l'utilisateur [LSW97], les couvertures qui éliminent une règle selon le contexte global dans lequel elle se trouve [Zak00a], la recherche de paires de règles dont l'une est valide et l'autre est rare [Suz03], l'utilisation de mesures d'intérêt [GH06] ayant pour but de faire émerger les règles les plus significatives des autres ou encore la visualisation de l'espace des règles [Bla05].

Dans ce chapitre d'état de l'art, nous nous intéressons plus particulièrement aux couvertures de règles (section 2.2) et aux mesures d'intérêt (section 2.3), ces notions étant largement utilisées dans la suite de ce travail.

2.2 Les couvertures de règles

L'objectif d'une couverture de règles est de diminuer le nombre de règles produites tout en assurant de pouvoir regénérer exactement l'ensemble de toutes les règles valides ainsi que leur support et leur confiance si besoin est [Kry02]. Outre la réduction du nombre de règles, l'utilisation d'une couverture présente deux avantages : dans certains cas, l'extraction est rendue faisable puisque la couverture est moins volumineuse que l'ensemble complet des règles ; de plus, elle peut permettre de se focaliser sur les règles les plus intéressantes.

Illustration Reprenons l'exemple du tableau 1 page 4 pour illustrer la redondance des règles d'association. Les règles suivantes ont toutes un support de $\frac{1}{3}$ et une confiance égale à 1 :

$\mathbf{r_1 : a_2 \rightarrow a_3}$	$r_2 : a_2a_6 \rightarrow a_3$	$r_3 : a_2a_6 \rightarrow a_8$	$r_4 : a_2a_8 \rightarrow a_3$	$r_5 : a_2a_8 \rightarrow a_6$
$\mathbf{r_6 : a_2a_6 \rightarrow a_3a_8}$	$\mathbf{r_7 : a_2a_8 \rightarrow a_3a_6}$	$r_8 : a_2a_3a_6 \rightarrow a_8$	$r_9 : a_2a_3a_8 \rightarrow a_6$	$r_{10} : a_2a_6a_8 \rightarrow a_3$

Or, on voit que les règles r_2, r_3, r_4 et r_5 n'apportent pas l'information maximale dans le sens où il existe des règles (r_6 et r_7) constituées des mêmes prémisses et concluant sur un motif qui contient plus d'attributs, r_6 et r_7 véhiculent donc plus d'information. Les règles r_8, r_9 et r_{10} comportent, quant à elles, un attribut de plus qu'il n'est nécessaire pour aboutir à la même conclusion avec une confiance égale à 1. Non seulement les règles r_1, r_6 et r_7 sont les plus pertinentes, mais il est de plus possible de retrouver toutes les autres règles à partir de ces trois-là. Pour retrouver r_2 à r_5, il faut garder les prémisses de r_6 et r_7 et décliner les conclusions composées d'un seul attribut. On regénère r_8, r_9 et r_{10} en faisant basculer un attribut de la conclusion vers la prémisse dans les règles r_6 et r_7.

Toujours dans l'exemple jouet du tableau 1, les règles suivantes ont également un support de $\frac{1}{3}$ mais une confiance égale à $\frac{2}{3}$:

$$\begin{array}{lll} r_{11} : a_2 \rightarrow a_6 & r_{12} : a_2 \rightarrow a_8 & r_{13} : a_2 \rightarrow a_3a_6 \\ r_{14} : a_2 \rightarrow a_3a_8 & r_{15} : a_2 \rightarrow a_6a_8 & \mathbf{r_{16} : a_2 \rightarrow a_3a_6a_8} \\ r_{17} : a_2a_3 \rightarrow a_6 & r_{18} : a_2a_3 \rightarrow a_8 & r_{19} : a_2a_3 \rightarrow a_6a_8 \end{array}$$

On peut effectuer un raisonnement similaire lorsque la confiance n'est pas égale à 1. En fait, l'attribut a_3 n'est pas nécessaire pour conclure sur a_6, a_8 ou a_6a_8 (règles r_{17}, r_{18} et r_{19}) et l'attribut a_2 permet à lui tout seul de conclure sur la conjonction d'attributs a_6a_8 comme dans la règle r_{15}. Par ailleurs, les règles r_{11} à r_{15} n'apportent pas l'information maximale car r_{16} a une conclusion plus spécifique. Il résulte de ces observations que la règle $r_{16} : a_2 \rightarrow a_3a_6a_8$ est une synthèse intéressante de ces 9 règles : c'est celle qui apporte le plus d'information à partir d'hypothèses minimales. À partir de la seule règle r_{16} et des règles exactes valides, il est possible de regénérer les 9 règles énumérées plus haut. Il suffit, à partir de r_{16}, de construire les règles de même prémisse mais avec une conclusion plus générale et les règles de prémisse plus générale et de conclusion plus spécifique en excluant les règles de confiance 1 déjà générées.

Règles informatives Nous introduisons maintenant la notion de règle *informative*, développée par Bastide *et al.* [BPT+00] et qui est basée sur ces observations. Ces règles constituent une couverture des règles valides et leur construction est liée aux propriétés des représentations condensées. Parmi les règles de même support et de même confiance, on sélectionne celles dont la prémisse est minimale au sens de l'inclusion et dont la conclusion est maximale i.e. celles qui apportent le plus d'information à partir d'hypothèses minimales. Bastide *et al.* montrent qu'une telle règle est constituée d'un motif libre en prémisse et que la réunion de sa prémisse et de sa conclusion est un motif fermé (voir la section 1.2 pour une définition de ces termes). Comme vu sur l'exemple précédent, Bastide *et al.* définissent deux ensembles de règles : les règles informatives exactes qui constituent une base pour les règles exactes et les règles informatives approximatives qui forment une base pour les autres règles. C'est la réunion de ces deux ensembles qui constitue une couverture des règles valides. Remarquons qu'il est encore possible de réduire l'ensemble des règles considérées en se focalisant sur une base dite *réduite* qui ne conserve que les règles basées sur certains motifs fermés. Les autres règles sont alors déduites par un axiome de transitivité. Plusieurs prototypes sont dédiés à l'extraction de ces couvertures comme PRINCE [HYS05] (base réduite) ou ZART [Sza06].

Il existe d'autres bases de règles comme par exemple la base Guigues-Duquenne [GD86], la base de Luxenburger [Lux91] étendue par Kryszkiewicz [Kry02], les règles d'association non redondantes de Zaki [Zak04]. Cependant, ces bases ne permettent pas toujours de regénérer exactement l'ensemble des règles valides ou le support et la confiance des règles. Leurs propriétés sont détaillées dans [GYNS06]. [GYNS06] introduit également une nouvelle base nommée $\mathcal{IGB}$ qui est sans perte d'information.

Règles de caractérisation δ-fortes La notion de minimalité de la prémisse est parfois utilisée pour bénéficier de la concision de l'information extraite sans forcément chercher à définir une couverture de règles. Un autre avantage est la limitation du phénomène de sur-apprentissage lorsque ces règles sont utilisées en classification. Par exemple, la définition des règles de caractérisation δ-fortes [CB02], qui sera utile au chapitre 3, repose sur la notion de minimalité.

Définition 14 *Soient γ et δ deux entiers positifs. Une règle de caractérisation δ-forte est une règle δ-forte de la forme $r : X \rightarrow c_i$ γ-fréquente à prémisse minimale i.e. il n'existe aucune règle de la forme $Y \rightarrow c_i$ avec $Y \subset X$ qui admet une confiance supérieure à $1 - \frac{\delta}{\gamma}$.*

La propriété 5 [CB02] relie les règles de caractérisation δ-fortes aux motifs δ-libres :

Propriété 5 *Soit r une règle de caractérisation δ-forte. La prémisse de r est un motif δ-libre.*

Dans les chapitres 3 et 6, nous verrons que nous avons pu utiliser avec succès notre méthode d'extraction des motifs δ-libres dans les données larges pour l'obtention de règles de caractérisation δ-fortes, ce qui a permis la caractérisation de classes dans des données d'expression de gènes.

Conclusion L'intérêt majeur des couvertures est de réduire le nombre de règles sans perte d'information. Malgré cela, le nombre de règles d'association reste généralement trop élevé pour pouvoir être appréhendé par l'utilisateur. De plus, l'utilisation des seuls critères de support et de confiance pour évaluer les règles se révèle insuffisante [BMS97, Gui00]. C'est pour sélectionner les règles les plus pertinentes de manière fine que de nombreux indices, appelés *mesures d'intérêt*, ont été mis au point.

2.3 Qualité des règles : les mesures d'intérêt

2.3.1 Vue générale

L'idée générale d'une mesure d'intérêt est d'associer une valeur numérique à une règle afin de quantifier son intérêt. De nombreuses mesures d'intérêt ont été définies pendant les 2 dernières décennies, un excellent panorama en est donné dans [GH06]. Plusieurs articles [HH99, Fre99, McG05] tentent d'organiser les travaux portant sur les mesures d'intérêt en donnant une vue d'ensemble de ceux-ci. L'ensemble des mesures d'intérêt est généralement divisé en deux groupes : les mesures objectives [BVW03] et les mesures subjectives [ST96]. Les premières ne dépendent que des propriétés intrinsèques aux règles d'association considérées alors que les secondes intègrent des connaissances expertes. Nous nous focalisons uniquement sur les mesures objectives dans ce manuscrit. Les plus utilisées sont indiquées dans les tableaux 2.1 et 2.2, leurs définitions proviennent de [TKS02, VMP+05, Bla05, GH06].

2.3.2 Caractériser les bonnes mesures

Face à l'abondance des mesures d'intérêt, certains auteurs [PS91, MM95, Fre99, HH03, LT04] se sont intéressés à définir ce qu'est une « bonne » mesure d'intérêt. En 1991, Piatetsky-Shapiro [PS91] définit un cadre destiné à évaluer la qualité d'une mesure. Il énonce alors trois propriétés que doit vérifier une bonne mesure. Ces propriétés notées P1, P2 et P3, sont résumées à la définition 15. Elles portent sur le comportement d'une mesure en fonction de la prémisse et de la conclusion d'une règle d'association.

Définition 15 *Soit $r : X \rightarrow Y$ une règle d'association. Une mesure M doit vérifier les trois propriétés suivantes :*

- ***P1*** *: $M(r) = 0$ si X et Y sont statistiquement indépendants dans $\mathcal{D}$;*
- ***P2*** *: Lorsque $\mathcal{F}(X)$ et $\mathcal{F}(Y)$ sont fixées, $M(r)$ est strictement croissante en la fréquence de r $\mathcal{F}(XY)$;*
- ***P3*** *: Lorsque $\mathcal{F}(XY)$ est fixée, $M(r)$ est strictement décroissante :*

Nom	Définition
Support	$\frac{\mathcal{F}(XY)}{\lvert\mathcal{D}\rvert}$
Coverage	$\frac{\mathcal{F}(X)}{\lvert\mathcal{D}\rvert}$
Prevalence	$\frac{\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert}$
Confiance/Confidence	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)}$
Laplace (k=2)	$\frac{\mathcal{F}(XY)+1}{\mathcal{F}(X)+2}$
Sensibilité/Sensitivity/Recall	$\frac{\mathcal{F}(XY)}{\mathcal{F}(Y)}$
Jaccard	$\frac{\mathcal{F}(XY)}{\mathcal{F}(Y)+\mathcal{F}(X)-\mathcal{F}(XY)}$
Intérêt/Lift	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)} \times \frac{\lvert\mathcal{D}\rvert}{\mathcal{F}(Y)}$
Information Gain	$\log\left(\frac{\mathcal{F}(XY)}{\mathcal{F}(X)} \times \frac{\lvert\mathcal{D}\rvert}{\mathcal{F}(Y)}\right)$
Taux des exemples et contre-exemples	$\frac{2\mathcal{F}(XY)-\mathcal{F}(X)}{\mathcal{F}(XY)}$
Indice de Ganascia	$\frac{2\mathcal{F}(XY)-\mathcal{F}(X)}{\mathcal{F}(X)}$
Moindre-contradiction	$\frac{2\mathcal{F}(XY)-\mathcal{F}(X)}{\mathcal{F}(Y)}$
Rule-Interest [PS91]	$\frac{\mathcal{F}(XY)\times\lvert\mathcal{D}\rvert-\mathcal{F}(X)\times\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert}$
Nouveauté/Novelty [LFZ99]	$\frac{\mathcal{F}(XY)\times\lvert\mathcal{D}\rvert-\mathcal{F}(X)\times\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert^2}$
Added Value	$\frac{\mathcal{F}(XY)\times\lvert\mathcal{D}\rvert-\mathcal{F}(X)\times\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert\times\mathcal{F}(X)}$
Indice de Loevinger/Certainty Factor	$\frac{\mathcal{F}(XY)\times\lvert\mathcal{D}\rvert-\mathcal{F}(X)\times\mathcal{F}(Y)}{\mathcal{F}(X)\times(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}$
Coefficient de corrélation/ϕ-coefficient	$\frac{\mathcal{F}(XY)\times\lvert\mathcal{D}\rvert-\mathcal{F}(X)\times\mathcal{F}(Y)}{\sqrt{\mathcal{F}(X)\times\mathcal{F}(Y)\times(\lvert\mathcal{D}\rvert-\mathcal{F}(X))\times(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}}$
Sebag & Schoenauer	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)-\mathcal{F}(XY)}$
Multiplicateur de cotes/Growth rate	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)-\mathcal{F}(XY)} \times \frac{\lvert\mathcal{D}\rvert-\mathcal{F}(Y)}{\mathcal{F}(Y)}$
Conviction	$\frac{\mathcal{F}(X)}{\mathcal{F}(X)-\mathcal{F}(XY)} \times \frac{\lvert\mathcal{D}\rvert-\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert}$
Rapport de cotes/Odds ratio	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)-\mathcal{F}(XY)} \times \frac{\lvert\mathcal{D}\rvert-\mathcal{F}(X)-\mathcal{F}(Y)+\mathcal{F}(XY)}{\mathcal{F}(Y)-\mathcal{F}(XY)}$
Spécificité/Specificity	$\frac{\lvert\mathcal{D}\rvert-\mathcal{F}(Y)-\mathcal{F}(X)+\mathcal{F}(XY)}{\lvert\mathcal{D}\rvert-\mathcal{F}(X)}$
Indice de Sokal et Michener/Success Rate	$\frac{\lvert\mathcal{D}\rvert-\mathcal{F}(Y)-\mathcal{F}(X)+2\mathcal{F}(XY)}{\lvert\mathcal{D}\rvert}$

TAB. 2.1 – Exemples de mesures d'intérêt.

- *en* $\mathcal{F}(X)$ *quand* $\mathcal{F}(Y)$ *est fixée ;*
- *en* $\mathcal{F}(Y)$ *quand* $\mathcal{F}(X)$ *est fixée.*

La propriété P1 indique qu'une mesure doit prendre la valeur nulle lorsque la prémisse et la conclusion d'une règle apparaissent tout à fait indépendamment l'une de l'autre dans une base de données. Cela signifie qu'une telle règle est due au hasard et n'a que peu d'intérêt. Dans les propriétés P2 et P3, une mesure d'intérêt est considérée de façon implicite comme une fonction dont on interprète les variations relativement à la fréquence d'une règle, de sa prémisse et de sa conclusion. La propriété P2 impose la croissance d'une mesure d'intérêt avec la fréquence d'une règle. En d'autres termes, toute bonne mesure d'intérêt doit privilégier le choix des règles qui apparaissent le plus fréquemment dans la base de données, lorsque les autres quantités intervenant dans le calcul de cette mesure sont fixées. La propriété P3 indique que les règles dont la prémisse ou la conclusion sont fréquentes dans $\mathcal{D}$ présentent un intérêt moindre. En effet, une conclusion fréquente n'est pas forcément pertinente : lorsqu'un motif est très présent dans la base, il existe une probabilité élevée de conclure sur ce motif, même par hasard. Dans [PS91], Piatetsky-Shapiro définit également une mesure d'intérêt appelée *Rule Interest* qui satisfait les trois propriétés qui viennent d'être détaillées. Dans le chapitre 4, nous montrerons comment le cadre que nous proposons pour les mesures d'intérêt est lié à celui de Piatetsky-Shapiro mais aussi ce qui différencie ces deux approches tant au niveau des objectifs visés que des résultats obtenus.

Nom	Définition
Indice de Roger et Tanimoto	$\frac{\lvert\mathcal{D}\rvert - \mathcal{F}(X) - \mathcal{F}(Y) + 2\mathcal{F}(XY)}{\lvert\mathcal{D}\rvert + \mathcal{F}(X) + \mathcal{F}(Y) - 2\mathcal{F}(XY)}$
Collective strength	$\frac{\mathcal{F}(XY)(\lvert\mathcal{D}\rvert-\mathcal{F}(Y)-\mathcal{F}(X)+\mathcal{F}(XY))}{\mathcal{F}(X)\mathcal{F}(Y)+(\lvert\mathcal{D}\rvert-\mathcal{F}(X))(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))} \times \frac{\lvert\mathcal{D}\rvert^2-\mathcal{F}(X)\mathcal{F}(Y)-(\lvert\mathcal{D}\rvert-\mathcal{F}(X))(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}{\lvert\mathcal{D}\rvert-\mathcal{F}(XY)-(\lvert\mathcal{D}\rvert-\mathcal{F}(X)-\mathcal{F}(Y)+\mathcal{F}(XY))}$
Indice de Dice	$\frac{\mathcal{F}(XY)}{\mathcal{F}(XY) + \frac{1}{2}(\mathcal{F}(X) - \mathcal{F}(XY) + \mathcal{F}(Y) - \mathcal{F}(XY))}$
Indice de Kulczynski	$\frac{\mathcal{F}(XY)}{2}\left(\frac{1}{\mathcal{F}(X)} + \frac{1}{\mathcal{F}(Y)}\right)$
Indice d'Ochiai/IS mesure	$\frac{\mathcal{F}(XY)}{\sqrt{\mathcal{F}(X)\mathcal{F}(Y)}}$
Contribution orientée au χ^2	$\frac{\mathcal{F}(XY) - \frac{\mathcal{F}(X)\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert}}{\sqrt{\frac{\mathcal{F}(X)\mathcal{F}(Y)}{\lvert\mathcal{D}\rvert}}}$
Indice d'implication	$\frac{\mathcal{F}(X) - \mathcal{F}(XY) - \frac{\mathcal{F}(X)(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}{\lvert\mathcal{D}\rvert}}{\sqrt{\frac{\mathcal{F}(X)(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}{\lvert\mathcal{D}\rvert}}}$
Kappa κ	$\frac{\lvert\mathcal{D}\rvert\mathcal{F}(XY)+\lvert\mathcal{D}\rvert(\lvert\mathcal{D}\rvert-\mathcal{F}(X)-\mathcal{F}(Y)+\mathcal{F}(XY))-\mathcal{F}(X)\mathcal{F}(Y)-(\lvert\mathcal{D}\rvert-\mathcal{F}(X))(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}{\lvert\mathcal{D}\rvert^2-\mathcal{F}(X)\mathcal{F}(Y)-(\lvert\mathcal{D}\rvert-\mathcal{F}(X))(\lvert\mathcal{D}\rvert-\mathcal{F}(Y))}$
Relative risk	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)} \times \frac{\lvert\mathcal{D}\rvert - \mathcal{F}(X)}{\mathcal{F}(Y) - \mathcal{F}(XY)}$
Indice de Yule	$\frac{\mathcal{F}(XY)(\lvert\mathcal{D}\rvert-\mathcal{F}(Y)-\mathcal{F}(X)+\mathcal{F}(XY))-(\mathcal{F}(X)-\mathcal{F}(XY))(\mathcal{F}(Y)-\mathcal{F}(XY))}{\mathcal{F}(XY)(\lvert\mathcal{D}\rvert-\mathcal{F}(Y)-\mathcal{F}(X)+\mathcal{F}(XY))+(\mathcal{F}(X)-\mathcal{F}(XY))(\mathcal{F}(Y)-\mathcal{F}(XY))}$
Leverage	$\frac{\mathcal{F}(XY)}{\mathcal{F}(X)} - \frac{\mathcal{F}(X) \times \mathcal{F}(Y)}{\lvert\mathcal{D}\rvert^2}$

TAB. 2.2 – Exemples de mesures d'intérêt (suite).

2.3.3 Comparer les mesures entre elles

D'autres auteurs définissent également des propriétés sur les mesures dans le but de les regrouper de manière homogène afin de faire ressortir leurs similarités et différences. Tan et al. [TKS02] étudient 21 mesures d'intérêt et proposent cinq propriétés permettant d'indiquer pour quel domaine applicatif ces mesures sont appropriées. Ils proposent même un algorithme dédié à la seule tâche de choisir une mesure adaptée. Il est observé expérimentalement qu'un encadrement du support des règles induit des similitudes fortes dans le comportement de nombreuses mesures d'intérêt. Nous donnerons au chapitre 4 les arguments théoriques qui confirment ces observations. Après s'être intéressés à plusieurs contextes applicatifs, les auteurs concluent par la phrase suivante : « We show that there is no measure that is consistently better than others in all cases. »

Pour déterminer des points communs à plusieurs mesures, Bayardo et Agrawal proposent dans [BA99] d'exprimer les mesures d'intérêt en fonction des mesures de support et de confiance lorsque cela est possible. Fürnfranz et Flach montrent que certaines mesures peuvent être exprimées les unes en fonction des autres [FF05] et en déduisent des similarités au sein de plusieurs ensembles de mesures. Vaillant *et al.* mettent également en évidence des similitudes de comportements [VLL04] en effectuant un clustering sur une vingtaine de mesures. Dans [PNSL06], une étude graphique à partir de courbes de niveaux est menée pour 6 indices afin de déterminer le plus approprié à la discrimination de règles dans des données décrivant des véhicules automobiles.

2.3.4 Bilan

Malgré les efforts fournis pour clarifier le domaine des mesures d'intérêt, il n'existe pas de cadre générique consensuel pour l'ensemble de ces mesures. Les différents travaux évoqués poursuivent des objectifs distincts :

- présenter une nouvelle mesure en justifiant son introduction par la capture inédite de certaines caractéristiques de l'information apportée par les règles d'association. Nous n'avons pas détaillé ces travaux car ils sont trop nombreux.
- dire quelles mesures sont les meilleures ou quelles mesures un utilisateur final devrait utiliser.
- proposer une taxonomie des mesures d'intérêt en fonction de propriétés communes mises en évidence de manière théorique ou pratique.

Dans la pratique, l'utilisation de mesures d'intérêt reste délicate : il est compliqué de choisir la mesure adéquate (qui capture le type d'information souhaité) et de fixer un seuil pour l'utiliser. C'est pourquoi nous proposons dans le chapitre 4 une vue unifiée d'un grand nombre de mesures d'intérêt.

2.4 Conclusion

Dans ce chapitre, nous avons décrit le fonctionnement et l'intérêt des couvertures de règles. De plus, nous avons énuméré les différents travaux sur les mesures d'intérêt et montré les limites de leur utilisation en pratique.

Au chapitre 4, nous apportons des contributions à ces deux volets de la sélection de règles d'association. Nous proposons un cadre générique qui englobe un grand nombre de mesures d'intérêt et qui montre des similitudes dans leur comportement. En nous appuyant sur les motifs libres, nous définissons un ensemble réduit de règles qui optimisent toutes ces mesures simultanément. Dans le cas des règles de classification, nous obtenons ainsi une couverture des règles qui optimisent toutes les mesures d'intérêt de notre cadre.

Conclusion

Cette partie a permis de pointer deux obstacles majeurs s'opposant à la découverte d'information pertinente lors d'un processus de fouille de données. D'une part, le chapitre 1 a mis en lumière le manque de méthodes dédiées aux données larges. En particulier, il conclut sur un constat d'échec pour la découverte des motifs δ-libres dans ce type de contexte. Dans le même temps, il souligne l'intérêt de ces motifs et démontre l'enjeu essentiel de leur extraction. D'autre part, le chapitre 2 a mis en évidence des lacunes dans l'évaluation des règles d'association. Notamment, l'abondance des mesures d'intérêt, parfois présentée comme un atout, est ici vue comme un handicap car elle mène à la confusion tout en donnant une fausse impression de variété.

Les deux premiers chapitres de la partie suivante proposent des solutions à ces problèmes. Nous présentons au chapitre 3 une approche qui permet l'extraction des motifs δ-libres dans les données larges. L'idée clé de cette approche est de travailler sur les objets plutôt que sur les attributs, en s'appuyant plus précisément sur l'utilisation des extensions des motifs. Par ailleurs au chapitre 4, nous donnons un cadre formel pour les mesures d'intérêt et présentons un ensemble de règles informatives qui optimisent de nombreuses mesures d'intérêt.

Deuxième partie

Découverte et usages des motifs minimaux

Introduction

La partie précédente a d'une part mis en évidence l'échec des méthodes classiques pour l'extraction des motifs δ-libres dans les données comportant un grand nombre d'attributs ainsi que l'inadéquation des travaux visant à sélectionner les règles les plus pertinentes. D'autre part, elle a montré que les motifs libres et δ-libres sont au centre de ces problématiques. C'est pourquoi nous proposons dans chacun des chapitres suivants une contribution à l'extraction et aux usages des motifs libres.

Le problème des données larges est de générer des motifs d'attributs qui sont longs. Au contraire, dans de tels contextes, les motifs d'objets sont courts. Nous montrons au chapitre 3 l'intérêt de travailler avec des motifs d'objets, plus précisément l'extension des motifs. Nous proposons une nouvelle méthode d'extraction des motifs δ-libres qui exploite cette idée et qui est efficace dans les données larges. De plus, une adaptation de cette méthode à l'extraction de règles de caractérisation de classes est proposée.

Le calcul de nombreuses mesures d'intérêt dépend des mêmes paramètres. C'est pourquoi ces mesures adoptent des comportements proches en pratique. Nous proposons au chapitre 4 un cadre formel qui englobe de nombreuses mesures usuelles (appelées SBMs) et qui garantit l'existence de valeurs minimales pour toutes ces mesures. Les règles qui vérifient cette propriété sont identifiées et appelées règles optimisées. Celles dont la prémisse est un motif libre et dont la conclusion provient d'un motif fermé, fournissent l'information maximale au sens des règles informatives de Bastide *et al.* [BPT$^+$00], et nous indiquons une méthode efficace pour les déterminer.

Enfin, nous immergeons la notion de traverse minimale d'un hypergraphe dans le cadre de la fouille de données. Nous montrons au chapitre 5 comment la résolution du problème de l'obtention des motifs δ-libres dans les données larges par l'utilisation de l'extension permet le calcul efficace des traverses minimales d'un hypergraphe dans un algorithme par niveaux.

Chapitre 3

Extraction de motifs locaux et de règles fondés sur les δ-libres

Sommaire

Dans ce chapitre, nous présentons une nouvelle méthode d'extraction des motifs δ-libres dans les bases de données comportant un grand nombre d'attributs [HC05]. Cette méthode repose sur l'utilisation de l'extension d'un motif i.e. des objets qui contiennent ce motif. La section 3.1 explique comment l'extension peut être utilisée pour la recherche de motifs δ-libres. La section 3.2 donne les critères d'élagages que nous appliquons pour diminuer la taille de l'espace de recherche. Nous présentons un nouveau critère qui résulte de la combinaison des contraintes de fréquence et de liberté. La section 3.3 détaille notre algorithme FTMINER et la section 3.4 décrit les expériences que nous avons menées. Enfin la dernière section de ce chapitre montre comment notre méthode peut être appliquée à la caractérisation de classes.

3.1 Une approche basée sur l'extension

3.1.1 Motivations, intuitions

Nous avons vu au chapitre 1 que les méthodes existantes d'extraction de motifs dans les données larges ne fonctionnent pas pour la contrainte de δ-liberté [RBCB03]. La principale faiblesse de l'algorithme d'extraction de motifs δ-libres présenté dans [BBR00] réside dans le calcul des presque-fermetures. Premièrement, ce calcul est particulièrement coûteux en temps dans les données comportant un grand nombre d'attributs (voir les expériences à la section 3.4). Deuxièmement, les fermetures sont des motifs relativement longs (les presque-fermetures le sont encore plus) que l'on doit manipuler et stocker. Il en résulte rapidement d'importants problèmes d'espace mémoire. Les extractions peuvent donc facilement nécessiter plusieurs jours ou être infaisables [RBCB03].

Notre proposition est d'éviter le calcul des fermetures en leur préférant des motifs d'objets : les extensions, comme définies dans la section 1.1.2. En effet, les extensions sont des motifs relativement courts dans les données larges, ce qui facilite leur manipulation et leur stockage. De plus, ils permettent une vérification immédiate de la liberté d'un motif, ce qui évite de stocker tous les motifs candidats à un niveau du treillis comme dans l'algorithme Apriori (cf. section 1.1.1). On évite ainsi d'un même coup les problèmes de mémoire et de temps évoqués plus haut. Cette méthode diffère de la technique de transposition évoquée dans la section 1.3. Rappelons que la transposition de données consiste à extraire des motifs d'objets et à en déduire les motifs d'attributs initialement recherchés. Au contraire, nous proposons d'extraire des motifs d'attributs puis de vérifier s'ils correspondent à ce que nous recherchons à l'aide de l'extension associée.

3.1.2 Exprimer la fréquence avec l'extension

L'approche que nous proposons tire profit de la propriété suivante qui relie la fréquence d'un motif à son extension :

Propriété 6 *Soit X un motif d'attributs. Sa fréquence est égale à la cardinalité de son extension i.e. on a :*

$$\mathcal{F}(X) = |g_{\mathcal{D}}(X)|.$$

La démonstration de cette propriété est immédiate si l'on considère la définition de l'extension. Celle-ci étant composée des objets qui contiennent un motif donné X, sa cardinalité est clairement égale au nombre d'occurrences de X dans $\mathcal{D}$. Montrons maintenant comment il est possible de redéfinir les contraintes de fréquence et de δ-liberté (cette dernière est définie à la section 1.2.3) à partir de l'extension d'un motif.

Définition 16 (Motif fréquent) *Soit γ un entier strictement positif. Un motif X est γ-fréquent si et seulement si $|g_{\mathcal{D}}(X)| \geq \gamma$.*

Définition 17 (Motif δ-libre) *Soit δ un entier positif. Un motif X est δ-libre si et seulement si pour tout motif Y inclus strictement dans X, $|g_{\mathcal{D}}(Y)| > |g_{\mathcal{D}}(X)| + \delta$.*

La définition 16 assure que si l'on dispose de l'extension d'un motif, on peut déterminer s'il est fréquent ou non. Cela s'avère moins trivial pour la δ-liberté : la définition 17 indique que la

vérification de cette propriété nécessite additionnellement la connaissance de l'extension des sous-ensembles du motif considéré. Ainsi, il est envisageable de construire un algorithme par niveaux utilisant ces (re)définitions pour extraire les motifs γ-fréquents et δ-libres ; c'est d'ailleurs l'objet de la section 3.3. Pour cela, il est nécessaire de connaître l'extension de chaque motif candidat et de ses sous-ensembles. Il faut donc disposer d'une méthode qui permet de calculer efficacement les extensions pour pouvoir réellement les exploiter. La section suivante fournit une telle méthode de calcul.

3.1.3 Déterminer l'extension des motifs candidats

La propriété suivante [Zak00b] découle de la propriété 3 qui est énoncée à la section 1.1.2. Elle exprime l'extension d'un motif candidat généré lors d'un parcours par niveaux de l'espace de recherche, en fonction de l'extension de chacun des deux motifs générateurs (le candidat est lui-même issu de la réunion de deux motifs générateurs, cf. section 1.1.1).

Propriété 7 (Extension d'un candidat) *Soient $X \cup \{a_1\}$ et $X \cup \{a_2\}$ deux motifs où a_1 et a_2 sont deux attributs. L'extension de $X \cup \{a_1 a_2\}$ est égale à $g_{\mathcal{D}}(X \cup \{a_1\}) \cap g_{\mathcal{D}}(X \cup \{a_2\})$.*

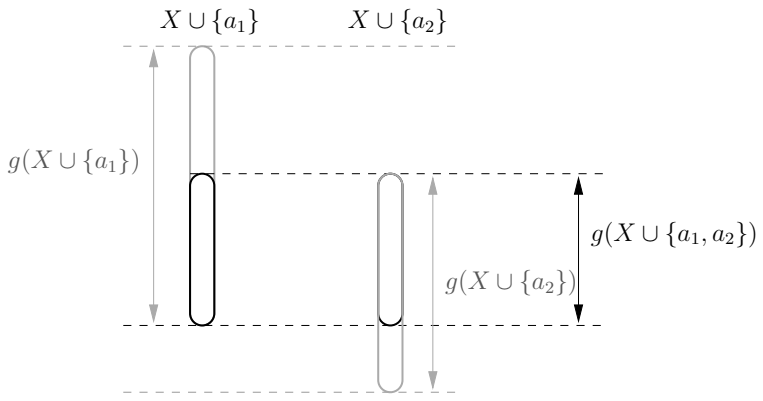

FIG. 3.1 – Calcul de l'extension d'un candidat.

La propriété 7 permet d'effectuer par une simple intersection d'extensions le calcul de l'extension d'un motif candidat. La figure 3.1 montre que l'intersection des objets contenant $X \cup \{a_1\}$ avec ceux qui contiennent $X \cup \{a_2\}$ est composée des objets qui contiennent $X \cup \{a_1, a_2\}$. Par exemple, dans la base de données du tableau 1 donné à la page 1, $g_{\mathcal{D}}(a_1 a_8) = o_3 o_4$ et $g_{\mathcal{D}}(a_3 a_8) = o_3 o_5 o_6$ d'où l'extension de $a_1 a_3 a_8$ qui est égale à $o_3 o_4 \cap o_3 o_5 o_6 = o_3$. L'intérêt majeur de cette approche repose sur le nombre réduit d'objets dans les données larges : ainsi, l'intersection de deux extensions est peu coûteuse. Remarquons que Zaki et Hsiao mettent en oeuvre une approche similaire dans leurs algorithmes d'extraction de motifs fréquents ECLAT [Zak00b] et de motifs fermés CHARM [ZH02] ; les extensions sont nommées *tidsets* [5]. Cependant, ces deux algorithmes effectuent un parcours en profondeur de l'espace de recherche (et non en largeur comme nous le proposons) et il n'est pas fait mention de l'intérêt d'une telle approche pour les données larges.

[5] Dans [ZG03], il est proposé d'utiliser plutôt les complémentaires des tidsets appelés *diffsets*. Cette approche est appliquée avec succès dans l'algorithme dfNDI [CG05]. Nous pensons cependant que dans le cas des données larges où le nombre d'objets est peu élevé, l'utilisation des complémentaires des extensions n'apporte aucun gain.

Nous venons de montrer comment il est possible de réduire la taille des motifs manipulés par rapport à la taille des fermetures des motifs recherchés. Dans la section suivante, nous proposons un nouveau critère d'élagage avec pour objectif de diminuer le nombre de candidats à tester lors de la recherche des motifs fréquents δ-libres.

3.2 Raffinement de l'élagage

Dans cette section, nous présentons les critères d'élagage mis en oeuvre dans notre méthode d'extraction de libres fréquents. En dehors des critères d'élagage liés à l'anti-monotonie de la fréquence et de la δ-liberté, la nouveauté consiste ici à tirer parti de la combinaison de ces deux contraintes pour aller plus loin que l'élagage lié à l'anti-monotonie des deux contraintes.

3.2.1 Élagages classiques

Puisque la fréquence et la δ-liberté sont des contraintes anti-monotones (voir les pages 12 et 20), on peut appliquer les critère d'élagage propres aux contraintes anti-monotones (propriétés 1 et 2) à chacune de ces contraintes. Par exemple, le motif a_3a_5 de notre exemple n'est pas libre car $\mathcal{F}(a_3a_5) = 4 = \mathcal{F}(a_5)$. On peut donc élaguer le treillis à partir de a_3a_5 (cf. figure 1.3). Par conséquent, on ne testera aucune spécialisation de ce motif.

La section suivante montre qu'il est possible d'améliorer la simple conjonction des élagages que nous venons d'évoquer en définissant un élagage subtil qui découle de l'utilisation simultanée de la fréquence et de la δ-liberté.

3.2.2 Élagage par combinaison de contraintes

Les motifs recherchés sont à la fois γ-fréquents et δ-libres. Or, dans certains cas, la combinaison de ces deux contraintes montre une incompatibilité qui donne lieu à un nouveau critère d'élagage. Nous verrons dans les expériences de la section 3.4 que, loin d'être anecdotique, cette situation est en fait très fréquente. Notre nouveau critère produit alors un élagage très efficace. Ce nouveau critère est plus puissant que la simple conjonction des élagages liés à l'anti-monotonie de chacune des deux contraintes.

Pour illustrer cet élagage, reprenons l'exemple du tableau 1 en fixant $\gamma = 2$ et $\delta = 1$. Les motifs a_5 et a_7 sont 2-fréquents ($\mathcal{F}(a_5) = 4$ et $\mathcal{F}(a_7) = 2$) et 1-libres puisque $\mathcal{F}(\emptyset_{\mathcal{A}}) = 6 > \mathcal{F}(a_5)+1 = 5$ (rappelons que les motifs 1-libres sont soulignés de deux traits sur la figure 1.3). Les critères classiques énoncés plus haut ne permettent pas d'élaguer les sur-ensembles de a_5 ou de a_7. Le motif a_5a_7 est donc généré par réunion de ces deux attributs et il faut alors vérifier si il est 2-fréquent et 1-libre. Or on constate que $\mathcal{F}(a_7) \leq \gamma+\delta = 3$. Pour que a_5a_7 soit 2-fréquent, il faut que sa fréquence soit supérieure ou égale à 2. Pour que ce même motif soit 1-libre, il faut que sa fréquence soit strictement inférieure à $\mathcal{F}(a_7) - 1 = 1$ d'où une contradiction : a_5a_7 ne peut pas être à la fois 2-fréquent et 1-libre. Le motif a_5a_7 peut être écarté sans calcul de fréquence.

Le théorème 1 et le critère d'élagage 1 expriment les observations précédentes d'une manière plus formelle.

Théorème 1 *Soit X un motif. Si X est γ-fréquent et δ-libre alors tout sous-ensemble Y de X vérifie $|g_{\mathcal{D}}(Y)| > \gamma + \delta$.*

Preuve *Ce théorème résulte directement des définitions de la fréquence et de la δ-liberté en fonction de l'extension données à la section 3.1.2. X est γ-fréquent et δ-libre donc les définitions 16 et 17 impliquent que $Y \subset X,\ \gamma + \delta \leq |g_{\mathcal{D}}(X)| + \delta < |g_{\mathcal{D}}(Y)|$.*

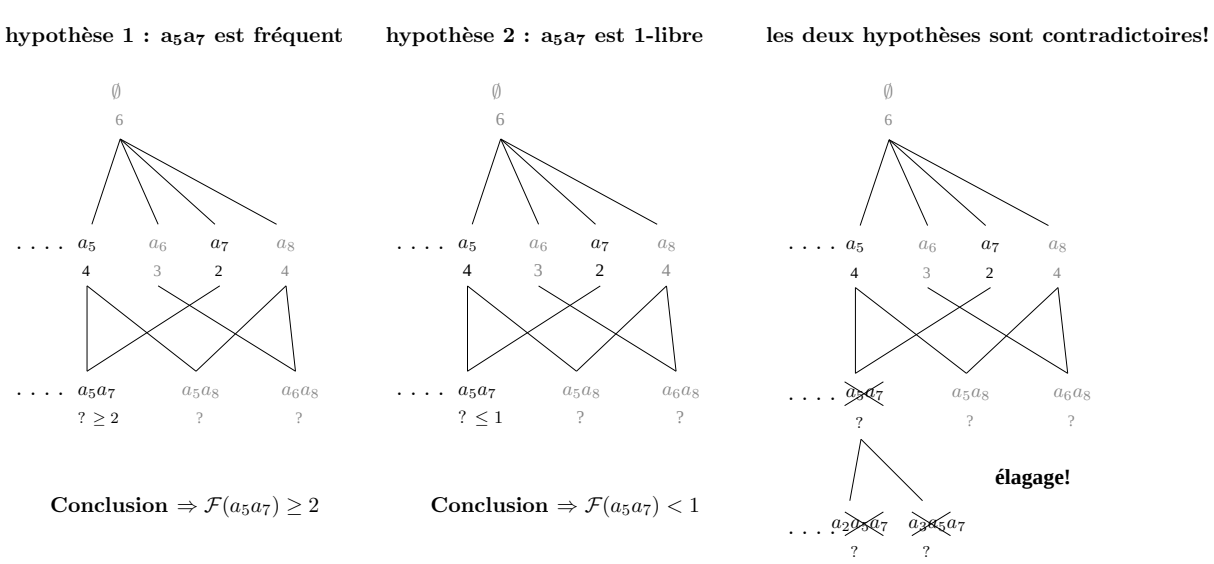

FIG. 3.2 – Élagage issu de la conjonction des contraintes de fréquence et de δ-liberté.

La contraposée du théorème 1 donne le critère d'élagage suivant : si la fréquence d'un motif est inférieure ou égale à la somme des deux paramètres γ et δ alors aucun de ses sur-ensembles n'est à la fois δ-libre et γ-fréquent.

Critère d'élagage 1 *Soit X un motif tel que $|g_{\mathcal{D}}(X)| \leq \gamma + \delta$. Aucune spécialisation de X n'est à la fois γ-fréquente et δ-libre. On peut donc élaguer les spécialisations de X dans un algorithme par niveaux d'extraction des motifs γ-fréquents et δ-libres.*

Ce nouveau critère d'élagage est évalué expérimentalement dans la section 3.4.3. Comme sa vérification repose uniquement sur des extensions, il est naturellement exploitable sans surcoût par notre approche.

3.3 Extraction des motifs δ-libres : l'algorithme FTMINER

Nous donnons dans cette section notre algorithme d'extraction des motifs δ-libres FTMINER (pour Free faT[6] databases MINER). FTMINER (cf. algorithme 1) est élaboré suivant le principe des algorithmes par niveaux et tire parti des observations précédentes en utilisant largement les extensions de motifs. Son originalité réside dans le fait qu'il n'y a qu'une seule passe sur la base de données effectuée lors de l'initialisation, au niveau 1. Ensuite, les motifs candidats d'un niveau $k > 1$ ne sont pas tous stockés simultanément, évitant ainsi des problèmes de mémoire. Ils sont générés et testés un par un puisque le calcul de leur extension ne nécessite aucun accès à la base de données (cf. propriété 7).

$\mathcal{F}ree_k$ est l'ensemble des motifs γ-fréquents et δ-libres de longueur k. $\mathcal{G}en_k$ est l'ensemble des générateurs de longueur k qui seront utilisés lors de la génération des candidats de longueur $k+1$ i.e. des éléments de $\mathcal{F}ree_k$ non éliminés par le critère d'élagage 1.

L'initialisation des ensembles $\mathcal{F}ree_1$ et $\mathcal{G}en_1$ est effectuée aux lignes 1 et 2 de l'algorithme 1 en parcourant la base de données $\mathcal{D}$. La boucle principale démarre à la ligne 4. Elle stoppe au

[6] En anglais, le terme « fat » est parfois utilisé à la place de « large » pour caractériser les bases de données larges comme indiqué par D. Hand pendant sa conférence invitée à PKDD'04.

premier niveau rencontré où il n'y a plus de générateurs. À la ligne 6, on génère un candidat de longueur $k+1$ en fusionnant deux générateurs de longueur k ayant un préfixe commun de longueur $k-1$. Cette procédure est similaire à la procédure de génération des candidats de l'algorithme APRIORI décrite à la page 13 du chapitre 1. Le calcul de l'extension de ce candidat repose sur la propriété 7 et nécessite l'intersection des deux générateurs. On vérifie ligne 8 que le candidat est γ-fréquent avec la définition 16. On teste la δ-liberté aux lignes 10 à 12 avec la définition 17. Lorsque le candidat vérifie les contraintes requises, on l'ajoute à $\mathcal{F}ree_{k+1}$ (ligne 13). Si de plus sa fréquence est supérieure à $\gamma+\delta$ i.e. il n'est pas éliminé par le critère d'élagage 1, on l'ajoute à l'ensemble des générateurs $\mathcal{G}en_{k+1}$ (lignes 14 et 15).

Entrée : base de données $\mathcal{D}$, seuil de fréquence γ, nombre d'exceptions δ
Sortie : les motifs γ-fréquents et δ-libres contenus dans $\mathcal{D}$
`// initialisation de` $\mathcal{F}ree_1$`, l'ensemble des attributs δ-libres`
1 $\mathcal{F}ree_1 := \{a \in \mathcal{A} \mid |\mathcal{O}| - \delta > |g_{\mathcal{D}}(a)| \geq \gamma\}$;
`// initialisation de` $\mathcal{G}en_1$`, l'ensemble des attributs générateurs`
2 $\mathcal{G}en_1 := \{a \in \mathcal{F}ree_1 \mid |g_{\mathcal{D}}(a)| > \gamma + \delta)\}$;
3 $k := 1$;
`// boucle principale`
4 **tant que** $\mathcal{G}en_k \neq \emptyset$ **faire**
5 **pour tout** $(Y \cup \{A\}, Y \cup \{B\}) \in \mathcal{G}en_k \times \mathcal{G}en_k$ **faire**
`// génération d'un candidat` X `de longueur` $k+1$
6 $X := Y \cup \{A\} \cup \{B\}$;
`// calcul de son extension`
7 $g_{\mathcal{D}}(X) := g_{\mathcal{D}}(Y \cup \{A\}) \cap g_{\mathcal{D}}(Y \cup \{B\})$;
`// test pour la contrainte de fréquence`
8 **si** $|g_{\mathcal{D}}(X)| \geq \gamma$ **alors**
9 $i := 1$;
`// test pour la contrainte de δ-liberté`
`// les attributs de` X `sont notés` $x_1, x_2, \ldots, x_{k+1}$
10 **tant que** $i \leq k+1$ **et** $X \backslash \{x_i\} \in \mathcal{G}en_k$ **et** $|g_{\mathcal{D}}(X)| + \delta < |g_{\mathcal{D}}(X \backslash \{x_i\})|$ **faire**
11 $i := i + 1$;
fin
12 **si** $i = k+2$ **alors**
13 $\mathcal{F}ree_{k+1} := \mathcal{F}ree_{k+1} \cup \{X\}$;
14 **si** $|g_{\mathcal{D}}(X)| > \gamma + \delta$ **alors**
15 $\mathcal{G}en_{k+1} := \mathcal{G}en_{k+1} \cup \{X\}$;
fin
fin
fin
fin
16 $k := k+1$;
fin
17 **retourner** $\bigcup_{i=1}^{k-1} \mathcal{F}ree_i$;

Algorithme 1 : FTMINER

Théorème 2 *L'algorithme* FTMINER *extrait tous les motifs γ-fréquents et δ-libres de la base de données $\mathcal{D}$ donnée en entrée, et seulement ceux-ci.*

Preuve *Tout motif X dans $\mathcal{F}ree_k$ est γ-fréquent (test à la ligne 8) et δ-libre (test aux lignes 10*

à 12), d'après les définitions 16 et 17. L'algorithme FTminer *est donc correct.*
De plus, FTminer *couvre entièrement l'espace de recherche puisqu'il effectue un parcours par niveaux de celui-ci. Les critères d'élagage utilisés sont les critères classiques liés à l'anti-monotonie de la fréquence et de la liberté, et le nouveau critère énoncé à la page 41. Puisqu'ils sont corrects, l'algorithme* FTminer *est complet.*

3.4 Expériences

3.4.1 Protocole expérimental

Objectifs et matériel utilisé L'objectif de ces expériences est double. D'une part, elles visent à évaluer l'efficacité de FTminer par rapport aux méthodes existantes et à montrer que FTminer permet de calculer les motifs γ-fréquents et δ-libres dans des situations où les autres prototypes échouent (section 3.4.2). D'autre part, elles soulignent l'importance du nouveau critère d'élagage énoncé à la section 3.2.2 (section 3.4.3).

C'est pourquoi dans la section 3.4.2, nous commençons par comparer FTminer au prototype MVminer. MVminer a été développé par François Rioult (GREYC) et permet l'extraction de motifs en présence de valeurs manquantes [RC03]. Pour ces expériences menées sur des données ne comportant pas de valeurs manquantes, il est équivalent à ACminer implémenté par Artur Bykowski (LIRIS) et basé sur l'algorithme MinEx [BBR00, BBR03] (cf. section 1.2.3 à la page 19). Dans la pratique, ACminer s'avère être plus rapide que MVminer pour extraire les motifs δ-libres. Cependant, MVminer a l'avantage de consommer moins d'espace mémoire que ACminer. Or dans les expériences effectuées sur les données SAGE, ACminer échoue pour toutes les extractions. C'est pourquoi nous avons préféré comparer FTminer à MVminer, ce dernier permettant de fournir un point de comparaison pour ce type d'expériences.

Tous les tests ont été effectués avec un processeur Xeon 2,20 GHz avec 3 Go de RAM sous Linux. Tous les temps d'exécution sont donnés en secondes.

Présentation des données Nous commençons par tester les deux prototypes sur des benchmarks usuels provenant de l'université d'Irvine en Californie (UCI). Nous utilisons plus particulièrement les benchmarks `CMC`, `ABALONE`, `MUSHROOM` et `PUMSB` qui sont téléchargeables en ligne à l'url `http://www.ics.uci.edu/~mlearn/MLSummary.html`. Comme il existe peu de benchmarks avec beaucoup d'attributs, nous avons transposé des benchmarks aux dimensions classiques afin d'obtenir des données larges. `CMC` et `ABALONE` produisent ainsi des jeux de données de dimensions respectives $30 \times 1\,474$ et $30 \times 4\,178$ [7]. Cette manipulation, limitée à la préparation des données, ne doit pas prêter à confusion ; rappelons que notre méthode n'utilise pas la technique de transposition. Par curiosité, nous comparons également les deux algorithmes sur des données aux dimensions plus « classiques » : pour cela nous utilisons `MUSHROOM` ($8\,124 \times 120$) et `PUMSB` ($49\,046 \times 7\,118$), évidemment sans les transposer.

Outre des benchmarks usuels, nous testons FTminer sur les données d'expression de gènes qui sont des données larges comme nous l'avons vu en introduction. Les données SAGE que nous utilisons [8] se présentent sous la forme d'une matrice binaire comportant 90 lignes et 27 679 colonnes i.e. une matrice très large (voir la section 6.1 pour une description détaillée). Pour s'assurer que le succès de notre méthode n'est pas lié à une quelconque particularité des données SAGE, nous testons également FTminer sur une autre base de données génomique, la base de

[7] Les dimensions des bases sont données sous la forme *nombre de lignes*×*nombre de colonnes.*
[8] Ces données ont au préalable été préparées par Sylvain Blachon du laboratoire CGMC [Bla07].

données GDS464 qui provient du Gene Expression Omnibus repository. Elle est téléchargeable à l'url http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=464 et contient l'expression de 7 085 gènes dans 90 situations biologiques.

3.4.2 Comportement dans les données larges

Benchmarks transposés

La figure 3.3 donne les temps d'exécution de MVMINER et FTMINER lors de l'extraction des motifs libres pour la base CMC transposée (courbe de gauche) et pour ABALONE transposée (courbe de droite). Pour CMC, le seuil de fréquence γ est compris entre 10 et 6 (soit 33% et 20%). Pour ABALONE, il varie entre 9 et 6 (ce qui équivaut à 30% et 20%).

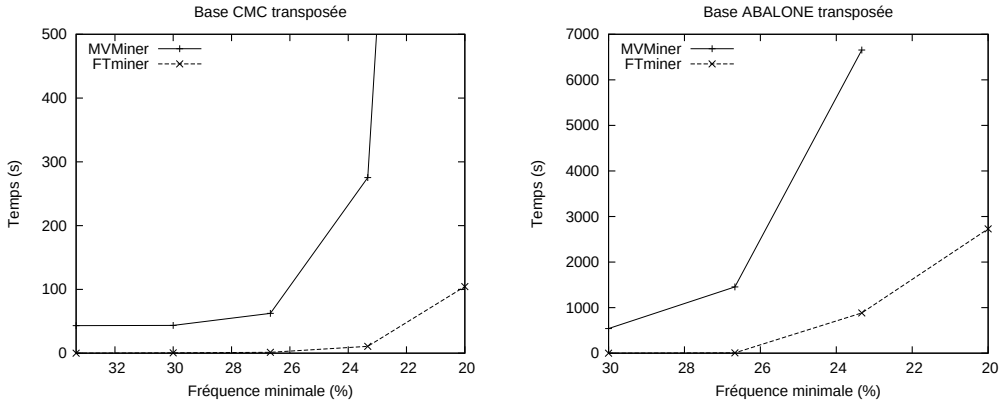

FIG. 3.3 – Performances sur des benchmarks transposés.

Il était attendu que les temps d'exécution diminuent quand le seuil de fréquence augmente. FTMINER permet d'extraire les libres de fréquence supérieure à 6 (20%) de CMC en moins de deux minutes alors que MVMINER a besoin de plus de trois quarts d'heure pour terminer l'extraction [9] (ACMINER échoue faute d'espace mémoire). Lorsque γ vaut 5, seul FTMINER permet d'obtenir tous les motifs libres. MVMINER échoue également dans l'extraction des 6-fréquents du jeu de données ABALONE (ACMINER échoue dès que γ est inférieur ou égal à 8). Quelques tests complémentaires ont montré que FTMINER est beaucoup plus rapide que ACMINER lorsque ce dernier réussit à terminer l'extraction demandée pour ces benchmarks.

Données d'expression de gènes

Nous effectuons maintenant des comparaisons similaires sur les données réelles d'expression de gènes.

Données SAGE La courbe de gauche de la figure 3.4 donne, pour les données SAGE, les temps d'exécution des deux prototypes pour l'extraction des 3-libres quand γ varie de 30 à 24 (on utilise une échelle logarithmique en ordonnée). Quand γ vaut 30, FTMINER extrait les 3-libres en 30 secondes alors que MVMINER a besoin d'une journée entière. Pour un seuil de fréquence de 29, FTMINER termine l'extraction en 50 secondes alors que MVMINER met plus de deux jours.

[9] Ce temps n'apparaît pas sur la figure car il écraserait trop le reste des courbes et les rendrait illisibles.

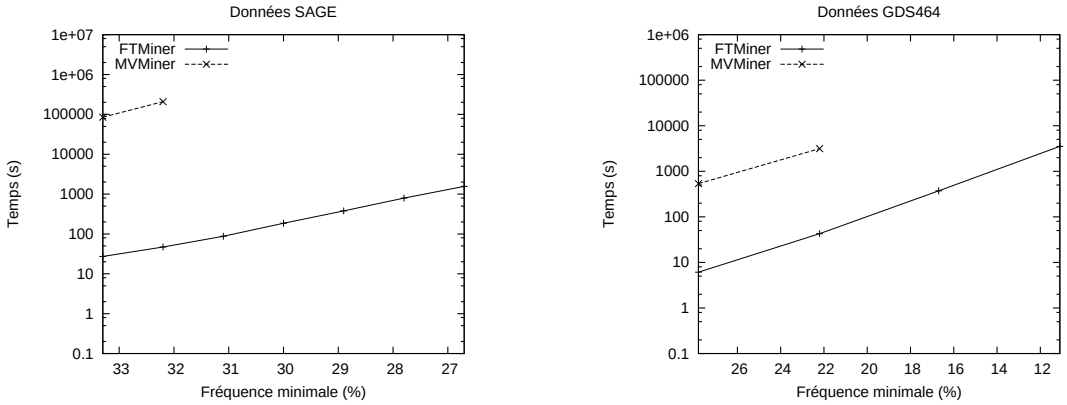

FIG. 3.4 – Performances sur des données d'expression de gènes.

Données du Gene Expression Omnibus repository La courbe de droite de la figure 3.4 donne les temps d'extraction des motifs 2-libres en fonction de γ sur les données du Gene Expression Omnibus repository, à nouveau avec une échelle logarithmique. L'extraction devient impossible à partir d'une fréquence de 20% pour MVMINER

Ces expériences montrent que notre approche permet l'extraction de motifs libres dans des données réelles très larges quand les autres méthodes échouent.

Données aux dimensions classiques À titre de curiosité, nous avons comparé FTMINER et MVMINER pour l'extraction des motifs libres sur des bases aux dimensions usuelles. La figure 3.5 montre que FTMINER est à nouveau plus rapide que MVMINER même si `MUSHROOM` et `PUMSB` comportent beaucoup d'objets et peu d'attributs. Il est particulièrement difficile d'extraire des motifs fréquents dans `PUMSB`, c'est pourquoi les seuils de fréquence sont très élevés lorsqu'on travaille sur ce jeu de données. Quand le seuil de fréquence relatif est fixé à $75,5\%$ sur cette base, FTMINER manque rapidement de mémoire et ne peut terminer l'extraction alors que MVMINER termine en 8 829 secondes (deux heures et demie environ). Cela n'est pas suprenant puisque lorsque le nombre d'objets est élevé, le calcul des extensions des motifs candidats est une opération coûteuse. FTMINER se comporte cependant de façon correcte dans ces cas qui lui sont défavorables et pour lesquels il n'a pas été conçu.

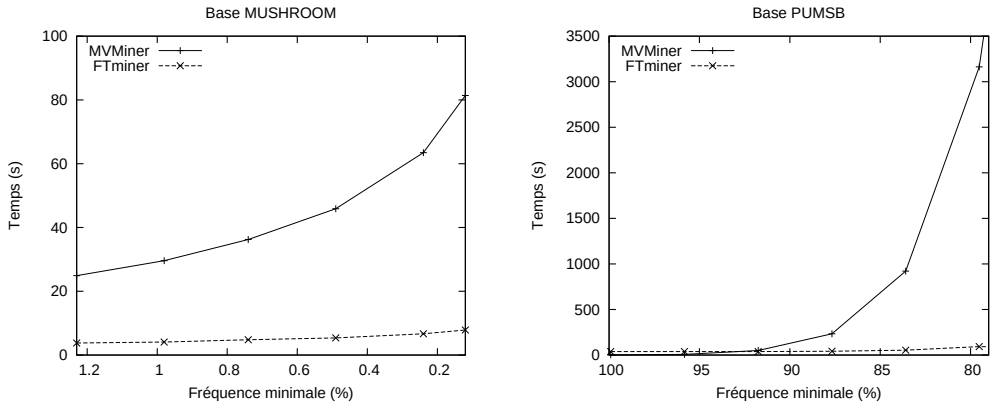

FIG. 3.5 – Performances sur des données aux dimensions « classiques ».

3.4.3 Évaluation du nouveau critère d'élagage

Dans cette section, nous quantifions l'apport du critère d'élagage (critère 1) présenté dans la section 3.2.2. La figure 3.6 montre les temps d'extraction en fonction de δ sur les données SAGE avec et sans ce critère. γ est fixé à 27 et δ varie de 6 à 2. Le gain en temps est important : quand δ est égal à 5, FTMINER met 31 secondes à réaliser l'extraction en appliquant cet élagage contre 527 secondes sans l'appliquer. En moyenne, sur cette expérience, les temps d'extraction sont divisés par 7. Cela provient de la diminution spectaculaire du nombre de candidats qui découle de l'utilisation de ce critère d'élagage : par exemple pour $\gamma = 27$ et $\delta = 5$, ce nombre est divisé par 52, passant de 732 557 270 à 14 056 991.

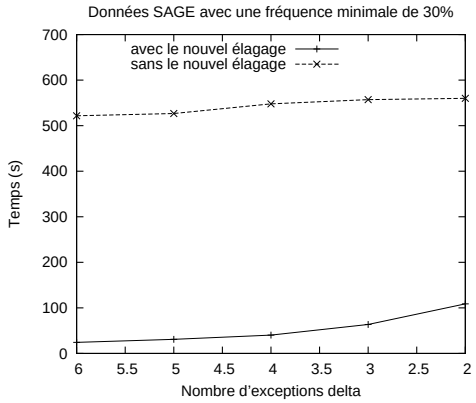

FIG. 3.6 – Efficacité du nouveau critère d'élagage de FTMINER.

3.4.4 Bilan

Ces expériences prouvent que l'utilisation de l'extension et d'un nouveau critère d'élagage permettent d'extraire les motifs δ-libres dans le contexte difficile des données larges où d'autres approches échouent. Elles montrent également que notre méthode basée sur l'extension peut aussi être utilisée sur des bases de données aux dimensions plus habituelles. Pour obtenir ces résultats, nous avons combiné les atouts des représentations condensées (recherche de motifs δ-libres) et la dissymétrie des jeux de données dits larges (utilisation de l'extension préférée à celle de la fermeture). Notre approche doit son succès à deux points clés : la diminution de la taille des motifs à traiter en utilisant l'extension et la diminution du nombre de candidats grâce au nouveau critère d'élagage. La section suivante décrit un prolongement de notre approche pour la caractérisation de classes dans les données larges.

3.5 Caractérisation de classes

Dans cette section, nous montrons que l'approche présentée dans les sections précédentes peut être étendue à la recherche de règles de caractérisation δ-fortes (cf. définition 14 à la page 27).

3.5.1 Règles de caractérisation δ-fortes

Notre intérêt pour la recherche de règles de caractérisation dans les données larges tire son origine d'un problème concret. À partir des données d'expression de gènes, une attente des biologistes est de pouvoir déterminer les ensembles de gènes corégulés qui sont impliqués dans le

développement de cancers. Les règles de caractérisation δ-fortes [CB02] répondent particulièrement bien à ce type d'attente. Tout d'abord, ces règles concluent uniquement sur des valeurs de classes. De plus, leurs prémisses étant minimales, ces règles sont à la fois générales et concises. Elles prennent en compte l'incertitude des données en autorisant un nombre borné d'exceptions. Techniquement, une règle de caractérisation δ-forte a un motif δ-libre pour prémisse (cf. propriété 5 page 27). Notons cependant que la réciproque est fausse puisque tout motif δ-libre ne donne pas lieu à une règle de caractérisation δ-forte [CB02].

Nous reprenons notre exemple habituel (la base de données du tableau 1) que nous enrichissons d'un objet o_7 et de deux valeurs de classes c_1 et c_2 pour fournir un exemple de contexte de caractérisation de classes au tableau A.1. Notons que le motif a_5a_7 est 2-fréquent et que les objets contenant a_5a_7 contiennent tous la valeur de classe c_1. Par ailleurs, la confiance des règles $a_5 \rightarrow c_1$ et $a_7 \rightarrow c_1$ est strictement inférieure à 1 puisqu'elles admettent toutes deux des exceptions. La règle $a_5a_7 \rightarrow c_1$ est donc une règle de caractérisation 0-forte. Remarquons que conformément à la propriété 5, le motif a_5a_7 est 0-libre.

		Attributs								Classe	
		a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	c_1	c_2
Objets	o_1	1	0	1	0	1	0	1	0	1	0
	o_2	0	1	1	0	1	0	1	0	1	0
	o_3	1	0	1	0	1	0	0	1	1	0
	o_4	1	0	0	1	0	1	0	1	0	1
	o_5	0	1	1	0	0	1	0	1	0	1
	o_6	0	1	1	0	1	1	0	1	0	1
	o_7	1	0	1	0	0	1	1	1	1	0

TAB. 3.1 – Un exemple de contexte de classification.

Pour déterminer le nombre d'exceptions d'une règle δ-forte, il faut être en mesure de calculer la presque-fermeture de sa prémisse (cf. section 1.2.3). La section suivante montre comment l'extension intervient dans la découverte des règles de caractérisation δ-fortes.

3.5.2 Utiliser l'extension pour l'extraction de règles

La propriété 8 permet de vérifier l'appartenance d'un attribut à la presque-fermeture d'un motif à partir de l'extension.

Propriété 8 (Appartenance à la presque-fermeture) *Soient δ un entier positif et X un motif. Un attribut a dans $\mathcal{A}$ appartient à la presque-fermeture de X si et seulement si $|g_{\mathcal{D}}(X)| - |g_{\mathcal{D}}(X) \cap g_{\mathcal{D}}(a)| \leq \delta$.*

En d'autres termes, la propriété 8 signifie que la règle $X \rightarrow a$ est δ-forte. Cette propriété découle directement de la propriété 6 et des définitions de l'extension et de la presque-fermeture d'un motif.

Rappelons en outre que la propriété 3 (cf. page 3) permet de décomposer l'extension d'un motif en l'intersection de deux de ses sous-ensembles : le calcul de $g_{\mathcal{D}}(Xa)$ est ainsi décomposé en $g_{\mathcal{D}}(X) \cap g_{\mathcal{D}}(a)$. Il est donc nécessaire de connaître l'extension des attributs qui représentent les valeurs de classes pour déterminer si ils appartiennent ou non à la presque-fermeture d'un

motif. Comme ils sont en nombre limité, leurs extensions peuvent être initialisées en effectuant un parcours de la base de données puis stockées à moindre coût pendant l'exécution de l'algorithme.

3.5.3 L'algorithme FTCminer

Nous détaillons dans cette section le fonctionnement de notre algorithme FTCminer [10] qui extrait les règles de caractérisation δ-fortes. FTCminer reprend en grande partie le fonctionnement de FTminer. Les motifs δ-libres X sont extraits comme dans FTminer. La différence réside dans le test d'existence d'une valeur de classe c_i telle que $X \rightarrow c_i$ soit une règle de caractérisation δ-forte à l'aide la propriété 8.

Élagage L'algorithme FTCminer fait évidemment usage des critères d'élagages exposés à la section 3.2. Additionnellement, un critère reposant sur la propriété 9 [CB02] est utilisé.

Propriété 9 *Soit $X \rightarrow c_i$ une règle de caractérisation δ-forte. Pour tout attribut de classe c_j, la règle $Y \rightarrow c_j$ avec $X \subset Y$ n'est pas une règle de caractérisation δ-forte.*

La propriété 9 implique qu'il est inutile de considérer les spécialisations de la prémisse d'une règle de caractérisation δ-forte puisqu'aucune d'entre elles ne peut donner lieu à une règle de caractérisation δ-forte. Dans FTCminer, les prémisses des règles de caractérisation δ-fortes ne seront donc pas utilisées pour la génération de candidats de longueur supérieure.

Algorithme Nous explicitons uniquement les parties de FTCminer qui diffèrent de FTminer dans l'algorithme 2 :

- la ligne 1' est une étape d'initialisation ajoutée pour l'algorithme 2 ;
- les lignes 12 à 15 de l'algorithme 1 sont remplacées par les lignes 12 à 15 de l'algorithme 2 ;
- la ligne 17 de l'algorithme 2 remplace la ligne 17 de l'algorithme 1.

Entrée : base de données $\mathcal{D}$, seuil de fréquence γ, nombre d'exceptions δ
Sortie : les règles de caractérisation δ-fortes contenues dans $\mathcal{D}$
`// initialisation de` $\mathcal{R}$`, l'ensemble des règles de caractérisation` δ`-fortes`
1' $\mathcal{R} := \{a \rightarrow c_i \mid a \in \mathcal{F}ree_1$ et $|g_{\mathcal{D}}(a)| - |g_{\mathcal{D}}(a) \cap g_{\mathcal{D}}(c_i)| \leq \delta)\}$;
12 **si** $i = k+2$ **alors**
  $\mathcal{F}ree_{k+1} := \mathcal{F}ree_{k+1} \cup \{X\}$;
13 **si** $\exists c_i \in \mathcal{C}$ *tel que* $|g_{\mathcal{D}}(X)| - |g_{\mathcal{D}}(X) \cap g_{\mathcal{D}}(c_i)| \leq \delta$ **alors**
    $\mathcal{R} := \mathcal{R} \cup \{X \rightarrow c_i\}$;
  fin
  sinon
14 **si** $|g_{\mathcal{D}}(X)| > \gamma + \delta$ **alors**
15 $\mathcal{G}en_{k+1} := \mathcal{G}en_{k+1} \cup \{X\}$;
  fin
  fin
fin
17 **retourner** $\mathcal{R}$;

Algorithme 2 : FTCminer

À la ligne 1', on initialise $\mathcal{R}$ avec les attributs δ-libres qui permettent de conclure sur une valeur de classe avec moins de δ exceptions. Ensuite, pour chaque δ-libre extrait avec FTminer,

[10] Les lettres FTC signifient Free faT Characterization.

on regarde dans la liste $\mathcal{C}$ des attributs de classes si il y a une règle valide i.e. avec moins de δ exceptions avec la propriété 8 (cf. ligne 13). Si c'est le cas, la règle obtenue est ajoutée à $\mathcal{R}$ mais pas à l'ensemble des candidats de longueur $k+1$ appliquant ainsi l'élagage issu de la propriété 9. Si aucune règle n'est générée à partir du motif X, celui-ci est ajouté à l'ensemble des générateurs (ligne 15).

Nous illustrons l'utilisation de FTCminer au chapitre 6 avec l'extraction de règles de caractérisation δ-fortes dans les données SAGE.

3.6 Conclusion

Nous avons proposé dans ce chapitre une approche permettant de déterminer les motifs δ-libres dans des jeux de données très larges. Pour cela, nous avons montré l'intérêt de la notion d'extension et réalisé une étude fine des possibilités d'élagage. Nous avons étendu cette approche à la caractérisation de classes en adaptant notre algorithme à l'extraction de règles δ-fortes qui concluent sur des attributs de classes. L'intérêt pratique des méthodes présentées dans ce chapitre est souligné par une application à la découverte de gènes corégulés au chapitre 6.

Chapitre 4

Qualité des règles d'association : une vue unifiée des mesures d'intérêt

Sommaire

Au chapite 2, nous avons souligné l'intérêt des règles d'association mais aussi que leur surabondance rend leur utilisation difficile. Nous avons brièvement présenté des techniques visant à faire face à cette profusion de règles, dont les règles informatives et les mesures d'intérêt. Dans la première partie de ce chapitre, nous proposons un cadre formel qui permet de mieux appréhender le fonctionnement des mesures d'intérêt [HC07]. Nous montrons qu'un grand nombre d'entre elles (que nous appelons SBMs pour `Simultaneously Bounded Measures`) dépendent des mêmes paramètres et qu'elles ont un comportement similaire. Ce résultat permet de garantir des valeurs minimales pour toutes les SBMs et la production de règles de qualité par rapport à l'ensemble de ces mesures. En outre, nous étendons la notion de règle informative habituellement réservée aux seules mesures de support et de confiance à toutes les SBMs. Nous montrons qu'il est possible d'extraire efficacement un ensemble réduit de règles à prémisse minimale et à conclusion maximale qui véhiculent l'information la plus pertinente et par conséquent, que l'utilisation des règles qui optimisent toutes les SBMs est aisée en pratique.

La section 4.1 introduit notre travail sur les mesures d'intérêt. Dans la section 4.2, nous menons une étude des propriétés communes à un grand nombre de mesures usuelles et nous définissons un cadre formel pour les SBMs. En nous appuyant sur ce cadre, nous montrons que les SBMs ont des comportements similaires. Dans la section 4.3, nous identifions les règles d'association admettant une valeur minimale pour chaque SBM. De plus, nous définissons un ensemble réduit parmi ces règles et fournissons une méthode efficace d'extraction de cet ensemble. La section 4.4 montre comment notre cadre peut être simplifié dans le cas particulier des règles de classification. Enfin, la section 4.5 décrit les expériences que nous avons menées.

4.1 Préambule

Positionnement Dans ce chapitre, nous nous intéressons uniquement aux mesures d'intérêt objectives (cf. section 2.3). Nous avons vu à la section 2.3.2 que de nombreux travaux visent à déterminer les caractéristiques d'une bonne mesure d'intérêt. Dans ce travail, notre but est différent. Même si nous énonçons des propriétés caractérisant les mesures d'intérêt, nous ne cherchons pas à dépeindre une mesure « idéale » [PS91] mais à concevoir un cadre générique montrant les ressemblances entre de nombreuses mesures. La section 2.3 a largement souligné le foisonnement des mesures d'intérêt et le besoin de les répertorier ainsi que d'en comprendre le fonctionnement. Notre cadre répond à ces attentes en donnant une vue unifiée des mesures d'intérêt objectives et en explicitant leur fonctionnement.

Qualité des règles : quelques intuitions Il existe quelques règles tacites qui relèvent du bon sens et qui font l'unanimité pour définir la pertinence d'une règle d'association. En premier lieu, une règle doit être suffisamment fréquente pour être représentative. Ensuite, une règle doit admettre un petit nombre d'exceptions pour être fiable. Il est évident qu'une règle plus souvent fausse que vraie présente peu d'intérêt. Enfin, une règle qui conclut sur un motif très fréquent apporte peu d'information. En effet, un motif qui est présent dans presque tous les objets d'une base de données est peu significatif et la probabilité de conclure sur ce motif par hasard est élevée. Nous pensons donc que la plupart des mesures d'intérêt ont été définies de manière à ne pas contredire ces quelques remarques de bon sens. Le cadre théorique que nous présentons à la section 4.2 peut être vu comme une formalisation des ces observations et son étude aboutit à la conclusion que de nombreuses mesures d'intérêt sont équivalentes. Il montre qu'il est possible de garantir une valeur minimale pour beaucoup de mesures, à condition d'imposer à une règle d'association r une valeur minimale γ pour la fréquence de sa prémisse, une valeur maximale η pour la fréquence de sa conclusion et un maximum δ pour le nombre d'exceptions. C'est finalement l'étude du comportement des mesures d'intérêt en fonction des trois paramètres γ, η et δ qui révèle des similarités importantes.

Exemple introductif Beaucoup de mesures d'intérêt sont construites de manière similaire : elles dépendent de la fréquence d'une règle et de celle de sa prémisse. Considérons l'exemple de la confiance sur les règles δ-fortes (définition 13 à la page 24). Supposons que la règle $r : X \rightarrow Y$ ait moins de 2 exceptions (elle est 2-forte) et que sa prémisse X apparaisse dans au moins 5 objets de $\mathcal{D}$. Alors sa confiance sera au moins égale à $\frac{3}{5}$ car la proportion d'objets où la règle r n'est pas vérifiée est au plus égale à $\frac{2}{5}$. Ce résultat garantit une confiance minimale de $1 - \frac{\delta}{\gamma}$ pour toute règle δ-forte dont la prémisse dépasse le seuil de fréquence γ [11]. Il est alors naturel

[11] L'existence d'une confiance minimale est démontrée dans [CB02] pour les règles de caractérisation δ-fortes.

de chercher à étendre ce résultat de valeur seuil à d'autres mesures d'intérêt. Malheureusement, beaucoup de mesures tiennent compte de la fréquence de la conclusion d'une règle. Une variable supplémentaire entre alors dans le calcul de la valeur de la mesure et la généralisation du résultat précédent n'est pas triviale [12]. Prenons l'exemple de la sensibilité avec la même règle $r : X \rightarrow Y$ 2-forte dont la prémisse est 5-fréquente. On est assuré que la fréquence de r est supérieure à 3. Supposons de plus que la conclusion a une fréquence inférieure à 4. Alors la sensibilité de r est nécessairement supérieure à $\frac{3}{4}$. Nous verrons plus formellement à la section 4.2.4 comment le fait de majorer la fréquence de la conclusion d'une règle permet de généraliser l'exemple de la confiance minimale à de nombreuses mesures d'intérêt.

4.2 Un cadre unificateur : les SBMs

Cette section présente notre cadre formel pour les mesures d'intérêt. Tout d'abord, les mesures d'intérêt sont assimilées à des fonctions (section 4.2.1). Ensuite, nous pointons les caractéristiques communes à un grand nombre de mesures usuelles à la section 4.2.2 et les mesures d'intérêt qui les vérifient sont appelées SBMs à la section 4.2.3. La section 4.2.4 montre qu'il est possible de garantir des valeurs minimales pour toutes les SBMs simultanément et la section 4.2.5 explicite le comportement de ces minorants.

4.2.1 Appréhender les mesures d'intérêt comme des fonctions

Dans cette section, nous réécrivons une mesure d'intérêt M sous la forme d'une fonction à trois variables. Nous verrons dans la section suivante que cette réécriture permet d'exprimer des propriétés qui caractérisent de nombreuses mesures d'intérêt.

Définition 18 (Fonction associée à une mesure) *Soient $r : X \rightarrow Y$ une règle d'association et M une mesure d'intérêt. $\Psi_M(x, y, z)$ est la fonction continue obtenue en remplaçant $\mathcal{F}(X)$ par x, $\mathcal{F}(Y)$ par y et $\mathcal{F}(XY)$ par z dans l'expression $M(r)$.*

Par exemple, on obtient pour la fonction associée au Lift : $\Psi_{Lift}(x, y, z) = \frac{z \times |\mathcal{D}|}{x \times y}$. Rappelons que $|\mathcal{D}|$ correspond au nombre d'objets dans la base de données $\mathcal{D}$. Cette quantité est fixée lorsqu'on travaille sur une base de données particulière et nous ne la considérons pas comme une variable dans la fonction Ψ_M. Si on revient au cadre de Piatetsky-Shapiro [PS91], il est possible de reformuler les propriétés P2 et P3 (définition 15 à la page 27) à l'aide de la définition 18. P2 signifie que « la fonction Ψ_M est strictement croissante en z ». P3 revient à dire que « Ψ_M est strictement décroissante en x et en y ».

Nous avons vu en préambule qu'il existe un consensus autour de la notion de pertinence d'une règle d'association. Or limiter le nombre d'exceptions d'une règle introduit une dépendance entre $\mathcal{F}(XY)$ la fréquence de la règle et $\mathcal{F}(X)$ celle de sa prémisse puisque cela revient à imposer que la différence entre $\mathcal{F}(XY)$ et $\mathcal{F}(X)$ soit inférieure à δ. Cela signifie que les variables x et z de la fonction Ψ_M sont liées. La figure 4.1 schématise ce lien. Le domaine en gris montre le domaine de toutes les règles d'association puisque la fréquence d'une règle est nécessairement inférieure à la fréquence de sa prémisse. La bande hachurée symbolise l'influence du paramètre δ : la fréquence de la règle reste à une distance au plus δ de la fréquence de sa prémisse. La droite d'équation $x = \gamma$ représente la contrainte de fréquence minimale pour la prémisse de la règle.

[12] Nous pensons que c'est pour contourner cette difficulté que le cadre de Bayardo et Agrawal [BA99] ne traite que le cas des règles d'association dont la conclusion est fixée (donc la fréquence de la conclusion est également fixée).

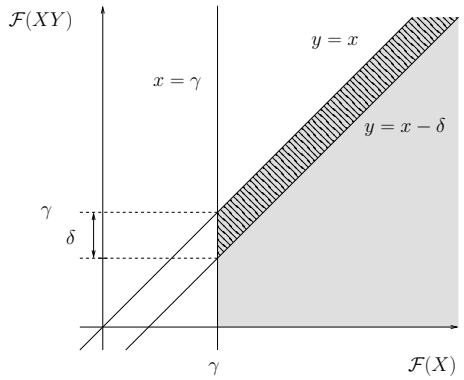

FIG. 4.1 – Dépendance induite par δ entre la fréquence d'une règle $\mathcal{F}(XY)$ et celle de sa prémisse $\mathcal{F}(X)$.

La définition 19 traduit cette dépendance entre les variables x et z dans la fonction associée à M.

Définition 19 (Fonction δ-dépendante) *Soit M une mesure d'intérêt. La fonction δ-dépendante $\Psi_{M,\delta}$ associée à M est définie comme suit : $\Psi_{M,\delta}(x, y) = \Psi_M(x, y, x - \delta)$.*

Le changement de variable $z = x - \delta$ revient à dire que la fréquence d'une règle et celle de sa prémisse sont suffisamment proches (car δ, le nombre d'exceptions, est faible) pour les représenter par une seule variable (x) dans la fonction δ-dépendante $\Psi_{M,\delta}$. En poursuivant l'exemple du Lift, on obtient : $\Psi_{Lift,\delta}(x, y) = \frac{(x-\delta)\times|\mathcal{D}|}{x\times y}$.

4.2.2 Déterminer les similarités intrinsèques aux mesures d'intérêt

Nous donnons ici un ensemble de propriétés communes à un grand nombre de mesures d'intérêt. Nous avons cherché à déterminer les propriétés minimales nécessaires à la découverte de similarités dans le comportement des mesures afin de définir le cadre le plus général possible. Ces propriétés vont servir à définir notre cadre dans la prochaine section.

La propriété P2' impose aux mesures d'intérêt la croissance avec la fréquence d'une règle.

Propriété 10 (P2' : croissance avec la fréquence d'une règle) *Soit M une mesure d'intérêt. Ψ_M est croissante en z.*

P2' est très proche de la propriété P2 définie par Piatetsky-Shapiro [PS91] (cf. définition 15 à la page 27). Par souci de généralité, P2' n'impose pas la croissance **stricte** d'une mesure avec la fréquence d'une règle, contrairement à P2. Par conséquent, toute mesure vérifiant P2 vérifie également P2' (la réciproque est fausse). Cependant en pratique, nous n'avons rencontré aucune mesure constante en la fréquence d'une règle.

La propriété P3' traduit la décroissance d'une mesure avec la fréquence de la conclusion d'une règle.

Propriété 11 (P3' : décroissance avec la fréquence de la conclusion) *Soit M une mesure d'intérêt. Ψ_M est décroissante en y.*

P3' reprend la deuxième partie de la propriété P3 de Piatetsky-Shapiro, sans toutefois imposer une décroissance **stricte**. De nouveau, si une mesure d'intérêt satisfait P3 alors elle satisfait aussi

P3'. Contrairement à P2, la relaxation de P3 en P3' permet d'agrandir le champ des mesures qui vérifient cette propriété. Ainsi, dans le tableau 2.1, on recense six mesures d'intérêt qui vérifient P3' mais pas P3 : le support, la confiance, le taux des exemples et contre-exemples, l'indice de Ganascia, le laplacien et la mesure de Sebag & Shoenauer. La propriété P3 donnée par Piatetsky-Shapiro est très restrictive et n'est vérifiée que par peu de mesures. À ce propos, Julien Blanchard [Bla05] énonce que :

> Piatetsky-Shapiro [PS91] considère aussi qu'un bon indice doit décroître avec n_b[13]. Cette condition est trop contraignante pour apparaître dans une définition générale des indices de règles [...] puisque certains indices ne dépendent pas de n_b.

Dans la section précédente, nous avons vu que la fonction $\Psi_{M,\delta}$ formalise le lien entre la fréquence de la prémisse et la fréquence de la règle. Nous pensons que celui-ci influence fortement le comportement d'une mesure. C'est pourquoi nous proposons une nouvelle propriété, nommée *propriété de croissance liée* et notée P4, qui capture ce lien. La propriété P4 impose qu'une mesure M soit croissante suivant la variable x de $\Psi_{M,\delta}$. Cela revient à considérer les mesures d'intérêt globalement croissantes avec la fréquence d'une règle et celle de sa prémisse.

Propriété 12 (P4 : croissance liée) *Soit M une mesure d'intérêt. $\Psi_{M,\delta}$ est croissante en x.*

Remarquons que si Ψ_M est croissante à la fois en x et en z, alors M vérifie nécessairement P4. Cependant, la réciproque est fausse. Par exemple, la confiance vérifie P4 (i.e., $\Psi_{Conf,\delta}(x,y) = 1-\frac{\delta}{x}$ est croissante) mais $\Psi_{Conf}(x,y,z) = \frac{z}{x}$ est décroissante en x. P4 ne donne pas des conditions de variation directement sur les variables intervenant dans le calcul de la mesure M mais sur une fonction associée à M après changement de variable. Nous pensons que P4 permet d'exprimer une caractéristique originale et importante d'une mesure : le comportement de M relativement à l'évolution conjointe des deux variables fréquence de la prémisse et nombre d'exceptions. Cette caractéristique n'apparaît pas dans le cadre de Piatetsky-Shapiro. Nous pensons que les propriétés P2 et P3 du cadre proposé par Piatetsky-Shapiro sont parfois contradictoires : il est imposé aux mesures d'intérêt de croître en la fréquence de la règle et de décroître en la fréquence de la prémisse alors que ces deux quantités restent proches l'une de l'autre dès lors que l'on borne le nombre d'exceptions d'une règle.

4.2.3 Les SBMs

Nous définissons maintenant l'ensemble des mesures de notre cadre : les SBMs. Nous verrons dans la section suivante comment ces mesures peuvent être simultanément optimisées et qu'elles se comportent toutes de manière similaire.

Définition 20 *Une mesure d'intérêt est une SBM si et seulement si elle vérifie les propriétés P2', P3' et P4.*

Rappelons que les tableaux 2.1 et 2.2 (pages 28 et 29) recensent les mesures d'intérêt les plus utilisées. Même si les propriétés P2', P3' et P4 peuvent sembler restrictives, il s'avère que parmi toutes ces mesures, seules quatre ne sont pas des SBMs. La Prevalence est croissante en la fréquence de la conclusion et ne satisfait donc pas P3'. Le Leverage et le Relative Risk vérifient

[13] Le nombre n_b correspond à la fréquence de la conclusion d'une règle d'association qui est notée $\mathcal{F}(Y)$ dans ce manuscrit.

bien P2' et P3' mais pas P4[14]. L'indice de Yule ne vérifie ni P2', ni P3', ni P4[15]. En guise d'exemple, il est prouvé que le Rule-Interest est une SBM à la section B.1 de l'annexe B.

L'ensemble des SBMs englobe donc un grand nombre de mesures usuelles. De par sa définition en intension, il est même infini. Cette caractéristique transparaît aussi dans le théorème 3 qui montre qu'en combinant des SBMs, on obtient toujours une SBM.

Théorème 3 *Toute combinaison linéaire à coefficients positifs de SBMs est une SBM.*

Preuve *Soient M_1, ...,M_k k SBMs, $(\alpha_1, \ldots, \alpha_k)$ un élément de $(\mathbb{R}^+)^k$ et β_1 un nombre réel positif. Montrons que $M = \alpha_1 M_1 + \ldots + \alpha_k M_k + \beta$ est une SBM. Tout d'abord, on a naturellement les deux égalités suivantes :*

$$\Psi_M(x, y, z) = \alpha_1 \Psi_{M_1}(x, y, z) + \ldots + \alpha_k \Psi_{M_k}(x, y, z) + \beta$$

$$\Psi_{M,\delta}(x, y) = \alpha_1 \Psi_{M_1,\delta}(x, y) + \ldots + \alpha_k \Psi_{M_k,\delta}(x, y) + \beta.$$

En effet, puisque $M_1(r)+M_2(r) = (M_1+M_2)(r)$, la fonction associée à la somme de deux mesures est la somme des fonctions associées à chacune des mesures. On peut faire le même raisonnement avec la fonction δ-dépendante de la somme de deux mesures et avec la multiplication par un réel. Les mesures M_1, ...,M_k étant des SBMs, elles vérifient les propriétés P2', P3' et P4. D'après P2' et P3', les fonctions Ψ_{M_1},..., Ψ_{M_k} sont croissantes en z et décroissantes en y. Donc leurs dérivées en z sont positives et leurs dérivées en y sont négatives. Or, la dérivée de $\Psi_M(x, y, z)$ en z (resp. en y) est égale à la somme coefficientée par $(\alpha_1, \ldots, \alpha_k)$ des dérivées de Ψ_{M_1},..., Ψ_{M_k}, elle reste donc positive (resp. négative) (puisque α_1,...,α_k sont positifs). Donc M vérifie P2' et P3'. Puisque d'après P4, les fonctions $\Psi_{M_1,\delta}$,..., $\Psi_{M_k,\delta}$ sont croissantes en x, une combinaison linéaire à coefficients positifs de ces fonctions reste croissante en x (la dérivée de $\Psi_{M,\delta}$ en x est la somme coefficientée des dérivées des $\Psi_{M_i,\delta}$ pour i allant de 1 à k, ces dérivées étant positives, on en déduit que la dérivée de $\Psi_{M,\delta}$ en x l'est aussi) donc M vérifie P4.

Le théorème 3 est utile pour construire de nouvelles SBMs ou pour prouver qu'une mesure existante est une SBM. Considérons par exemple la nouveauté [LFZ99] définie par $Nouveaute(r) = \frac{\mathcal{F}(XY) \times |\mathcal{D}| - \mathcal{F}(X) \times \mathcal{F}(Y)}{|\mathcal{D}|^2}$. Elle s'exprime en fonction du Rule-Interest de Piatetsky-Shapiro en quotientant par le nombre d'objets de la base de données : $Nouveaute = f(RuleInterest)$ avec $f(x) = \frac{x}{|\mathcal{D}|}$. Comme le Rule-Interest est une SBM, on en déduit immédiatement grâce au théorème 3 que la nouveauté en est une aussi.

4.2.4 Minoration simultanée des SBMs

Nous prouvons maintenant que toute SBM admet un minorant qui dépend des paramètres γ (fréquence minimale de la prémisse d'une règle), η (fréquence maximale de la conclusion) et δ (nombre d'exceptions maximal).

Théorème 4 (Minoration des SBMs) *Soit $r : X \rightarrow Y$ une règle d'association. Si $\mathcal{F}(X) \geq \gamma$, si $\mathcal{F}(Y) \leq \eta$ et si r admet moins de δ exceptions, alors pour toute SBM M, $M(r)$ est supérieure à $\Psi_{M,\delta}(\gamma, \eta)$.*

[14] Plus précisément, le Relative Risk vérifie P4 si et seulement si $|\mathcal{D}| \geq \mathcal{F}(Y) + \delta$, ce qui est généralement le cas. Cette condition est prouvée à la section B.2 de l'annexe B.

[15] À nouveau, dans le cas particulier où $|\mathcal{D}| \geq \mathcal{F}(Y) + \delta$, l'indice de Yule vérifie P2', P3' et P4.

Preuve *D'après P2', $\Psi_M(x, y, z)$ croît avec z. Puisque la règle r a moins de δ exceptions, $\mathcal{F}(XY) \geq \mathcal{F}(X) - \delta$ et par conséquent $\Psi_M(x, y, z) \geq \Psi_M(x, y, x - \delta) = \Psi_{M,\delta}(x, y)$. x est minoré par γ et y est majoré par η. Puisque $\Psi_{M,\delta}$ est croissante en x (d'après P4) et décroissante en y (c'est une conséquence de P3'), $\Psi_{M,\delta}(x, y)$ est minoré par $\Psi_{M,\delta}(\gamma, \eta)$.*

Les tableaux 4.1 et 4.2 explicitent les minorants $\Psi_{M,\delta}(\gamma, \eta)$ des SBMs définies dans les tableaux 2.1 et 2.2.

Le théorème 4 signifie que toute règle ayant une prémisse γ-fréquente, dont la conclusion est η-infréquente et dont le nombre d'exceptions est inférieur à δ a une qualité supérieure ou égale à $\Psi_{M,\delta}(\gamma, \eta)$. Comme ce résultat est valable pour toute mesure M, une conséquence immédiate de ce théorème est qu'une telle règle a une qualité suivant chaque mesure supérieure au minorant de cette mesure. On dispose ainsi d'un réservoir de règles de bonne qualité par rapport à l'ensemble des SBMs. La définition 21 les nomme *règles optimisées*.

Définition 21 (Règle optimisée) *Une règle d'association $r : X \rightarrow Y$ satisfaisant les conditions suivantes :*
- $\mathcal{F}(X) \geq \gamma$
- $\mathcal{F}(Y) \leq \eta$
- $\mathcal{F}(X) - \mathcal{F}(XY) \leq \delta$

est une règle optimisée. L'ensemble des règles optimisées est noté R_{op}.

Dans la suite, on appelle motif η-infréquent un motif dont la fréquence est inférieure à η, où η est un entier strictement positif fixé. La propriété 13 précise les contraintes de fréquence vérifiées par les règles optimisées. Ce résultat va permettre d'améliorer l'efficacité de l'extraction des règles optimisées informatives (cf. section 4.3.3).

Propriété 13 *Les règles optimisées satisfont les conditions suivantes :*

1. $\gamma - \delta \leq \mathcal{F}(XY) \leq \mathcal{F}(Y) \leq \eta$
2. $\gamma \leq \mathcal{F}(X) \leq \eta + \delta$

Preuve *Rappelons qu'une règle optimisée $r : X \rightarrow Y$ vérifie les trois conditions suivantes : $\mathcal{F}(X) \geq \gamma$, $\mathcal{F}(Y) \leq \eta$ et $\mathcal{F}(X) - \mathcal{F}(XY) \leq \delta$.*

1. *La troisième condition implique que $\mathcal{F}(XY) \geq \mathcal{F}(X) - \delta$ d'où $\mathcal{F}(XY) \geq \gamma - \delta$. La deuxième partie de l'inégalité est immédiate puisque $Y \subseteq XY$.*
2. *Toujours avec la troisième condition, on a $\mathcal{F}(X) \leq \mathcal{F}(XY) + \delta$ et on a vu en 1. que $\mathcal{F}(XY) \leq \eta$ donc on a $\mathcal{F}(X) \leq \eta + \delta$.*

La propriété 13 montre que l'optimisation simultanée des mesures de notre cadre revient à minorer et majorer les fréquences des différentes parties d'une règle d'association.

4.2.5 Comportement des minorants

Nous étudions dans cette section le comportement des minorants donnés par le théorème 4. Les courbes de la figure 4.2 illustrent le comportement des minorants de plusieurs SBMs en fonction des paramètres γ, η et δ. $|\mathcal{D}|$ est fixé à 499 car nous avons voulu nous placer dans le même contexte que les expérimentations de la section 4.5 menées sur des données réelles qui comportent 499 objets. Remarquons que les minorants des mesures de sensibilité et de Jaccard sont très proches l'un de l'autre (courbes G1, E1 et D1). On peut d'ailleurs vérifier que les

SBM	minorant
Support	$\frac{\gamma - \delta}{\lvert\mathcal{D}\rvert}$
Coverage	$\frac{\gamma}{\lvert\mathcal{D}\rvert}$
Confiance/Confidence	$1 - \frac{\delta}{\gamma}$
Laplace (k=2)	$\frac{\gamma - \delta + 1}{\gamma + 2}$
Sensibilité/Sensitivity/Recall	$\frac{\gamma - \delta}{\eta}$
Jaccard	$\frac{\gamma - \delta}{\eta + \delta}$
Intérêt/Lift	$\frac{\gamma - \delta}{\gamma} \times \frac{\lvert\mathcal{D}\rvert}{\eta}$
Information Gain	$\log\left(\frac{\gamma - \delta}{\gamma} \times \frac{\lvert\mathcal{D}\rvert}{\eta}\right)$
Taux des exemples et contre-exemples	$1 - \frac{\delta}{\gamma - \delta}$
Indice de Ganascia	$1 - \frac{2\delta}{\gamma}$
Moindre-contradiction	$\frac{\gamma - 2\delta}{\eta}$
Rule-Interest	$\gamma - \delta - \frac{\gamma\,\eta}{\lvert\mathcal{D}\rvert}$
Nouveauté/Novelty	$\frac{\gamma - \delta}{\lvert\mathcal{D}\rvert} - \frac{\gamma\,\eta}{\lvert\mathcal{D}\rvert^2}$
Added Value	$\frac{\gamma - \delta}{\gamma} - \frac{\eta}{\lvert\mathcal{D}\rvert}$
Indice de Loevinger/Certainty Factor	$\frac{\gamma \times (\lvert\mathcal{D}\rvert - \eta) - \delta \times \lvert\mathcal{D}\rvert}{\gamma \times (\lvert\mathcal{D}\rvert - \eta)}$
Coefficient de corrélation/ϕ-coefficient	$\frac{\gamma \times (\lvert\mathcal{D}\rvert - \eta) - \delta \times \lvert\mathcal{D}\rvert}{\sqrt{\gamma \times (\lvert\mathcal{D}\rvert - \gamma) \times \eta \times (\lvert\mathcal{D}\rvert - \eta)}}$
Sebag & Schoenauer	$\frac{\gamma - \delta}{\delta}$
Multiplicateur de cotes/Growth rate	$\frac{\gamma - \delta}{\delta} \times \frac{\lvert\mathcal{D}\rvert - \eta}{\eta}$
Conviction	$\frac{\lvert\mathcal{D}\rvert - \eta}{\lvert\mathcal{D}\rvert} \times \frac{\gamma}{\delta}$
Odds ratio/Rapport de cotes	$\frac{\gamma - \delta}{\delta} \times \frac{\lvert\mathcal{D}\rvert - \eta - \delta}{\eta - \gamma + \delta}$
Spécificité/Specificity	$\frac{\lvert\mathcal{D}\rvert - \eta - \delta}{\lvert\mathcal{D}\rvert - \gamma}$
Indice de Sokal et Michener/Success Rate	$1 + \frac{\gamma - 2\delta - \eta}{\lvert\mathcal{D}\rvert}$

Tab. 4.1 – Minorants des SBMs définies dans le tableau 2.1.

SBM	minorant
Indice de Roger et Tanimoto	$\frac{\lvert\mathcal{D}\rvert+\gamma-\eta-2\delta}{\lvert\mathcal{D}\rvert-\gamma+\eta+2\delta}$
Collective strength	$\frac{(\gamma-\delta)(\lvert\mathcal{D}\rvert-\eta-\delta)}{\gamma\eta+(\lvert\mathcal{D}\rvert-\gamma)(\lvert\mathcal{D}\rvert-\eta)} \times \frac{\lvert\mathcal{D}\rvert^2-\gamma\eta-(\lvert\mathcal{D}\rvert-\gamma)(\lvert\mathcal{D}\rvert-\eta)}{\eta+2\delta-\gamma}$
Indice de Dice	$\frac{\gamma-\delta}{\gamma+\frac{1}{2}(\eta-\gamma+2\delta)}$
Indice de Kulczynski	$\frac{\gamma-\delta}{2}\left(\frac{1}{\gamma}+\frac{1}{\eta}\right)$
Indice d'Ochiai	$\frac{\gamma-\delta}{\sqrt{\gamma\eta}}$
Contribution orientée au χ^2	$\frac{\gamma-\delta-\frac{\gamma\eta}{\lvert\mathcal{D}\rvert}}{\sqrt{\frac{\gamma\eta}{\lvert\mathcal{D}\rvert}}}$
Indice d'implication	$\frac{\delta-\frac{\gamma(\lvert\mathcal{D}\rvert-\eta)}{\lvert\mathcal{D}\rvert}}{\sqrt{\frac{\gamma(\lvert\mathcal{D}\rvert-\eta)}{\lvert\mathcal{D}\rvert}}}$
Kappa κ	$2\frac{\gamma(\lvert\mathcal{D}\rvert-\eta)-\lvert\mathcal{D}\rvert\delta}{\lvert\mathcal{D}\rvert\gamma+\lvert\mathcal{D}\rvert\eta-2\gamma\eta}$

TAB. 4.2 – Minorants des SBMs définies dans le tableau 2.2.

expressions qui définissent ces mesures (cf. tableau 2.1 à la page 28) diffèrent très peu. Les courbes G1 à G5 donnent les valeurs des minorants lorsque γ varie entre 10 et 150 et que $\eta = 200$ et $\delta = 5$. Les courbes E1 à E5 montrent les variations des minorants en fonction de η qui varie de 150 à 498 quand $\gamma = 100$ et $\delta = 5$. Enfin, pour les courbes D1 à D5, δ varie de 0 à 25, η est fixé à 200 et γ à 100.

On constate que la valeur des minorants augmente lorsque le seuil minimal de fréquence γ croît et que le nombre maximal d'exceptions δ et la fréquence maximale de la conclusion η décroissent. Ceci est en adéquation avec les intuitions concernant la qualité des règles données à la section 4.1. De plus, le théorème 5 prouve ces observations expérimentales.

Théorème 5 (Variations des minorants) *$\Psi_{M,\delta}(\gamma,\eta)$ est une fonction croissante en γ et décroissante en η et en δ.*

Preuve *D'après P4, $\Psi_{M,\delta}(x,y)$ est croissante donc $\Psi_{M,\delta}(\gamma,\eta)$ est croissante en γ.*
P3' implique que $\Psi_{M,\delta}(x,y)$ est décroissante en y donc $\Psi_{M,\delta}(\gamma,\eta)$ est décroissante en η.
D'après P2', $\Psi_M(x,y,z)$ est croissante en z donc si $\delta_1 \geq \delta_2$ alors $\Psi_M(x,y,x-\delta_2) \geq \Psi_M(x,y,x-\delta_1)$ d'où $\Psi_{M,\delta_2}(\gamma,\eta) \geq \Psi_{M,\delta_1}(\gamma,\eta)$.

Le théorème 5 signifie que tous les minorants se comportent globalement de la même manière. Nous en déduisons que les SBMs ont des comportements très proches en fonction des paramètres γ, η et δ. Ce résultat est en adéquation avec l'étude de Tan *et al.* [TKS02] (cf. section 2.3.3) qui montre, de façon expérimentale, que borner le support des règles induit des similitudes dans le comportement de certaines mesures.

Cela n'empêche pas que certaines mesures conservent au niveau local des spécificités qui nécessitent quelques finesses dans leur interprétation, nous en donnons maintenant quelques exemples. Par exemple, les minorants représentés sur la courbe G1 sont linéaires en γ alors que ceux de la courbe G2 sont de la forme $-\frac{1}{\gamma}$. Cela signifie que dans le premier cas, la valeur de la mesure va continuer à croître au même rythme quand la fréquence de la règle augmente alors que

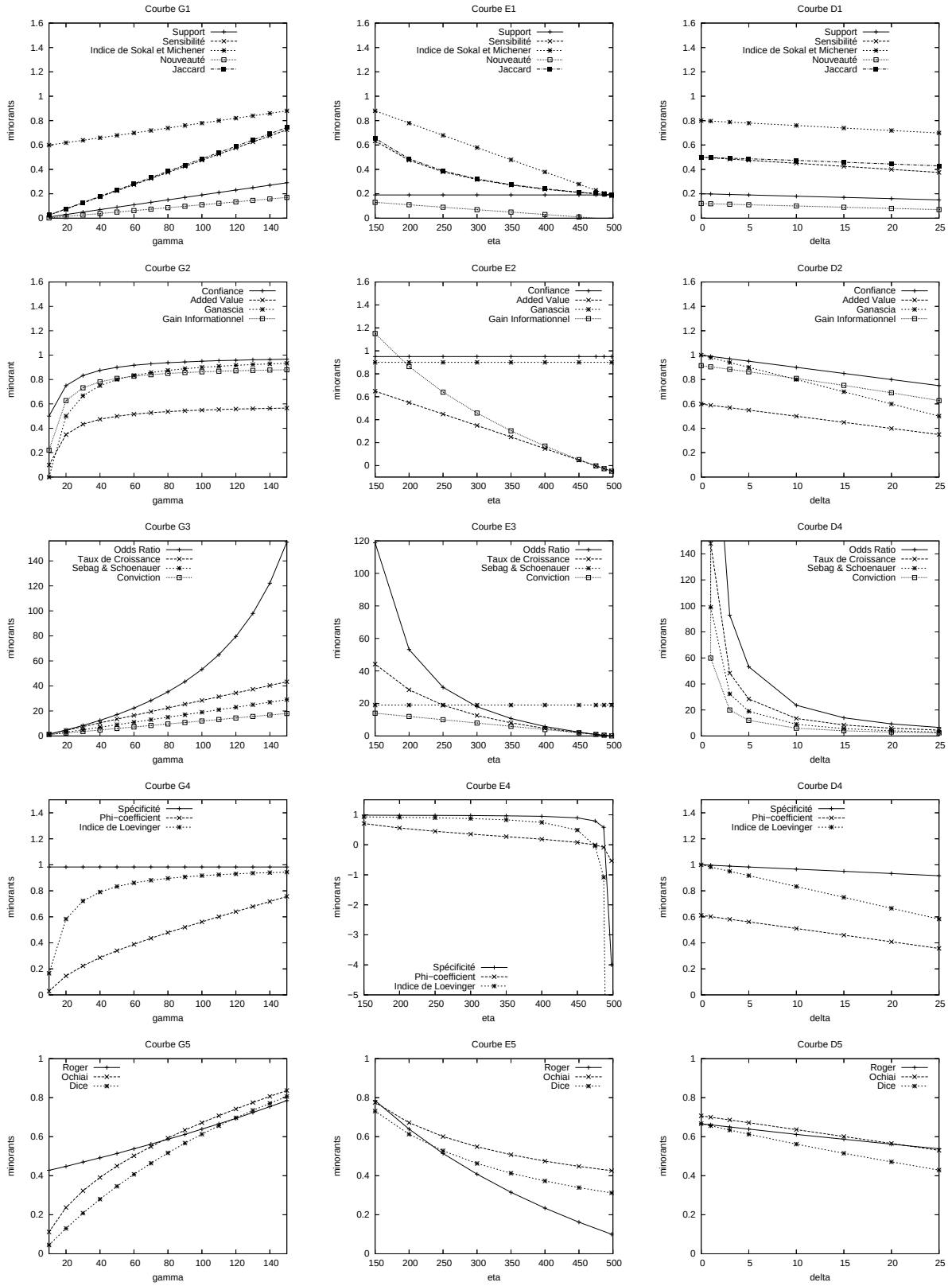

FIG. 4.2 – Comportement des minorants en fonction de γ, δ et η.

dans le deuxième cas, la croissance est rapide au départ et ralentit ensuite (lorsque la fréquence de la règle est très élevée, la valeur de la mesure n'augmente plus beaucoup). L'odds ratio a tendance à beaucoup privilégier les règles très fréquentes puisque la courbe G3 montre une croissance très rapide pour des valeurs élevées de γ. Des remarques semblables peuvent être formulées pour η et les courbes des minorants en fonction de ce paramètre montrent des décroissances plus ou moins rapides. Plusieurs minorants sont constants en η : ceux des mesures de support (E1), de confiance (E2), de Ganascia (E2) et de Sebag & Schoenauer (E3). Ce sont les minorants de mesures dans lesquelles la fréquence de la conclusion n'intervient pas et qui appartiennent à notre cadre grâce à la souplesse de la propriété P3' (voir section 4.2.2). Les minorants des SBMs représentés sur la courbe E4 montrent que celles-ci éliminent les règles dont la conclusion est très fréquente puisque la décroissance du minorant s'accentue fortement pour une fréquence de conlusion comprise entre 450 et 499. Excepté sur la courbe D3, tous les minorants sont linéaires en δ. En D3, on retrouve les minorants des mesures dont le dénominateur contient le nombre d'exceptions d'une règle et dont la valeur tend vers l'infini quand le nombre d'exceptions tend vers 0.

4.3 Identification et extraction des règles optimisées

Nous commençons par identifier la collection complète des règles qui admettent pour chaque SBM une valeur supérieure au minorant donné par le théorème 4. Nous montrons ensuite comment extraire efficacement un ensemble réduit de règles *informatives* dans le sens qu'elles véhiculent le plus d'information.

4.3.1 Ensemble des règles à valeurs garanties pour les SBMs

Le théorème 6 est la réciproque du théorème 4 : il montre que toute règle ayant, pour toute SBM, une valeur supérieure aux minorants donnés au théorème 4 satisfait des conditions de fréquence minimale, de fréquence maximale pour sa conclusion et a un nombre borné d'exceptions.

Théorème 6 (Exhaustivité) *Soit $r : X \rightarrow Y$ une règle d'association. Si pour toute SBM M, $M(r) \geq \Psi_{M,\delta}(\gamma, \eta)$ alors r est une règle optimisée.*

Preuve *Nous définissons $M_1(r) = \mathcal{F}(X)$, $M_2(r) = \dfrac{1}{\mathcal{F}(Y)}$ et $M_3(r) = \dfrac{1}{\mathcal{F}(X) - \mathcal{F}(XY)}$. Il est trivial de vérifier que M_1, M_2 et M_3 sont des SBMs. Par hypothèse, on a les inégalités suivantes : $M_1(r) \geq \Psi_{M_1,\delta}(\gamma, \eta) = \gamma$, $M_2(r) \geq \Psi_{M_2,\delta}(\gamma, \eta) = \dfrac{1}{\eta}$ et $M_3(r) \geq \Psi_{M_3,\delta}(\gamma, \eta) = \dfrac{1}{\delta}$ qui prouvent que $\mathcal{F}(X) \geq \gamma$, $\mathcal{F}(Y) \leq \eta$ et $\mathcal{F}(X) - \mathcal{F}(XY) \leq \delta$. Donc r est une règle optimisée.*

La combinaison des théorèmes 4 et 6 prouve l'égalité entre l'ensemble des règles qui optimisent toutes les SBMs et l'ensemble des règles optimisées. Ce résultat est un point clé pour l'extraction de toutes les règles ayant une valeur supérieure au minorant théorique $\Psi_{M,\delta}(\gamma, \eta)$ pour toute SBM car il suffit alors d'extraire les règles optimisées. Or, les règles optimisées satisfont des contraintes simples portant sur la fréquence des parties d'une règle qui facilitent leur extraction comme le montrent les deux sections suivantes.

4.3.2 Règles optimisées informatives

Nous avons introduit à la section 2.2 la notion de règle informative [PBTL99a]. Rappelons que ces règles sont caractérisées par une prémisse minimale et une conclusion maximale. Ce

sont donc des règles qui apportent un maximum d'information à partir de propriétés minimales. Techniquement, une règle informative est construite à partir d'un motif libre (cf. définition 8 à la page 18) qui forme sa prémisse et d'un fermé à partir duquel sa conclusion est déduite. Dans cette section, nous étendons cette construction aux règles optimisées. Par analogie avec les règles informatives, nous définissons les règles optimisées informatives comme suit.

Définition 22 (Règle optimisée informative) *Une règle $r : X \rightarrow Y$ de la forme :*
- *X est un motif γ-fréquent et libre*
- *Y est un motif η-infréquent*
- *$X \cap Y = \emptyset$*
- *XY est un motif fermé*
- *r admet moins de δ exceptions dans $\mathcal{D}$*

est appelée règle optimisée informative. L'ensemble des règles optimisées informatives est noté $Inform(R_{op})$.

Remarquons que XY n'est pas nécessairement égal à la fermeture du motif X. Ainsi, tout motif fermé qui contient X (et donc $h(X)$) tel que le nombre d'exceptions de r ne dépasse pas δ et tel que $\mathcal{F}(Y) \leq \eta$, permet de construire une règle optimisée informative. Dans le tableau 1 à la page 4, $a_2 \rightarrow a_3a_6a_8$ est une règle optimisée informative si l'on fixe $\gamma = 2$, $\delta = 1$ et $\eta = 3$. En effet, sa prémisse a_2 est un motif libre et $a_2a_3a_6a_8$ est un motif fermé (différent de $h(a_2)$) de fréquence 2 (on peut retrouver ces informations sur le treillis de la figure 1.3). De surcroît, sa conclusion $a_3a_6a_8$ est 3-infréquente et puisque $\mathcal{F}(a_2) = 3$, cette règle admet une exception.

La définition des règles optimisées informatives présente de nombreux avantages. Tout d'abord, elles sont *informatives* dans le sens que leur prémisse est minimale et leur conclusion est maximale i.e. elles apportent l'information maximale à partir d'hypothèses minimales. En outre, nous verrons au théorème 8 que ce sont les règles qui admettent les valeurs les plus élevées pour toutes les SBMs parmi les règles optimisées. Par ailleurs, il existe des algorithmes efficaces pour extraire les motifs libres et les motifs fermés (ces motifs sont à la base des représentations condensées) et l'extraction des règles informatives est possible là où l'extraction de toutes les règles ne l'est pas. Enfin, les règles optimisées informatives permettent la regénération de toutes les règles optimisées ; ce point est détaillé dans le paragraphe suivant.

Génération des règles optimisées C'est la procédure suivante qui permet de générer les règles optimisées à partir des règles optimisées informatives. Pour toute règle $r : X \rightarrow Y$ de $Inform(R_{op})$, nous construisons l'ensemble $\mathcal{R}_{gen}(r)$ des règles $r' : X' \rightarrow Y'$ vérifiant :

(a) $X \subseteq X' \subseteq h(X)$;
(b) $h(X'Y') = XY$;
(c) $X' \cap Y' = \emptyset$.

On note alors $\mathcal{R}_{gen}$ l'ensemble des règles générées par ce procédé :

$$\mathcal{R}_{gen} = \{ \bigcup_{r \in Inform(R_{op})} \mathcal{R}_{gen}(r) \}.$$

Toujours pour la règle $a_2 \rightarrow a_3a_6a_8$ du tableau 1, on obtient l'ensemble de règles suivant :

$$\mathcal{R}_{gen}(a_2 \rightarrow a_3a_6a_8) = \{a_2 \rightarrow a_6,\ a_2 \rightarrow a_8,\ a_2 \rightarrow a_3a_6,\ a_2 \rightarrow a_3a_8,\ a_2 \rightarrow a_6a_8,\ a_2a_3 \rightarrow a_6,\ a_2a_3 \rightarrow a_8,\ a_2a_3 \rightarrow a_6a_8\}.$$

Le théorème 7 montre que toutes les règles optimisées sont ainsi générées.

Théorème 7 *L'ensemble des règles optimisées informatives $Inform(R_{op})$ permet de regénérer toutes les règles optimisées.*

Preuve *Nous allons montrer que R_{op} est inclus dans $\mathcal{R}_{gen}$ i.e. que toutes les règles optimisées sont générées.*
Soit $r_o : X_o \rightarrow Y_o$ une règle optimisée. Soient X le plus grand motif libre contenu dans X_o et $Y = h(X_oY_o)\backslash X$. Considérons la règle $r : X \rightarrow Y$. Tout d'abord, il est évident que XY est un motif fermé et que $X \cap Y = \emptyset$. En outre, puisque X et X_o appartiennent à la même classe d'équivalence (d'après le ***(a)*** *de la procédure de génération), on a $\mathcal{F}(X) = \mathcal{F}(X_o) \geq \gamma$. De plus, $\mathcal{F}(XY) = \mathcal{F}(X_oY_o)$ donc r et r_o ont le même nombre d'exceptions qui est nécessairement inférieur à δ puisque r_o est optimisée. Par ailleurs, comme $X_o \cap Y_o = \emptyset$ (****(c)*** *dans la procédure de génération) et $X \subseteq X_o$, on a $Y_o \subseteq h(X_oY_o)\backslash X = Y$ et $\mathcal{F}(Y_o) \leq \eta$ par hypothèse donc $\mathcal{F}(Y) \leq \eta$. Par conséquent, $r : X \rightarrow Y$ est dans $Inform(R_{op})$, ce qui implique que r_o appartient à $\mathcal{R}_{gen}(r)$ et donc que r_o est générée par la procédure énoncée ci-avant.*

Remarquons que les ensembles R_{op} et $\mathcal{R}_{gen}$ ne sont pas égaux et qu'il est possible de produire des règles dont la conclusion ne vérifie pas la condition de fréquence imposée par la définition d'une règle optimisée (leur fréquence est supérieure à η donc trop élevée). Sur l'exemple précédent, les règles $a_2 \rightarrow a_8$ et $a_2a_3 \rightarrow a_8$ sont générées à partir de $a_2 \rightarrow a_3a_6a_8$ mais puisque $\mathcal{F}(a_8) = 4 > \eta = 3$, ce ne sont pas des règles optimisées. La connaissance de l'ensemble des motifs fermés avec leur fréquence suffirait pour éliminer les règles générées qui ne sont pas optimisées. La propriété 14 montre que toutes les règles générées vérifient de bonnes propriétés concernant la fréquence de leur prémisse et leur nombre d'exceptions, ce qui assure un support et une confiance supérieurs à des valeurs seuils.

Propriété 14 *Toute règle $r : X \rightarrow Y$ de $\mathcal{R}_{gen}$ a une prémisse γ-fréquente et admet moins de δ exceptions.*

Preuve *Il existe une règle optimisée informative $r' : X' \rightarrow Y'$ telle que $r \in \mathcal{R}_{gen}(r')$. Comme $X'Y'$ est fermé, on a $h(XY) = h(X'Y')$ et par conséquent, $\mathcal{F}(XY) = \mathcal{F}(X'Y')$. Puisque X et X' appartiennent à la même classe d'équivalence par construction, on a aussi $\mathcal{F}(X) = \mathcal{F}(X')$. On en déduit que $\mathcal{F}(X) \geq \gamma$ et que les deux règles ont le même nombre d'exceptions qui est inférieur à δ car r' est optimisée.*

La propriété 14 signifie que les règles générées qui ne sont pas optimisées ne le sont pas uniquement à cause de la fréquence de leur conclusion. Un point important est que ces règles ne sont pas « fausses » puisque leur nombre d'exceptions est limité. La figure 4.3 résume les positions relatives des ensembles de règles que nous venons d'évoquer. Le plus grand ensemble de règles représenté est celui des règles valides dans le sens classique support/confiance (cf. section 2.1 au chapitre 2). Puisque l'on fixe les paramètres γ, η et δ, les seuils de support et de confiance sont naturellement $min_{supp} = \Psi_{Support,\delta}(\gamma, \eta)$ et $min_{conf} = \Psi_{Confiance,\delta}(\gamma, \eta)$. Les règles optimisées (dans le rectangle avec des rayures horizontales) correspondent exactement aux règles valides dont la conclusion est η-infréquente et ayant moins de δ exceptions. Les règles optimisées informatives (dans le rectangle quadrillé) sont un sous-ensemble de celles-ci qui est aussi inclus dans les règles informatives au sens support/confiance (rectangle gris clair). L'ensemble $\mathcal{R}_{gen}$ est représenté par un rectangle gris foncé. Les règles générées à partir de $Inform(R_{op})$ contiennent à la fois les règles optimisées et des règles ayant une conclusion de fréquence supérieure à η.

Montrons à présent que les règles optimisées informatives sont celles qui ont, parmi les règles optimisées, les meilleures valeurs pour toutes les SBMs.

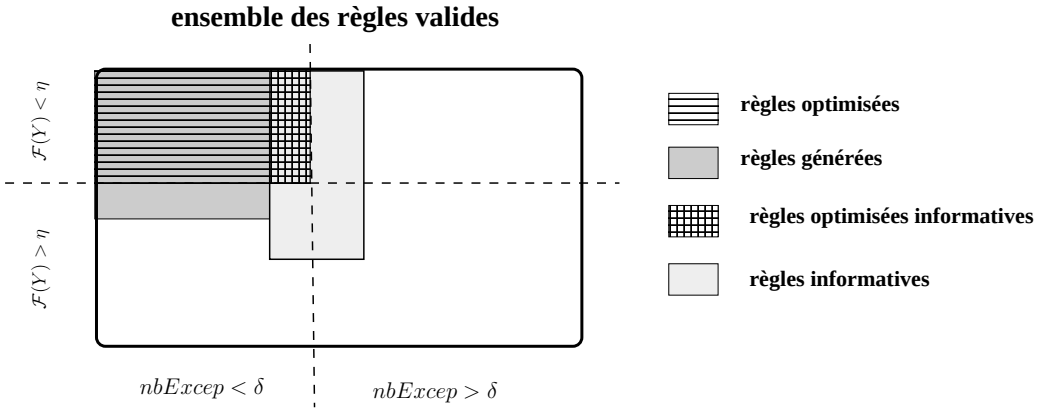

FIG. 4.3 – Position relative de divers ensembles de règles.

Théorème 8 *Soit $r : X \rightarrow Y$ une règle optimisée informative et $\mathcal{R}(r)$ l'ensemble des règles optimisées générées à partir de r. Alors pour toute SBM M et pour toute règle $r' : X' \rightarrow Y'$ de $\mathcal{R}(r)$, $M(r) \geq M(r')$.*

Preuve *Nous avons déjà vu dans la preuve précédente que les deux règles et leur prémisse ont la même fréquence. Par ailleurs, $Y' \subseteq Y$ donc $\mathcal{F}(Y') \geq \mathcal{F}(Y)$. Puisque les SBMs ne dépendent que de $\mathcal{F}(XY)$, $\mathcal{F}(X)$ et $\mathcal{F}(Y)$ et qu'elles sont décroissantes en cette dernière variable (cf. propriété P3') alors on a $M(r) \geq M(r')$.*

Après avoir défini et étudié les propriétés des règles optimisées informatives, nous donnons maintenant une méthode efficace de construction de ces règles.

4.3.3 Extraction

Cette section décrit notre méthode d'extraction des règles optimisées informatives, l'algorithme ROI (ROI pour RÈGLES OPTIMISÉES INFORMATIVES, cf. algorithme 3). L'ensemble des motifs libres $\mathcal{F}ree$ et celui des motifs fermés $\mathcal{C}losed$ sont donnés en entrée de cet algorithme[16]. L'ensemble $InformTemp$ contient les règles optimisées informatives déjà découvertes. On notera $E_{(\gamma_1,\gamma_2)}$ les motifs d'un ensemble E dont la fréquence est comprise entre γ_1 et γ_2 pour alléger l'écriture. À la ligne 1, on considère tous les motifs libres X dont la fréquence est comprise entre γ et $\eta + \delta$ (car la propriété 13 montre que la prémisse d'une règle optimisée satisfait ces contraintes). Pour chacun d'entre eux, on détermine les fermés de la forme $X \cup Y$ qui satisfont la propriété 13 et tels que la règle $X \rightarrow Y$ a moins de δ exceptions (ligne 2). À la ligne 3, on teste si la fréquence de la conclusion Y est inférieure à η (définition 21). Si toutes ces conditions sont réunies, la règle est une règle optimisée informative et elle est ajoutée à $InformTemp$ à la ligne 4.

[16] Rappelons qu'il existe des algorithmes efficaces d'extraction des représentations condensées, cf. section 1.2.

Entrée : $\mathcal{F}ree$ l'ensemble des motifs libres et $\mathcal{C}losed$ l'ensemble des motifs fermés de $\mathcal{D}$
Sortie : l'ensemble des règles optimisées informatives $Inform(R_{op})$
// Libres dont la fréquence est comprise entre γ et $\eta + \delta$
1 **pour tout** $X \in \mathcal{F}ree_{(\gamma,\eta+\delta)}$ **faire**
  // Fermés contenant X, avec une différence de fréquence inférieure à δ
2 **pour tout** $Z = XY \in \mathcal{C}losed_{(\gamma-\delta,\eta)}$ *tel que* $\mathcal{F}(X) - \mathcal{F}(Z) \leq \delta$ **faire**
    // Test sur la fréquence de la conclusion
3   **si** $\mathcal{F}(Y) \leq \eta$ **alors**
4      $InformTemp := InformTemp \cup \{X \rightarrow Y\}$
    fin
  fin
5 **retourner** $InformTemp$;
fin

Algorithme 3 : ROI

Théorème 9 *L'algorithme* ROI *extrait toutes les règles optimisées informatives.*

Preuve *Il est évident que toute règle extraite par* ROI *est par construction une règle optimisée informative. Donc l'algorithme* ROI *est correct.*
Les règles éliminées par ROI *sont en contradiction soit avec la définition 21 qui définit les règles optimisées informatives ; soit avec la propriété 13 qui est vérifiée par toute règle optimisée. Donc les règles éliminées ne sont pas des règles optimisées informatives et cela prouve que* ROI *est complet.*

Les expériences rapportées à la section 4.5 montrent qu'en pratique les règles optimisées informatives s'extraient facilement.

4.4 Un cas particulier : les règles de classification

Les règles de classification sont un cas particulier des règles d'association qui a de nombreuses applications. Nous montrons dans ce cas que notre cadre peut se simplifier et que nous pouvons établir de nouvelles propriétés [HC06b]. Nous prouvons notamment que les règles optimisées informatives forment une couverture des règles optimisées [HC06a].

4.4.1 Définition du cadre

Dans le cas des règles de classification, la conclusion d'une règle prend nécessairement sa valeur dans l'ensemble des attributs de classe que nous notons $\{c_1, c_2, \ldots, c_p\}$. La conclusion d'une règle et sa fréquence sont fixes par rapport à la base de données. La fréquence de la conclusion de la règle $X \rightarrow c_i$ est égale à $|\mathcal{D}_i|$ où $\mathcal{D}_i$ est la sous-base constituée des objets contenant la classe c_i dans $\mathcal{D}$ et il n'est plus nécessaire de la considérer comme variable dans l'expression d'une mesure d'intérêt. Nous pouvons ainsi éliminer la variable y dans les fonctions associées et δ-dépendantes. Reprenons l'exemple du Lift pour illustrer cette simplification. Les fonctions associée et δ-dépendante deviennent $\Psi_{Lift}(x, z) = \frac{z \times |\mathcal{D}|}{x \times |\mathcal{D}_i|}$ et $\Psi_{Lift,\delta}(x) = \frac{(x-\delta) \times |\mathcal{D}|}{x \times |\mathcal{D}_i|}$.

La suppression de y conduit naturellement à ôter la propriété P3' et le paramètre η du cadre. Une mesure d'intérêt est une SBM si elle vérifie P2' et P4. Le reste de la démarche (borner le nombre d'exceptions et imposer un seuil de fréquence minimal pour la prémisse de la règle) ne change pas. Il est alors possible de garantir l'existence d'un minorant pour toute mesure vérifiant P2' et P4 par un raisonnement similaire à celui des sections précédentes.

Théorème 10 *Soit $r : X \rightarrow c_i$ une règle de classification et M une mesure d'intérêt qui vérifie P2' et P4. Si $\mathcal{F}(X) \geq \gamma$ et si r admet moins de δ exceptions alors $M(r) \geq \Psi_{M,\delta}(\gamma)$.*

Le théorème 10 correspond au théorème 4 dans le cas des règles de classification et sa preuve est semblable à celle du théorème 4. Par rapport au cas général, un ajustement est nécessaire du fait qu'il existe plusieurs valeurs possibles pour $|\mathcal{D}_i|$. Lorsqu'une mesure d'intérêt est décroissante en la fréquence de la conclusion d'une règle, cette fréquence est minorée par $\mathtt{max}_{1 \leq i \leq p}|\mathcal{D}_i|$; sinon elle est minorée par $\mathtt{min}_{1 \leq i \leq p}|\mathcal{D}_i|$. Rappelons que les mesures d'intérêt sont en général décroissantes avec la fréquence de la conclusion, on peut citer en exemple toutes les SBMs. Revenons à l'exemple du Lift. $\Psi_{Lift}(x, z)$ étant croissante en z (P2'), $\Psi_{Lift,\delta}(x)$ étant croissante en x (P4) et la variable x étant minorée par γ, le théorème 10 implique que $Lift(r)$ est supérieur ou égal au minorant $\Psi_{Lift,\delta}(\gamma) = \frac{(\gamma-\delta)\times\mathcal{D}}{\gamma\times\mathtt{max}_{1 \leq i \leq n}|\mathcal{D}_i|}$. Une autre possibilité consiste à affecter un minorant, plus précis, qui dépend de $|\mathcal{D}_i|$ à chaque classe c_i, puisqu'en pratique le nombre de classe est souvent limité.

Remarquons que la simplification du cadre permet l'ajout de nouvelles mesures d'intérêt. Ainsi, la prevalence n'est pas une SBM dans le cas général puisqu'elle est croissante en la fréquence de la conclusion d'une règle. Cependant, elle vérifie P2' et P4 et elle appartient donc à notre cadre pour les règles de classification : $\Psi_{Prevalence}(x, z) = \Psi_{Prevalence,\delta}(x) = \frac{|\mathcal{D}_i|}{|\mathcal{D}|}$ d'où l'obtention d'un minorant égal à $\frac{\mathtt{min}_{1 \leq i \leq p}(|\mathcal{D}_i|)}{|\mathcal{D}|}$.

4.4.2 Couverture des règles optimisées

Par analogie avec le cas général (cf. définition 21), nous appelons règle de classification optimisée toute règle dont la prémisse est γ-fréquente et qui conclut sur un attribut de classe avec moins de δ exceptions. La définition des règles de classification optimisées informatives (définition 23) découle de la définition 22 du cas général :

Définition 23 (Règle de classification optimisée informative) *Une règle $r : X \rightarrow c_i$ de la forme :*

- *X est un motif γ-fréquent et libre*
- *r admet moins de δ exceptions dans $\mathcal{D}$*

est appelée règle de classification optimisée informative. L'ensemble des règles de classification optimisées informatives est noté $Couv(R)$.

La procédure de génération de règles décrite dans la section 4.3.2 peut être modifiée de la manière suivante : pour toute règle de classification optimisée informative $r : X \rightarrow c_i$, nous construisons l'ensemble $\mathcal{R}_{gen}(r)$ des règles $r' : X' \rightarrow c_i$ telles que $X \subseteq X' \subseteq h(X)$. On note à nouveau :

$$\mathcal{R}_{gen} = \{ \bigcup_{r \in Couv(R)} \mathcal{R}_{gen}(r) \}.$$

Cette procédure de génération de règles à partir des règles de classification optimisées informatives interdit la génération de règles non optimisées. Par conséquent, il est possible d'obtenir un résultat plus fort que dans le cas général : en effet, les règles de classification optimisées informatives forment une couverture des règles de classification optimisées. On obtient ainsi un résultat similaire à celui des couvertures de règles d'association avec les mesures de support et de confiance mais ce résultat s'étend à toutes les SBMs.

Théorème 11 *$Couv(R)$ forme une couverture des règles de classification optimisées.*

Preuve *Toute règle générée a la même fréquence et le même nombre d'exceptions que la règle à partir de laquelle elle a été générée donc elle est optimisée.*

Montrons maintenant que toutes règle optimisée est générée par la procédure décrite plus haut. Soit $r' : X' \rightarrow c_i$ *une règle optimisée. Notons* X *le plus grand libre inclus dans* X'*.* X *et* X' *sont dans la même classe d'équivalence donc* $\mathcal{F}(X) = \mathcal{F}(X')$ *et* $\mathcal{F}(Xc_i) = \mathcal{F}(X'c_i)$*. Puisque* r' *est optimisée, cela implique que* $\mathcal{F}(X) \geq \gamma$ *et* $\mathcal{F}(X) - \mathcal{F}(Xc_i) \leq \delta$ *donc* r *est une règle de classification optimisée informative et* $r' \in \mathcal{R}_{gen}(r)$*, ce qui prouve le résultat.*

Remarquons que dans le cas des règles de classification, le seul fait que X et X' appartiennent à la même classe d'équivalence suffit à assurer que r et r' ont exactement les mêmes valeurs pour toutes les SBMs.

4.4.3 Impact sur l'extraction

La simplification du cadre permet également d'améliorer la méthode d'extraction des règles informatives. Le nombre de classes étant en général limité, il est simple de stocker les extensions ou les fermetures des attributs de classe en mémoire et de les manipuler sans occasionner de problème de temps ou d'espace. Il n'est plus nécessaire de disposer de tous les motifs fermés pour déterminer les conclusions potentielles des règles. Il suffit de connaître les attributs de classe et leur fréquence. On évite ainsi deux étapes relativement coûteuses de l'algorithme Roi (cf. section 4.3.3) : celle qui vise à déterminer les fermés contenant une prémisse candidate à δ exceptions près (ligne 2 de l'algorithme 3), et celle qui consiste à retrouver la fréquence de la conclusion potentielle pour vérifier qu'elle ne dépasse pas γ (ligne 3 de l'algorithme 3). Une méthode simple et efficace pour extraire des règles de classification optimisées informatives consiste, pour chaque libre γ-fréquent, à déterminer les valeurs de classes c_i telles que la règle $X \rightarrow c_i$ ait moins de δ exceptions. L'algorithme 4 nommé ClaRminer [17] (ClaR pour Class Rules) fonctionne selon ce principe.

Entrée : $\mathcal{F}ree$ l'ensemble des motifs libres de $\mathcal{D}$, $\{c_1, \ldots, c_p\}$ l'ensemble des attributs de classe
Sortie : une couverture des règles de classification optimisées $Couv(R)$
`// Libres γ-fréquents`
pour tout $X \in \mathcal{F}ree_{(\gamma,|\mathcal{D}|)}$ **faire**
  pour tout *attribut de classe* c_i **faire**
    `// Vérifier que le nombre d'exceptions est inférieur à δ`
    si $\mathcal{F}(X) - \mathcal{F}(Xc_i) \leq \delta$ **alors**
      $CouvTemp := CouvTemp \cup \{X \rightarrow c_i\}$
    fin
  fin
  retourner $CouvTemp$;
fin

Algorithme 4 : ClaRminer

[17] Une variante de cet algorithme est donnée dans [HC06b]. Les deux algorithmes portent le même nom car leur différence est mineure : la deuxième version prend en entrée l'ensemble des motifs libres alors que la version initiale les calcule. Dans la première version, le calcul des libres étant combiné avec la détermination des règles, l'entrée de l'algorithme est la base de données.

4.5 Expériences

Ces expériences poursuivent deux objectifs. En premier lieu, nous souhaitons étudier le nombre de règles optimisées informatives extraites par l'algorithme Roi. Deuxièmement, nous voulons quantifier la qualité des règles extraites par rapport aux minorants théoriques dont nous avons prouvé l'existence (cf. théorème 4). Ces expériences sont menées sur des données réelles concernant des malades de l'hépatites récoltées à l'hopital universitaire de Chiba au Japon. Elles sont composées des examens de 499 patients décrits par 168 attributs. Ces données ont été utilisées dans les discovery challenges [HT05] associés aux conférences ECML/PKDD.

Nombre de règles optimisées informatives La figure 4.4 donne le nombre de règles optimisées informatives en fonction des paramètres γ, η et δ. La courbe de gauche représente (avec une échelle logarithmique pour l'ordonnée) le nombre de règles optimisées informatives quand γ varie de 40 à 150 (η est fixé à 200 et δ à 5). Bien sûr, le nombre de règles augmente de manière exponentielle lorsque γ diminue. Avec un seuil de fréquence à 80 (équivalent à un seuil relatif égal à 16%), on compte plus de 2 900 règles. En comparaison, il y a 126 828 règles dans la base réduite de Bastide *et al.* [BPT+00] avec le même seuil de fréquence et une confiance minimum égale au minorant $\Psi_{Conf,5}(80)$. Lorsque $\gamma = 100$, il n'y a que 248 règles optimisées informatives contre 43 015 dans la base de Bastide *et al.* Pour cette expérience, les temps d'extraction des règles optimisées informatives varient de 1 seconde pour $\gamma = 150$ à 4,5 minutes pour $\gamma = 60$ (la seule exception se produit avec $\gamma = 40$, l'extraction dure alors environ 1,5 heure). Les temps d'extraction sont également de l'ordre de quelques minutes pour les expériences suivantes, nous ne les détaillons donc pas.

La courbe du milieu donne le nombre de règles relativement à η qui varie entre 150 et 499 (γ vaut 100 et δ vaut 5). Le nombre de règles varie de 0 (quand $\eta = 150$) à 53 776 et augmente avec η. Remarquons que le seuil $\eta = 499$ revient à ne pas imposer de contrainte sur la fréquence de la conclusion d'une règle puisque la base contient 499 objets. Avec cette valeur, Roi extrait presque 54 000 règles, ce qui en fait une sortie inexploitable car beaucoup trop grande. Or, pour diminuer le nombre de règles, il n'est pas raisonnable d'augmenter γ sous peine de n'extraire que des trivialités. Diminuer le nombre d'exceptions risque également d'éliminer des règles intéressantes. Éliminer les règles ayant une conclusion trop fréquente, et qui peuvent être dues au hasard, nous apparaît ainsi comme une approche particulièrement pertinente.

La courbe de droite illustre l'augmentation du nombre de règles avec le nombre d'exceptions.

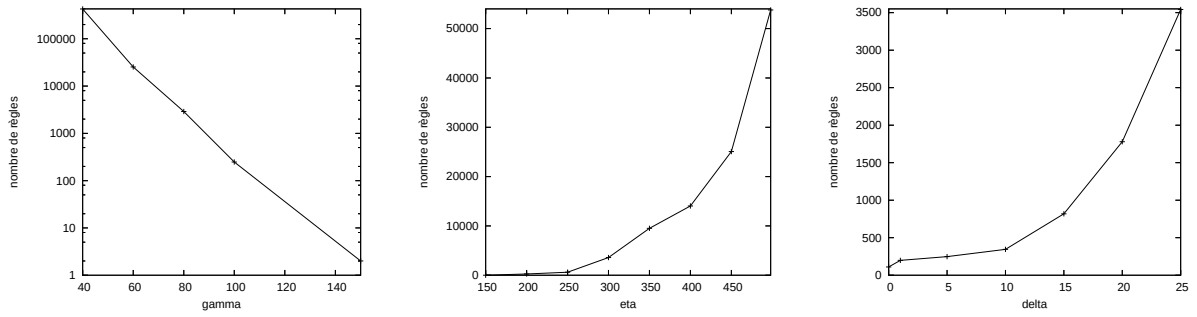

Fig. 4.4 – Nombre de règles optimisées informatives en fonction de γ, δ et η dans les données hépatites.

Qualité des règles Les courbes de la figure 4.5 représentent les minorants de quelques SBMs ainsi que leur valeur moyenne pour les règles optimisées informatives extraites de ce jeu de données. Sur les courbes G1, G2, et G3, γ varie de 40 à 150. E1, E2 et E3 donnent les moyennes en fonction de η qui varie entre 200 et 499. Sur les courbes D1 et D2, δ est compris entre 0 et 25. Comme l'Odds Ratio et le Taux de Croissance ne sont pas définis quand δ vaut zéro, les règles n'admettant aucune exception ne sont pas prises en compte dans le calcul de la moyenne de ces mesures (courbes G3, E3 et D3). Remarquons d'ailleurs que sur la courbe D3, δ n'est pas égal à zéro car l'Odds Ratio et le Taux de Croissance sont infinis et ne sont donc pas représentables.

Même si les écarts entre le minorant et la moyenne sont plus ou moins importants d'une mesure à l'autre, les valeurs moyennes sont globalement assez proches des minorants pour les mesures de Support, Sensibilité, l'indice de Sokal et Michener, la Confiance, l'Added Value et le Gain Informationnel. Sur la courbe G2, les moyennes de l'Added Value et du Gain Informationnel diminuent quand γ vaut 100 puis 150, contrairement aux autres mesures représentées. Nous n'avons pas d'explication à ce phénomène. Les valeurs moyennes de l'Odds Ratio et du Taux de Croissance (voir les courbes G3, E3 et D3) sont parfois éloignées du minorant mais nous pensons que cela est dû à leur amplitude (de 0 à l'infini). Finalement, les variations des minorants modélisent bien le comportement des moyennes sauf pour ces deux dernières mesures.

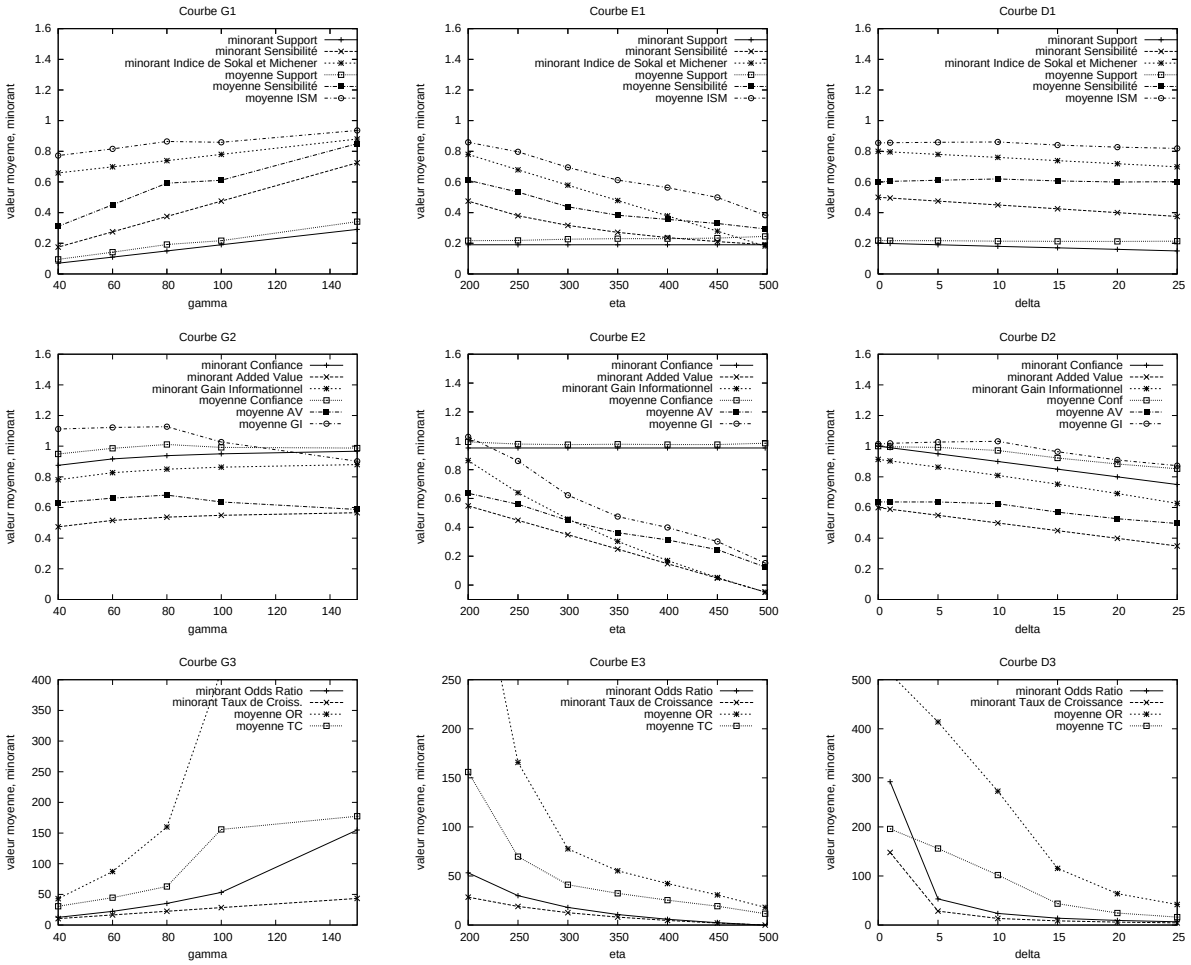

FIG. 4.5 – Comportement des mesures en fonction de γ, δ et η sur les règles d'association extraites des données hépatites.

4.6 Discussion et conclusion

Nous avons défini dans ce chapitre un cadre générique englobant un grand nombre de mesures d'intérêt appelées SBMs et permettant d'en donner une vue unifiée, comblant ainsi une des lacunes évoquées au chapitre 2. Nous avons mis en évidence les trois principaux paramètres impliqués dans le comportement de nombreuses mesures, ce qui facilite leur comparaison. Une analyse fine des SBMs a été menée, montrant des comportements voisins. Nous expliquons ce phénomène par le fait que, de façon implicite (cf. section 4.1), quelques principes relevant du bon sens ont été suivis par les concepteurs de mesures d'intérêt. Ce cheminement a conduit à définir des mesures qui sont, d'une certaine manière, redondantes les unes par rapport aux autres. Nous en concluons que le choix d'une mesures d'intérêt adéquate pourrait être considéré comme un problème secondaire pour l'utilisateur final.

Nous avons par ailleurs défini des règles optimisées informatives qui permettent de présenter une synthèse non redondante des règles qui optimisent simultanément toutes les SBMs. Ces règles sont un sous-ensemble des règles informatives de Bastide *et al.*. Nous avons fourni une méthode d'extraction des règles optimisées informatives et nos expériences ont montré qu'il est relativement facile d'obtenir ces règles en pratique. Rappelons que dans le cas particulier des règles de classification, les règles optimisées informatives forment une couverture des règles optimisées. Pour obtenir un résultat similaire dans le cas général, une première étape serait de proposer une nouvelle définition de la redondance d'une règle. En effet, qu'est-ce qu'une règle non redondante dans l'ensemble des règles optimisées ? Définir une règle non redondante comme étant à prémisse minimale et à conclusion maximale parmi les règles ayant les mêmes valeurs pour toutes les SBMs semble peu judicieux car ces règles sont peu nombreuses et cette définition ne respecte pas la structuration du treillis en classes d'équivalence.

Notons que dans [BA99], le nombre d'erreurs d'une règle est donné sous forme relative par la confiance, et non pas en terme d'occurrences dans la base de données. Or, c'est sans doute ce qui a limité le cadre proposé à un nombre restreint de mesures d'intérêt. L'explication « technique » est la suivante. Dans beaucoup de mesures, le nombre d'erreurs est exprimé sous l'une de ces deux formes : $\frac{\mathcal{F}(XY)}{\mathcal{F}(X)}$ (c'est le cas traité par Bayardo car exprimable avec la confiance) et $\mathcal{F}(X) - \mathcal{F}(XY)$ (c'est le cas supplémentaire qui peut être traité avec un nombre d'exceptions absolu car écrit comme une différence). Le cas « confiance » peut être traité avec un nombre d'exceptions absolu car en combinant une fréquence minimale pour la prémisse et une borne pour le nombre d'exceptions, il est possible de minorer le ratio confiance. Cependant la réciproque est fausse et minorer la confiance d'une règle ne permet pas de borner son nombre d'exceptions.

Chapitre 5

De la découverte de motifs au calcul des traverses minimales d'un hypergraphe

Sommaire

Le calcul des traverses minimales d'un hypergraphe est un problème algorithmique central ayant de nombreuses applications. En logique, ce problème est équivalent à la dualisation de formules booléennes [FK96, GK99] ; en data mining, il est lié entre autres à l'extraction de motifs fréquents [GKMT97] ou émergents [BMR03] ; il trouve aussi des applications en machine learning [EG02] (classification non-supervisée [HKKM98]), en biologie (reconstruction phylogénétique [Dam06]), pour la modélisation de réseaux de téléphonie mobile [SS98] ou encore les systèmes distribués [GMB85]. Nous proposons dans ce chapitre une approche originale pour déterminer les traverses minimales d'un hypergraphe [HBC07]. Cette approche tire parti des méthodes de découverte de motifs et plus précisément de la méthode d'extraction de motifs δ-libres proposée au chapitre 3.

La section 5.1 donne quelques définitions essentielles sur les hypergraphes et situe notre approche par rapport aux travaux antérieurs sur les traverses minimales. À la section 5.2, nous relions le problème des traverses minimales au cadre des représentations condensées. La section 5.3 détaille MTminer, notre algorithme de calcul des traverses minimales. Enfin, la dernière section est consacrée aux expérimentations montrant l'efficacité de notre approche.

5.1 Introduction

5.1.1 Définitions préliminaires

Un hypergraphe $\mathcal{H}$ est défini par une paire $(\mathcal{V}, \mathcal{E})$ où $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ est un ensemble de sommets et $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ un ensemble de parties non vides de $\mathcal{V}$ telles que

$$\bigcup_{1 \leq i \leq m} e_i = \mathcal{V}.$$

Les éléments de $\mathcal{E}$ sont appelés des *hyperarêtes*. Un hypergraphe partiel est la restriction de $\mathcal{H}$ à un sous-ensemble d'hyperarêtes $\mathcal{E}'$ inclus dans $\mathcal{E}$ et aux sommets contenus dans ces hyperarêtes. Un hypergraphe peut, de même qu'un graphe, être représenté par un dessin ou par une matrice d'adjacence. La figure 5.1 représente un hypergraphe composé de 8 sommets et de 6 hyperarêtes et la matrice d'adjacence correspondante est donnée au tableau 5.1. Ce dernier contient les mêmes valeurs que le tableau 1 de la page 4 qui représente le jeu de données servant d'exemple dans ce mémoire.

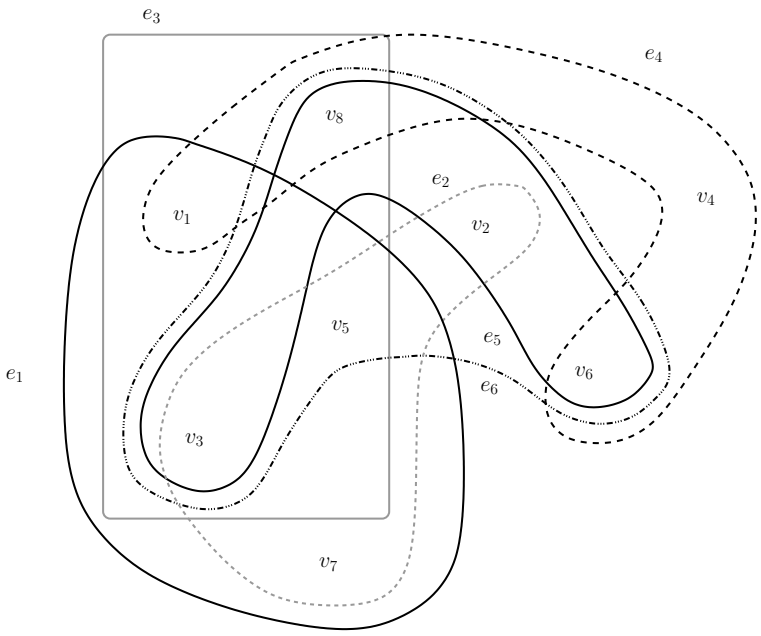

FIG. 5.1 – Un exemple d'hypergraphe.

Un ensemble $T \subseteq \mathcal{V}$ de sommets est un *transversal* si il intersecte toutes les hyperarêtes de $\mathcal{H}$. Dans l'hypergraphe donné du tableau 5.1, l'ensemble $v_1v_2v_3$ est un transversal car il recouvre les six hyperarêtes de $\mathcal{H}$. On notera $Tr(\mathcal{H})$ l'ensemble des transversaux de $\mathcal{H}$:

$$Tr(\mathcal{H}) = \{T \subseteq \mathcal{V} | \forall i \in \{1, 2, ..., m\},\ e_i \cap T \neq \emptyset\}.$$

T est un *transversal minimal* ou une *traverse minimale* s'il est de plus minimal au sens de l'inclusion i.e. aucun des sous-ensembles de T n'est un transversal. Par exemple, le transversal $v_1v_2v_3$ n'est pas minimal puisque v_1v_2 et v_1v_3 sont eux-mêmes des transversaux. Par contre, v_1v_2 et v_1v_3 sont des transversaux minimaux car aucun des sommets v_1, v_2 et v_3 ne constitue un transversal. L'ensemble des traverses minimales est noté $MinTr(\mathcal{H})$. $(\mathcal{V}, MinTr(\mathcal{H}))$ constitue un

Hyperarêtes \ Sommets	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
e_1	1	0	1	0	1	0	1	0
e_2	0	1	1	0	1	0	1	0
e_3	1	0	1	0	1	0	0	1
e_4	1	0	0	1	0	1	0	1
e_5	0	1	1	0	0	1	0	1
e_6	0	1	1	0	1	1	0	1

TAB. 5.1 – Matrice d'adjacence représentant l'hypergraphe de la figure 5.1.

hypergraphe appelé hypergraphe transversal [Ber89, p. 43] de $\mathcal{H}$. On note $t(\mathcal{H}) = \max_{T \in MinTr(\mathcal{H})} |T|$ le nombre maximal de sommets de la ou des plus grande(s) traverse(s) minimale(s). Toujours sur le même exemple, $v_1v_6v_7$ et $v_2v_4v_5$ sont les seules traverses minimales qui contiennent plus de 2 sommets donc on a $t(\mathcal{H}) = 3$.

5.1.2 État de l'art

Algorithmes de calcul des traverses minimales Le nombre de traverses minimales dans un hypergraphe $\mathcal{H}$ peut être exponentiel en $|\mathcal{H}| = n \times m$, la taille de $\mathcal{H}$. L'existence d'un algorithme résolvant ce problème avec une complexité polynomiale en $|\mathcal{H}|$ est donc impossible [18]. La question du calcul des traverses minimales d'un hypergraphe en un temps polynomial en la taille de la sortie est un problème ouvert. Le problème de décision associé est co-NP mais sa complexité exacte n'est pas connue [EG95].

Chronologiquement, le premier algorithme de calcul des traverses minimales est celui de Berge [Ber89, p. 52]. Cet algorithme commence par déterminer les traverses minimales d'une seule hyperarête (i.e. chacun des sommets de l'hyperarête), puis ajoute les autres hyperarêtes l'une après l'autre tout en mettant à jour les traverses minimales de l'hypergraphe partiel constitué par chaque ajout. En pratique, il n'est pas utilisable sur de gros hypergraphes [KS05, Rio05]. La dernière décennie a vu apparaître de nombreux algorithmes dédiés au calcul des traverses minimales. Ceux de Dong et Li [DL05], de Bailey *et al.* [BMR03] et de Kavvadias et Stavropoulos [KS05] ont en commun d'être des améliorations de l'algorithme initialement proposé par Berge. Dans [Hag07], Hagen démontre qu'aucun de ces algorithmes n'est polynomial en la taille de l'entrée et de la sortie [19]. En 1996, Fredman et Khachiyan [FK96] proposent un algorithme avec une complexité en temps de $P(n) + s^{\log(s)}$ où n est le nombre de sommets de l'hypergraphe donné en entrée, P est un polynôme et s est une combinaison de la taille de l'entrée et de la sortie. C'est l'algorithme qui détient actuellement la meilleure complexité. Cependant, Kavvadias et Stavropoulos ont comparé leur méthode [KS05] avec une implémentation de cet algorithme [BEGK03] et ont montré qu'il est moins efficace que le leur dans de nombreux cas pratiques. Enfin il existe des travaux dans lesquels on borne la taille de l'hypergraphe, plus précisément son nombre d'hyperarêtes [Wah04, KBEG05, Dam06]. Nous verrons que contrairement à ces derniers travaux, notre approche ne nécessite aucune hypothèse sur l'instance donnée en

[18] C'est pourquoi on s'intéresse plutôt à savoir si les algorithmes de calcul des traverses minimales sont polynomiaux en la taille de l'entrée et de la sortie (le nombre de traverses minimales).

[19] Les complexités en temps de ces algorithmes n'ont pas été données par les auteurs et ne sont pas connues à ce jour.

entrée.

Liens avec le data mining Beaucoup de problématiques issues de la fouille de données reposent sur le calcul des traverses minimales. Dans [MT97], Mannila et Toivonen énoncent un théorème qui relie les bordures d'une théorie (leur définition est donnée à la page 13) aux traverses minimales. Soit S l'ensemble des motifs vérifiant une contrainte anti-monotone q et $\mathcal{H} = (\mathcal{A}, \overline{Bd^+(S)})$ l'hypergraphe formé des complémentaires (par rapport à l'ensemble des attributs) des motifs de la bordure positive de S. Le théorème 12 [MT97] indique que les traverses minimales de $\mathcal{H}$ sont égales aux motifs de la bordure négative de S.

Théorème 12

$$MinTr((\mathcal{A}, \overline{Bd^+(S)})) = Bd^-(S)$$

Certains algorithmes d'extraction de motifs utilisent le théorème 12 comme test d'arrêt. Considérons l'algorithme d'extraction de motifs fréquents de Gunopulos *et al.* [GKM$^+$03]. Il effectue un parcours en profondeur de l'espace de recherche en spécialisant un attribut tant qu'il est fréquent. On est assuré d'avoir construit un motif fréquent maximal dès lors qu'il est impossible d'ajouter un attribut à ce motif sans le rendre infréquent. Ce motif est alors ajouté à une bordure positive temporaire. Pour savoir si l'algorithme a terminé ou le cas échéant, à partir de quel motif fréquent il faut réitérer cette procédure, le calcul des traverses minimales des complémentaires de la bordure positive temporaire est effectué. On teste alors si les motifs obtenus sont fréquents : si au moins l'un d'entre eux l'est, le calcul de la bordure positive continue, sinon l'agorithme s'arrrête. Bien qu'il utilise le même théorème, l'algorithme de Satoh et Uno [SU03] fonctionne différemment. Le calcul des traverses et la détermination de motifs fréquents sont simultanés alors que ces deux étapes sont distinctes dans la méthode précédente. Par ailleurs, Bailey *et al.* [BMR03] montrent qu'il existe un lien entre les motifs émergents et les traverses minimales. Nous reviendrons sur la problématique d'extraction des motifs fréquents suivant un parcours en profondeur au chapitre 7.

Dans [GKM$^+$03], il est énoncé qu'il est possible d'utiliser un algorithme par niveaux de type « extraction de motifs » pour calculer les traverses minimales, mais cette proposition n'est pas mise en œuvre. C'est justement l'approche que nous mettons en place dans la section 5.3. Nous verrons à la section 5.4 qu'elle est efficace et dans le chapitre 7, que le théorème 12 fournit un cas pratique d'utilisation de MTMINER. La section suivante décrit les grandes lignes de notre approche.

5.1.3 Principe de notre approche

Notre démarche consiste à exploiter le savoir-faire sur l'extraction de motifs pour résoudre un problème sur les hypergraphes. Plus précisément, nous réutilisons le principe des algorithmes par niveaux pour déterminer les traverses minimales d'un hypergraphe. La figure 5.2 résume le fonctionnement de notre approche. Celle-ci repose sur le fait que les bases de données et les hypergraphes peuvent se représenter de la même manière : sous forme de matrice booléenne. Les ensembles de sommets correspondent aux motifs et les hyperarêtes aux objets. Grâce à la définition d'une connexion de Galois appropriée reliant les ensembles de sommets et les ensembles d'hyperarêtes, nous établissons un parallèle entre l'extraction de motifs et le calcul de traverses minimales. L'extension de cette nouvelle connexion permet de définir des classes d'équivalence de façon analogue aux classes d'équivalence de fréquence utilisées en fouille de données (cf.

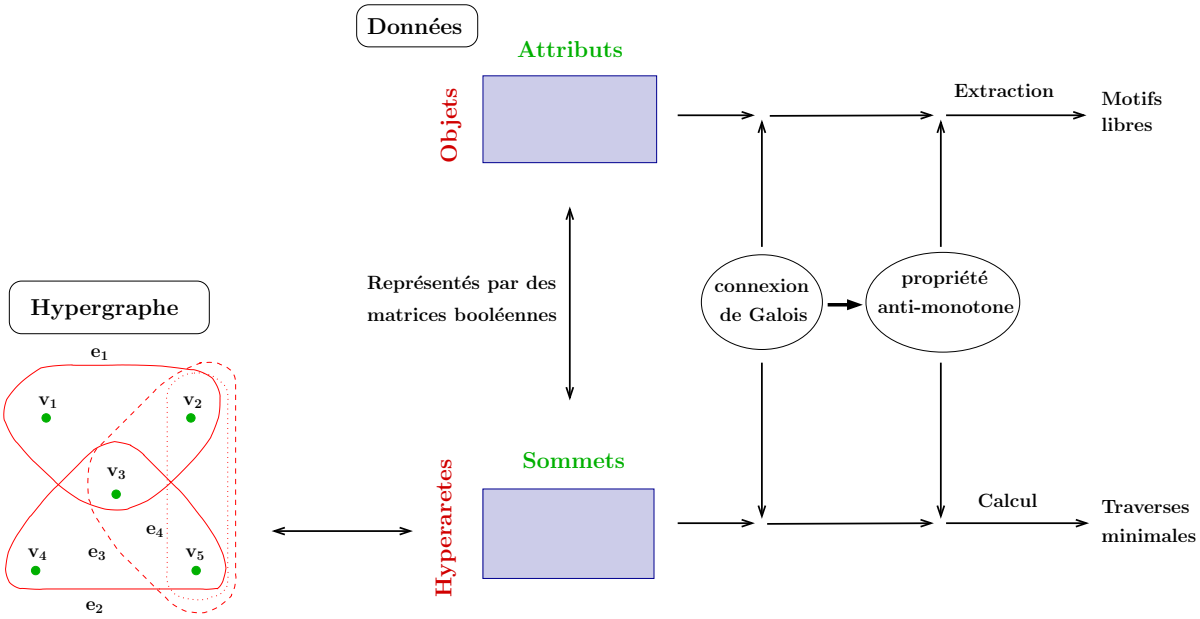

FIG. 5.2 – Principe général de notre approche pour calculer les traverses minimales.

section 1.1.2 et 1.2.1). Dans le cas des hypergraphes, les classes d'équivalence regroupent les ensembles de sommets qui recouvrent les mêmes hyperarêtes de $\mathcal{H}$. La taille de l'extension ou fréquence, qui compte le nombre d'occurrences d'un motif en fouile de données, correspond ici au nombre d'hyperarêtes non recouvertes par un ensemble de sommets. Les transversaux de $\mathcal{H}$ sont donc les ensembles de sommets ayant une extension vide (ils sont tous dans la même classe d'équivalence). Nous redéfinissons la notion de liberté (section 1.2.2) : les ensembles de sommets minimaux dans une classe d'équivalence (appelés *générateurs minimaux*) correspondent aux motifs libres d'une base de données. En combinant ces deux propriétés, nous montrons que les traverses minimales de $\mathcal{H}$ sont les générateurs minimaux d'extension vide.

Le cadre des représentations condensées étant transposable aux hypergraphes, nous réutilisons conjointement le principe des algorithmes par niveaux et les représentations condensées en nous inspirant de l'algorithme FTMINER présenté au chapitre 3. Notre algorithme évite de générer tous les ensembles de sommets en se concentrant sur les générateurs minimaux. Cette propriété de minimalité étant à nouveau anti-monotone, les critères d'élagage efficaces liés à l'anti-monotonie peuvent être réutilisés. De même que la fréquence, la taille de l'extension diminue lors de la spécialisation des ensembles de sommets. L'algorithme s'arrête quand il a trouvé un générateur minimal d'extension vide i.e. une traverse minimale.

Contrairement aux approches développées dans [GKM+03] et [SU03] qui utilisent les traverses minimales comme un outil mais dont l'objectif reste l'extraction de motifs, nous montrons que l'ECBD peut fournir des solutions efficaces à des problèmes célèbres pour leur difficulté algorithmique.

5.2 Plongement du problème des traverses minimales dans le cadre des représentations condensées

Cette section décrit le cœur de notre approche. Nous commençons par définir une nouvelle connexion de Galois nécessaire pour caractériser les traverses minimales avec l'extension. Notons que dans ce chapitre, les ensembles $2^{\mathcal{V}}$ et $2^{\mathcal{E}}$ représentent les ensembles de sommets et d'hyperarêtes, ils sont ordonnés par inclusion $\subseteq$.

5.2.1 Une nouvelle connexion de Galois

Nous définissons ici les opérateurs de Galois associés à un hypergraphe.

Définition 24 (Opérateurs associés à un hypergraphe) *Les opérateurs $f_{\mathcal{H}}$ et $g_{\mathcal{H}}$ associés à l'hypergraphe $\mathcal{H}$ sont définis de la manière suivante :*

$$\forall E \in 2^{\mathcal{E}},\ f_{\mathcal{H}}(E) = \{v \in \mathcal{V} |\ \forall e \in E,\ v \notin e\}$$

$$\forall V \in 2^{\mathcal{V}},\ g_{\mathcal{H}}(V) = \{e \in \mathcal{E} |\ \forall v \in V,\ v \notin e\}$$

Pour un ensemble E d'hyperarêtes, $f_{\mathcal{H}}(E)$ correspond aux sommets qui n'appartiennent à aucune hyperarête de E. Pour un ensemble V de sommets, $g_{\mathcal{H}}(V)$ donne les hyperarêtes ne contenant aucun sommet de V. De manière plus intuitive, cela signifie que V est un transversal de l'hypergraphe partiel constitué des hyperarêtes de $\mathcal{E} \backslash g_{\mathcal{H}}(V)$. Dans l'hypergraphe donné par le tableau 5.1, $g_{\mathcal{H}}(v_2v_4)$ vaut e_1e_3 car e_1 et e_3 sont les seules hyperarêtes de $\mathcal{H}$ ne contenant ni v_2 ni v_4. $f_{\mathcal{H}}(e_4e_5)$ est égal à v_5v_7 puisque v_5 et v_7 sont les seuls sommets de $\mathcal{V}$ n'appartenant ni à e_4 ni à e_5.

Théorème 13 (Connexion de Galois associée à un hypergraphe) *Le couple $(f_{\mathcal{H}}, g_{\mathcal{H}})$ établit une connexion de Galois entre $(2^{\mathcal{E}},\subseteq)$ et $(2^{\mathcal{V}},\subseteq)$.*

Preuve *Soient $V_1, V_2 \subseteq \mathcal{V}$ tels que $V_1 \subseteq V_2$ et $E_1, E_2 \subseteq \mathcal{E}$ tels que $E_1 \subseteq E_2$. Les hyperarêtes de $g_{\mathcal{H}}(V_2)$ ne contiennent aucun sommet de V_1 puisque $V_1 \subseteq V_2$ d'où $g_{\mathcal{H}}(V_2) \subseteq g_{\mathcal{H}}(V_1)$, ce qui prouve que $g_{\mathcal{H}}$ est décroissante. On montre de même que $f_{\mathcal{H}}$ est décroissante.*

Par définition, les sommets de V_1 n'appartiennent pas aux hyperarêtes de $g_{\mathcal{H}}(V_1)$ donc $V_1 \subseteq f_{\mathcal{H}} \circ g_{\mathcal{H}}(V_1)$. Cela montre que $f_{\mathcal{H}} \circ g_{\mathcal{H}}$ est extensive. On démontre de la même manière que $g_{\mathcal{H}} \circ f_{\mathcal{H}}$ est extensive également.

Par analogie avec le vocabulaire de fouille de données, nous appellerons $g_{\mathcal{H}}$ l'extension et $f_{\mathcal{H}}$ l'intension.

5.2.2 Identifier les traverses minimales avec l'extension

Nous donnons maintenant une caractérisation des traverses minimales en fonction de l'extension $g_{\mathcal{H}}$ de la définition 24. Pour faciliter la lecture, le tableau 5.2 récapitule les termes associés aux trois domaines abordés dans cette section : la fouille de données, les connexions de Galois et les hypergraphes.

Classes d'équivalence Un point clé de notre approche est la relation d'équivalence $\sim$ induite par le fait d'avoir la même extension : $\forall V, V' \subseteq \mathcal{V}, V \sim V' \Leftrightarrow g_{\mathcal{H}}(V) = g_{\mathcal{H}}(V')$. Cette relation

Fouille de données	Connexions de Galois	Hypergraphes
la connexion $(f_{\mathcal{D}}, g_{\mathcal{D}})$	une connexion de Galois (f, g)	la connexion $(f_{\mathcal{H}}, g_{\mathcal{H}})$
$(2^{\mathcal{O}}, \subseteq)$ les motifs d'objets*	l'ensemble partiellement ordonné $(A, \geq_1)$	$(2^{\mathcal{E}}, \subseteq)$ les ensembles d'hyperarêtes*
$(2^{\mathcal{A}}, \subseteq)$ les motifs d'attributs*	l'ensemble partiellement ordonné $(B, \geq_2)$	$(2^{\mathcal{V}}, \subseteq)$ les ensembles de sommets*
$g_{\mathcal{D}}$: les objets contenant un motif d'attributs donné	g : l'extension	$g_{\mathcal{H}}$: les hyperarêtes non recouvertes par un ensemble de sommets donné
les motifs d'attibuts apparaissant dans les mêmes objets (ou ayant la même fermeture)	une classe d'équivalence	les ensembles de sommets recouvrant les mêmes hyperarêtes
les motifs absents de $\mathcal{D}$	les éléments de B avec une extension vide	les transversaux de $\mathcal{H}$
les motifs libres	les éléments minimaux des classes d'équivalence	les traverses minimales hypergraphe partiel
la bordure négative des motifs de $\mathcal{D}$	les éléments minimaux de la classe d'équivalence avec une extension vide	les traverses minimales de $\mathcal{H}$

* ordonnés par inclusion

TAB. 5.2 – Correspondances des termes entre les bases de données, les connexions de Galois et les hypergraphes.

permet de définir des classes d'équivalence (cf. section 1.1.2) en regroupant tous les sommets qui recouvrent exactement les mêmes hyperarêtes de $\mathcal{H}$. Appliquée à la connexion $(f_{\mathcal{H}}, g_{\mathcal{H}})$, la relation $\sim$ va permettre de restreindre l'espace de recherche à un nombre réduit d'éléments bien choisis au lieu de parcourir la totalité des ensembles de sommets constituant l'espace de recherche.

Définition 25 *La classe d'équivalence d'un ensemble de sommets $V \subseteq \mathcal{V}$ est notée $\mathcal{R}_{g_{\mathcal{H}}}(V)$, elle est définie comme suit :*

$$\mathcal{R}_{g_{\mathcal{H}}}(V) = \{V' \in \mathcal{V} | g_{\mathcal{H}}(V') = g_{\mathcal{H}}(V)\}.$$

La classe d'équivalence $\mathcal{R}_{g_{\mathcal{H}}}(V)$ contient tous les transversaux de l'hypergraphe partiel constitué des hyperarêtes $\mathcal{E}\backslash g_{\mathcal{H}}(V)$ et leur minimaux sont les traverses minimales de ce même hypergraphe partiel. Lorsque l'hypergraphe partiel est $\mathcal{H}$ en entier alors les minimaux sont les traverses minimales de $\mathcal{H}$ i.e.ce que nous recherchons. Toujours avec le même exemple, la classe d'équivalence de v_3 est égale à :

$$\mathcal{R}_{g_{\mathcal{H}}}(v_3) = \{v_3, v_2v_3, v_2v_5, v_3v_5, v_3v_7, v_2v_3v_5, v_2v_3v_7, v_2v_5v_7, v_3v_5v_7, v_2v_3v_5v_7\}$$

Cette classe d'équivalence contient tous les ensembles de sommets dont l'extension est e_4 ou en d'autres termes tous les transversaux de l'hypergraphe partiel composé des hyperarêtes e_1, e_2, e_3, e_5 et e_6. Elle est partiellement représentée sur la figure 5.3.

Caractérisation des minimaux On définit les générateurs minimaux d'une classe d'équivalence similairement aux motifs libres (voir la définition 8 de la page 18).

Définition 26 *Un ensemble de sommets $V \subseteq \mathcal{V}$ est un générateur minimal si et seulement si V est minimal au sens de l'inclusion dans $\mathcal{R}_{g_{\mathcal{H}}}(V)$.*

Sur la figure 5.3, on voit que v_3v_4 est un générateur minimal car ni v_3 ni v_4 n'appartient à la classe d'équivalence $\mathcal{R}g_{\mathcal{H}}(v_3v_4)$. Le lemme suivant reformule la définition d'un générateur minimal.

Lemme 1 *$V \subseteq \mathcal{V}$ est un générateur minimal si et seulement si pour tout $v \in V, |g_{\mathcal{H}}(V)| < |g_{\mathcal{H}}(V\backslash\{v\})|$.*

Preuve *V est un générateur minimal si et seulement si :*

$$\begin{aligned}
\forall V' \subset V,\ V' \notin \mathcal{R}_{g_{\mathcal{H}}}(V) &\Leftrightarrow \forall V' \subset V,\ g_{\mathcal{H}}(V') \neq g_{\mathcal{H}}(V) \\
&\Leftrightarrow \forall V' \subset V,\ |g_{\mathcal{H}}(V')| > |g_{\mathcal{H}}(V)| \\
&\Leftrightarrow \forall v \in V,\ |g_{\mathcal{H}}(V\backslash\{v\})| > |g_{\mathcal{H}}(V)|
\end{aligned}$$

Nous nous servirons de cette définition dans l'algorithme donné à la section 5.3.2 car elle est plus propice à l'utilisation d'un algorithme par niveaux.

Caractérisation des transversaux Le lemme suivant caractérise les transversaux à partir de l'extension d'un ensemble de sommets.

Lemme 2 *Un ensemble de sommets V est un transversal de $\mathcal{H}$ si et seulement si $|g_{\mathcal{H}}(V)| = 0$.*

$|g_{\mathcal{H}}(V)| = 0$ implique que l'extension de V est vide et donc que V intersecte toutes les hyperarêtes de $\mathcal{H}$. Ce résultat est reformulé dans le corollaire suivant :

Corollaire 1 *Soit $\mathcal{R}^{\emptyset}_{g_\mathcal{H}} = \{V \subseteq \mathcal{V} \mid g_\mathcal{H}(V) = \emptyset\}$, la classe d'équivalence des motifs ayant une extension vide. On a l'égalité suivante : $\mathcal{R}^{\emptyset}_{g_\mathcal{H}} = Tr(\mathcal{H})$.*

Ainsi, nous avons identifié les transversaux d'un hypergraphe comme les ensembles de sommets ayant une extension vide. Remarquons que $v_1v_2 \dots v_n$ vérifie nécessairement $g_\mathcal{H}(v_1v_2 \dots v_n) = \emptyset$ d'où l'égalité entre $\mathcal{R}_{g_\mathcal{H}}(\{v_1v_2 \dots v_n\})$ et $Tr(\mathcal{H})$.

Il découle naturellement des caractérisations précédentes le théorème central qui caractérise les transversaux minimaux à partir de l'extension.

Théorème 14 *$V \subseteq \mathcal{V}$ appartient à $MinTr(\mathcal{H})$ si et seulement si V est un générateur minimal de $\mathcal{R}^{\emptyset}_{g_\mathcal{H}}$.*

Preuve *La démonstration de ce théorème est immédiate : elle provient de la combinaison du corollaire 1 et de la définition 26.*

L'extension de v_3v_4 est vide. Nous avons de plus vu que v_3v_4 est un générateur minimal donc v_3v_4 est une traverse minimale de $\mathcal{H}$.

5.3 Calcul des traverses minimales

Cette section détaille le fonctionnement de notre méthode pour le calcul des traverses minimales. Nous donnons tout d'abord les critères d'élagage qui sont utilisés puis l'algorithme MTMINER qui s'appuie largement sur ceux-ci ainsi que sur les résultats de la section précédente.

5.3.1 Stratégie d'élagage

La figure 5.3 représente un extrait du treillis généré lors du parcours par niveau des ensembles de sommets candidats à former une traverse minimale. Cet exemple illustre les différents élagages exploités par notre méthode de calcul des traverses minimales, les ensembles élagués étant barrés. Deux classes d'équivalence y apparaissent partiellement : $\mathcal{R}_{g_\mathcal{H}}(v_3)$ et $\mathcal{R}^{\emptyset}_{g_\mathcal{H}}$.

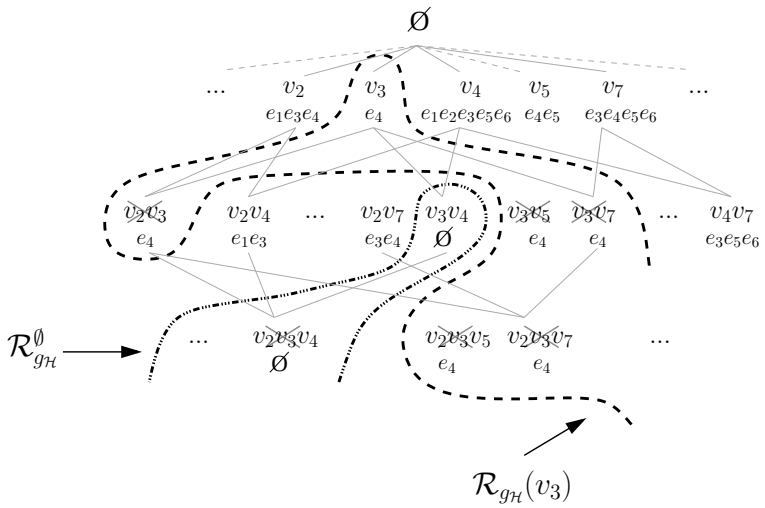

FIG. 5.3 – Les deux types d'élagages mis en œuvre dans MTMINER.

Le premier type d'élagage (voir la propriété 1 à la page 13) repose sur la propriété d'anti-monotonie de la minimalité dans les classes d'équivalence, bien connue en fouille de données

(cf. page 19) : si un ensemble de sommets n'est pas un générateur minimal alors il est possible d'élaguer l'espace de recherche à partir de celui-ci. En effet, aucun de ses sur-ensembles ne peut être un générateur minimal et le théorème 14 stipule qu'un transversal minimal est nécessairement un générateur minimal. Sur la figure 5.3, on constate que v_2v_3, v_3v_5 et v_3v_7 ne sont pas des générateurs minimaux puisque leur extension est égale à celle de v_3 (i.e. elle est égale à e_4). On élague donc ces ensembles de sommets et tous leurs sur-ensembles. Remarquons que le sommet v_3 est ici l'unique générateur minimal de la classe d'équivalence $\mathcal{R}_{g_\mathcal{H}}(v_3)$.

Par ailleurs, lorsqu'un ensemble de sommets est une traverse minimale, aucun de ses sur-ensembles ne peut, par définition, être aussi une traverse minimale. Il est donc inutile de considérer les sur-ensembles d'un transversal minimal, ce qui donne un second critère d'élagage. Puisque v_3v_4 est une traverse minimale (cette partie de $\mathcal{V}$ vérifie le théorème 14), on élague toutes ses sur-ensembles (seul $v_2v_3v_4$ est montré sur la figure 5.3 par souci de clarté).

5.3.2 L'algorithme MTMINER

MTMINER (MT pour Minimal Transversal) fonctionne suivant le principe des algorithmes par niveaux (cf. algorithme 5). Il effectue un parcours en largeur de l'espace de recherche en commençant par les singletons qui ne sont en général pas des transversaux (voir figure 5.4). Les ensembles de sommets sont générés et sont vérifiés un par un de façon similaire à l'algorithme FTMINER donné au chapitre 3. La simultanéité de la génération et de la vérification est rendue possible par l'usage de la propriété 7 (page 39) qui porte sur l'intersection de l'extension de deux ensembles. Lorsqu'un ensemble est un transversal, on est assuré qu'il est aussi minimal grâce au parcours effectué par l'algorithme.

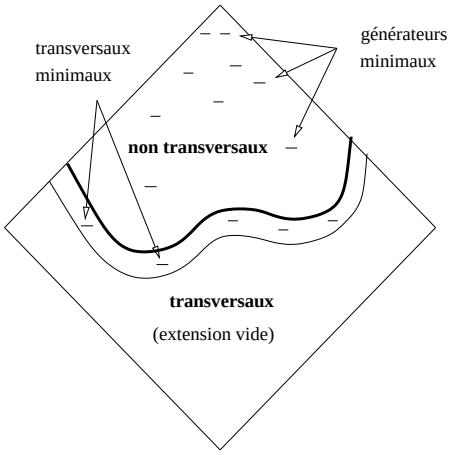

FIG. 5.4 – Espace de recherche pour MTMINER.

L'ensemble MT sert à stocker les traverses minimales découvertes et $\mathcal{G}en_k$ contient les générateurs minimaux utilisés pour générer des candidats au niveau $k+1$. La première étape consiste à initialiser MT avec les sommets d'extension vide qui sont des traverses minimales de $\mathcal{H}$ et $\mathcal{G}en_1$ avec les générateurs minimaux dont l'extension n'est pas vide. À un niveau k fixé, on génère un candidat V et on calcule son extension avec la propriété 7 à la page 7. On commence par tester si V est un générateur minimal. Puis on l'ajoute à MT si son extension vaut zéro ou on le stocke dans $\mathcal{G}en_{k+1}$ dans le cas contraire et il sert de générateur au niveau suivant. Si V n'est pas un générateur minimal, on l'élague ainsi que toutes ses spécialisations.

5.3.3 Complexité

Pour chaque traverse minimale T, MTMINER considère au plus $2^{|T|}$ ensembles de sommets. Par conséquent, l'algorithme effectue un nombre d'opérations inférieur à :

$$\sum_{T \in MinTr(\mathcal{H})} 2^{|T|}.$$

Cette borne supérieure n'est en général pas atteinte car à un niveau k donné, pour deux traverses minimales T_1 et T_2, les deux treillis ayant pour borne supérieure T_1 et T_2 ont souvent une intersection non vide. Les ensembles de sommets de cette intersection ne sont vérifiés qu'une seule fois.

Puisque $|T| \leq t(\mathcal{H})$, on a le théorème 15 :

Théorème 15 *Pour un hypergraphe $\mathcal{H}$ donné, l'algorithme* MTMINER *calcule $MinTr(\mathcal{H})$ en :*

$$\mathcal{O}(2^{t(\mathcal{H})} \times |MinTr(\mathcal{H})|).$$

Entrée : l'hypergraphe $\mathcal{H}$
Sortie : $MinTr(\mathcal{H})$, l'ensemble des traverses minimales de $\mathcal{H}$
// initialisation de MT
1 $MT := \{\{v\} \in \mathcal{V} \mid |g_{\mathcal{H}}(\{v\})| = 0\}$;
// initialisation de $\mathcal{G}en_1$
2 $\mathcal{G}en_1 := \{\{v\} \in \mathcal{V} \mid |\mathcal{E}| > |g_{\mathcal{H}}(\{v\})| > 0\}$;
3 $k := 1$;
// boucle principale
4 **tant que** $\mathcal{G}en_k \neq \emptyset$ **faire**
5 **pour tout** $(V \cup \{v_1\}, V \cup \{v_2\}) \in \mathcal{G}en_k \times \mathcal{G}en_k$ **faire**
// génération d'un candidat X de longueur $k+1$
6 $W := V \cup \{v_1\} \cup \{v_2\}$;
// calcul de l'extension
7 $g_{\mathcal{H}}(W) := g_{\mathcal{H}}(V \cup \{v_1\}) \cap g_{\mathcal{H}}(V \cup \{v_2\})$;
// vérification et élagage
8 $i := 1$;
// élagage par anti-monotonie de la minimalité
9 **tant que** $i \leq k+1$ **et** $W \backslash \{v_i\} \in \mathcal{G}en_k$ **et** $|g_{\mathcal{H}}(W)| < |g_{\mathcal{H}}(W \backslash \{v_i\})|$ **faire**
10 $i := i + 1$;
fin
11 **si** $i = k+2$ **alors**
// élagage des spécialisations d'une traverse minimale
12 **si** $|g_{\mathcal{H}}(W)| = 0$ **alors**
13 $MT = MT \cup \{W\}$;
sinon
14 $\mathcal{G}en_{k+1} := \mathcal{G}en_{k+1} \cup \{W\}$;
fin
fin
fin
15 $k := k + 1$;
fin
16 **retourner** MT ;

Algorithme 5 : MTMINER

La complexité de MTMINER dépend de $t(\mathcal{H})$ et $|MinTr(\mathcal{H})|$ qui est la taille de la sortie. Rappelons que la complexité du meilleur algorithme connu [FK96] dépend à la fois de la taille de l'hypergraphe d'entrée et de la sortie. Comme nous ne connaissons aucune relation entre $t(\mathcal{H})$ et la taille de l'entrée ou de la sortie, nous ne sommes pas en mesure de comparer ces deux méthodes d'un point de vue formel. C'est pourquoi nous effectuons une étude expérimentale à la section suivante. Nous verrons que $t(\mathcal{H})$ reste petit en pratique quand les hyperarêtes sont grandes.

5.4 Évaluation expérimentale

Nous comparons MTMINER avec deux autres prototypes dédiés au calcul des traverses minimales d'un hypergraphe donné : DUAL and THG. Rappelons qu'il existe de nombreuses méthodes de calcul des traverses minimales (cf. section 5.1.2) et notre choix s'est effectué de la manière suivante. D'une part, nous voulions comparer notre méthode à un algorithme dont la complexité est connue. DUAL est l'agorithme ayant la meilleure complexité théorique. Il est décrit dans [BEGK03] et est basé sur l'algorithme de Fredman et Khachiyan [FK96]. D'autre part, nous souhaitions également comparer MTMINER à un prototype efficace en pratique. THG est une amélioration de l'algorithme de Berge [Ber89, p. 52]. De nombreux tests dans [KS05] ont prouvé son efficacité pratique. Ces implémentations ont été téléchargées aux url `http://rutcor.rutgers.edu/~boros/IDM/DualizationCode.html` et `http://lca.ceid.upatras.gr/~estavrop/transversal/`. MTMINER est disponible à l'adresse `http://www.info.unicaen.fr/~chebert/mtminer.html`.

Ces expériences sont menées sur des hypergraphes générés aléatoirement. D'autres expériences avec une portée applicative sont présentées au chapitre 5 : la section 7.1 montre que MTMINER permet de calculer la bordure négative d'un ensemble de motifs fréquents à partir de sa bordure positive [MT97] alors que les autres prototypes échouent ; la section 7.2 donne un cas pratique d'utilisation du prototype MTMINER. Les expérimentations ont toutes été menées sur un processeur Xeon 2.20 GHz fonctionnant avec 3 Go de mémoire RAM sous Linux. Tous les temps d'exécution sont donnés en secondes.

Dans cette section, nous comparons MTMINER avec DUAL et THG sur des hypergraphes générés aléatoirement suivant le modèle d'Erdős-Rényi [ER59]. Les hypergraphes sont entre autres caractérisés par un paramètre p qui est la probabilité d'appartenance d'un sommet à une hyperarête donnée. C'est également la proportion de 1 dans la matrice d'incidence de l'hypergraphe considéré. Plus p est élevé, plus les hyperarêtes sont grandes et plus la matrice d'incidence est dense.

Comportement suivant p Notre premier objectif est d'étudier les temps d'exécution des trois prototypes en fonction de p. Nous savons que les performances de MTMINER dépendent largement de $t(\mathcal{H})$ et que celles de DUAL dépendent de $|MinTr(\mathcal{H})|$. Bien que n'en ayant pas de preuve, nous nous attendons à ce que $t(\mathcal{H})$ soit élevé lorsque la matrice de l'hypergraphe d'entrée est peu dense. Par conséquent, le cas des hypergraphes peu denses est difficile pour MTMINER. Les résultats sont consignés dans le tableau 5.3.

Excepté pour $p = 0,1$, ces expériences montrent que l'extraction des traverses minimales est très difficile lorsque les hypergraphes sont peu denses. Nous supposons que dans ce cas, il existe un grand nombre de traverses minimales et que celles-ci sont très longues. Cette double difficulté rend l'extraction impossible quel que soit le prototype utilisé. DUAL échoue dans la majorité des extractions sauf pour $p = 0,9$ et $p = 0,1$. Ce dernier résultat confirme que le paramètre

p	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1
Dual	326,70	fail	fail	fail	fail	fail	fail	fail	59,29
Thg	9,56	117,15	1 015,26	7 272,22	fail	fail	fail	fail	7 308,28
MTminer	0,25	4,14	48,72	530,02	fail	fail	fail	fail	fail
$t(\mathcal{H})$	3	5	7	8	?	?	?	?	41
$\lvert MinTr(\mathcal{H})\rvert$	26 939	339 372	2 634 205	16 237 137	?	?	?	?	4 396

Tab. 5.3 – Temps d'exécution avec $|\mathcal{V}| = 50$, $|\mathcal{E}| = 1\,000$ et p variant entre 0,9 et 0,1.

$t(\mathcal{H})$ n'intervient pas dans la complexité de cet algorithme. Par ailleurs, MTminer se montre beaucoup plus efficace que Thg.

Comportement sur des hypergraphes denses Le deuxième objectif est de mieux évaluer l'efficacité de MTminer par rapport à Dual et Thg lorsque l'hypergraphe d'entrée est dense et que le nombre d'hyperarêtes augmente. Le tableau 5.4 montre le gain de temps apporté par MTminer quand la matrice d'incidence a une densité de 0,8. Par exemple quand $|\mathcal{E}| = 20\,000$, Thg met environ 30 heures pour déterminer 7 628 650 de traverses minimales alors que MTminer n'a besoin que de 169 secondes (Dual échoue).

$\lvert\mathcal{E}\rvert$	200	400	600	800	1 000	2 000
Dual	297,80	1 042,72	1 865,88	2 681,69	4 143,26	17 854,75
Thg	4,11	15,82	40,01	67,18	120,03	672,07
MTminer	0,52	1,17	2,11	2,75	4,17	10,67
$\lvert\mathcal{E}\rvert$	3 000	5 000	7 000	10 000	20 000	
Dual	fail	fail	fail	fail	fail	
Thg	1 871,67	4 540,11	10 400,55	26 324,78	106 623,39	
MTminer	16,67	38,28	57,94	88,64	168,72	

Tab. 5.4 – Temps d'extraction avec $|\mathcal{V}| = 50$, $p = 0,8$ et $|\mathcal{E}|$ allant de 200 à 20 000.

En résumé, ces expériences montrent clairement que MTminer est plus efficace que les deux autres prototypes testés sur les hypergraphes denses.

5.5 Bilan

Dans ce chapitre, nous avons mis en évidence les liens existant entre les représentations condensées basées sur les motifs libres et les traverses minimales d'un hypergraphe. Ceux-ci ont permis de bénéficier de savoir-faire en extraction de motifs pour proposer une nouvelle méthode de calcul des traverses minimales. Malgré la difficulté algorithmique de ce problème, nous avons montré expérimentalement que notre méthode est efficace sur des hypergraphes générés aléatoirement même si, d'un point de vue théorique, nous n'avons pas réussi à établir une comparaison avec d'autres approches. Une étude de la complexité moyenne de notre algorithme permettrait peut-être de répondre à cette question et de mieux cerner son comportement. Le chapitre 7 propose des expériences complémentaires et un exemple concret d'utilisation de notre proptotype MTminer.

Notre approche repose notamment sur les concepts de connexion de Galois et d'extension. Remarquons que c'est la recherche d'une solution pour l'extraction de motifs δ-libres dans les données larges et l'utilisation de l'extension, qui ont facilité le parallèle avec les hypergraphes. Ainsi, le problème de l'extraction dans les données larges a abouti à des résultats qui dépassent largement le cadre de la fouille de données.

Conclusion

Dans cette partie, nous avons présenté nos principales contributions autour de l'extraction et des usages des motifs minimaux. Le chapitre 3 a proposé une nouvelle méthode d'extraction des motifs δ-libres dans les données comportant un grand nombre d'attributs. Celle-ci tire profit du petit nombre d'objets contenus dans les données larges et d'un critère d'élagage original qui provient de la combinaison des contraintes de fréquence et de δ-liberté. Cette approche a été étendue à la découverte de règles de caractérisation δ-fortes et nous verrons au chapitre 6 son intérêt dans des données biologiques.

Le chapitre 4 a introduit un cadre formel qui facilite l'étude du comportement des mesures d'intérêt. Ce cadre est basé sur trois propriétés qui sont finalement vérifiées par une grande partie des mesures les plus connues, appelées SBMs. Il est démontré que les SBMs sont toutes minorées simultanément par des règles nommées règles optimisées. Finalement, notre cadre souligne le fait que les SBMs sont toutes construites à partir du même « moule » et qu'elles ont des comportements similaires. Par ailleurs, nous définissons un ensemble réduit de règles (les règles optimisées informatives) qui permettent de présenter une synthèse non redondante des règles optimisées et nous donnons une méthode pour les extraire. Les règles optimisées informatives sont construites à partir de représentations condensées basées sur les motifs libres et les motifs fermés. Une simplification du cadre et l'obtention d'une couverture des règles optimisées sont possibles dans le cas particulier des règles de classification.

Au chapitre 5, nous avons présenté les liens qui unissent déjà les hypergraphes et la fouille de données. Un hypergraphe est considéré comme une base de données et ce domaine est plongé dans celui de l'extraction de motifs. La méthode développée au chapitre 3 est alors adaptée à la recherche des traverses minimales dans un hypergraphe. Les traverses minimales sont identifiées à des motifs minimaux moyennant un changement de connexion de Galois. Une approche par niveaux et l'anti-monotonie de la minimalité dans le treillis sont exploitées pour fournir une méthode efficace de calcul des traverses minimales. Les expériences montrent que notre méthode est particulièrement efficace dans les hypergraphes denses. C'est précisément sur des hypergraphes de cette nature que fonctionnent les algorithmes en profondeur de calcul de bordure. Le chapitre 7 montrera l'apport de notre prototype MTminer pour ces algorithmes ainsi que pour un travail de visualisation de données.

Troisième partie

Applications

Introduction

Cette partie montre l'apport de notre travail pour des problèmes réels. Les résultats obtenus dans la partie II sont appliqués à des contextes applicatifs variés. Le chapitre 6 présente une collaboration avec le laboratoire CGMC de Lyon. Cette collaboration porte sur l'utilisation et l'interprétation de règles de caractérisation δ-fortes pour la découverte de gènes corégulés impliqués dans le développement du cancer. Pour obtenir ces règles, nous avons utilisé le prototype FTCminer dans les données SAGE. Le chapitre 7 décrit deux contextes où notre approche pour calculer les traverses minimales d'un hypergraphe est utilisée avec succès : d'une part, pour la recherche de bordures de motifs fréquents et d'autre part, pour la mise en évidence de clusters dans le cadre d'un processus de visualisation de données.

Chapitre 6

Usage des motifs δ-libres : découverte de gènes corégulés dans les données larges

Sommaire

Dans ce chapitre, nous montrons comment notre méthode de découverte de motifs δ-libres (cf. chapitre 3) contribue à l'analyse du transcriptome en permettant la recherche d'information dans les données d'expression de gènes. Rappelons que les données SAGE, évoquées au chapitre d'introduction, décrivent l'expression d'un grand nombre de gènes et que leurs dimensions rendent les algorithmes usuels d'extraction de motifs inefficaces. C'est pourquoi nous utilisons l'approche décrite au chapitre 3 pour faire face aux difficultés algorithmiques induites par un grand nombre d'attributs.

Ce travail est le fruit d'une collaboration avec Sylvain Blachon et Olivier Gandrillon, biologistes au Centre de Génétique Moléculaire et Cellulaire (CGMC) CNRS UMR 5534 dans le cadre du projet Bingo relevant de l'ACI Masse de Données.

6.1 Présentation des données

6.1.1 Objectifs

L'auto-renouvellement des cellules est un phénomène biologique capital qui se situe notamment au cœur de processus cancéreux [RMCW01]. La compréhension de ce phénomène s'appuie sur l'étude du niveau d'expression des gènes de ces cellules. Serial Analysis of Gene Expression (SAGE) est une technique expérimentale [VZVK95] permettant de récolter les expressions de nombreux gènes dans des situations biologiques diverses. Parmi les attentes des biologistes,

on peut citer la détermination de gènes corégulés c'est-à-dire qui sont simultanément surexprimés [BBJ+02] ou sousexprimés. Ceux-ci permettent de déterminer des groupes de synexpression ou d'obtenir des règles permettant d'aboutir à une classification fine des différents types de cancer. Le tableau 1 contient l'expression de 8 gènes chez 6 humains. Un 1 dans la case $[i, j]$ de ce tableau signifie que le gène j est surexprimé (i.e. que son expression dépasse les valeurs « habituelles ») chez le patient i. Dans le même exemple, les gènes a_3, a_5 et a_7 sont surexprimés dans les situations o_1 et o_2.

6.1.2 Les données SAGE

Nous travaillons sur des données SAGE disponibles à l'url `http://cgap.nci.nih.gov/SAGE` et utilisées pour les discovery challenges 2004 et 2005 [20] associés aux conférences ECML/PKDD. Ces données décrivent le niveau d'expression de 27 679 gènes chez 90 patients humains. En fonctionnant, un gène produit des séquences ADN appelées ARN messager. En comptant le nombre de séquences produites par un gène, on peut estimer son niveau d'expression. Chaque ARN est constitué de milliers de nucléotides (composés d'une des quatre bases azotées : C pour Cytosine, T pour Thymine, A pour Adénine, G pour Guanine) et une séquence de 10 ou 14 nucléotides - appelée *tag* - est suffisante pour l'identifier. Les données SAGE indiquent en réalité le nombre d'occurrences d'un tag qu'il faut ensuite associer au gène dont l'ARN est issu. Le tableau 6.1 donne pour quelques-uns des tags présents dans les données SAGE leur identifiant, la séquence de nucléotides correspondante et la description du gène dont ils sont issus. Certains tags ne sont pas encore identifiés de manière précise et c'est pourquoi on leur associe plusieurs descriptions séparées par un point-virgule dans le tableau 6.1.

Identifiant	Séquence	Description
4287	AGCTCTCCCT	RPL17 CDNA sequence BC022357 ; PIGK Phosphatidylinositol glycan, class K
4602	AGGCTACGGA	Similar to ribosomal protein L13a, 60S ribosomal protein L13a, 23 kD highly basic protein
8255	CATCCAAAAC	HNRPH1 Heterogeneous nuclear ribonucleoprotein H1 (H)
11115	CTCTTCGAGA	GPX1 Glutathione peroxidase 1
19811	GTTGCTGCCC	NIFIE14 Seven transmembrane domain protein
22129	TCAGAGAATA	SLC25A22 Solute carrier family 25 (mitochondrial carrier : glutamate), member 22 ; IRS2 Insulin receptor substrate 2
25202	TGTGCTAAAT	RPL34 Ribosomal protein L34 ; USP36 Ubiquitin specific protease 36

TAB. 6.1 – Identification de tags.

Les situations biologiques se répartissent en 59 situations cancéreuses avec différents cancers et 31 non cancéreuses. La technique SAGE récolte l'expression des gènes sous forme de valeurs entières. Une étape de discrétisation est donc nécessaire pour se ramener aux contextes classiques de fouille de données dans lesquels seules des valeurs boléennes sont admises. Ce travail de préparation des données qui consiste à coder la sur-expression des gènes est détaillé dans [BBJ+02].

[20] Pour plus de renseignements, on se référera aux sites `http://lisp.vse.cz/challenge/ecmlpkdd2004/` et `http://lisp.vse.cz/challenge/ecmlpkdd2005/`.

Nous en indiquons uniquement les grandes lignes. Trois méthodes de discrétisation ont été appliquées aux données sous leur forme numérique :

1. la méthode `Xmax` : un tag est considéré comme sur-exprimé dans les $X\%$ de situations dans lesquelles il a les valeurs les plus élevées (ici, X est fixé à 5) ;
2. la méthode `max` $-$ `Xmax` : soit `max` la valeur maximale observée d'un tag. Ce tag est sur-exprimé lorsque sa valeur dépasse $100 - X\%$ de `max` (dans cette expérience, $X = 25$) ;
3. la méthode `median` : un tag est sur-exprimé si sa valeur dépasse la moyenne de ses valeurs extrêmes dans les données (minimum et maximum).

Le tableau 6.2 donne le pourcentage de gènes sur-exprimés pour chaque discrétisation ainsi que le nombre moyen de gènes sur-exprimés par ligne i.e. la taille moyenne d'un objet.

base	`Xmax`	`max` $-$ `Xmax`	`median`
gènes sur-exprimés (%)	4,49	2,01	3,64
nombre de gènes sur-exprimés par ligne	1 242,03	554,98	1 008,12

TAB. 6.2 – Caractéristiques des trois bases de données `Xmax`, `max` $-$ `Xmax` et `median`.

6.2 Extraction de règles de caractérisation δ-fortes

Dans cette section, nous commençons par donner une vue d'ensemble des règles de caractérisation δ-fortes extraites à partir des trois bases `Xmax`, `max` $-$ `Xmax` et `median` pour différents seuils de fréquence et en tolérant différents nombres d'exceptions. Ensuite, nous examinons plus en détail des règles et des tags qui nous ont semblé particulièrement pertinents pour caractériser les situations biologiques en fonction de leur classe (`cancer` et `normal`). Nous insistons sur le fait que cette sélection de règles et de tags n'a été possible que grâce à une concertation avec les experts du CGMC. Les extractions ont bien sûr été réalisées avec le prototype FTCMINER présenté au chapitre 3 (page 48). Nous n'énumérons pas les temps d'exécution de FTCMINER pour chaque extraction puisque la plus longue dure une vingtaine de secondes environ sur un processeur Xeon à 2.20 Ghz avec 3 Go de RAM sous le système d'exploitation Linux.

6.2.1 Vue d'ensemble des règles obtenues

Nombre de règles pour différents seuils γ et δ Le tableau 6.3 donne le nombre de règles de caractérisation δ-fortes extraites à partir des trois bases discrétisées lorsque les paramètres γ et δ prennent différentes valeurs ($\gamma = 5, 7, 9, 10, 15$ et $\delta = 1, 2, 3$). Le nombre de règles augmente avec δ et diminue avec γ. Pour $\gamma = 15$ et $\delta = 3$, il n'y a pas de règles dans `Xmax` et `max` $-$ `Xmax`. Les règles extraites à partir de `median` concluent toutes sur `cancer`. Cela est peut-être dû au déséquilibre entre les deux classes : puisqu'il n'y a que 31 situations « normales », fixer $\gamma = 15$ et $\delta = 3$ revient à rechercher des règles apparaissant dans presque 40% des objets portant l'étiquette `normal`. C'est à partir de la base `median` que l'on obtient le plus grand nombre de règles. Pour $\gamma = 7$ et $\delta = 2$, il y en a 13 (resp. 40) fois plus que dans `Xmax` (resp. `max` $-$ `Xmax`). C'est surprenant puisque `median` n'a pas la densité la plus élevée (cf. tableau 6.2), ce qui aurait pu être une explication. Remarquons qu'un phénomène similaire est rapporté dans [BBJ+02].

Intersection des règles issues des trois discrétisations Une première hypothèse a été qu'une règle qui apparaît dans plusieurs bases est relativement indépendante de la méthode

γ	15		10		10		9	
δ	3		3		2		2	
conclusion	`cancer`	`normal`	`cancer`	`normal`	`cancer`	`normal`	`cancer`	`normal`
`Xmax`	0	0	8	0	2	0	12	0
`max` − `Xmax`	0	0	10	1	1	0	9	0
`median`	45	0	638	8	369	1	777	3
γ	7		5		5		4	
δ	2		2		1		1	
conclusion	`cancer`	`normal`	`cancer`	`normal`	`cancer`	`normal`	`cancer`	`normal`
`Xmax`	278	29	4 837	1 322	2 838	341	12 602	2 952
`max` − `Xmax`	89	4	761	135	489	31	1 367	186
`median`	3 543	104	23 872	4 548	20 622	996	80 965	11 676

TAB. 6.3 – Nombre de règles de caractérisation δ-fortes issues des données SAGE.

de discrétisation i.e. qu'elle présente une robustesse qui la rend plus fiable. Il n'existe de telles règles que lorsque γ est égal à 4 et δ vaut 1, c'est pourquoi nous nous focalisons sur le résultat de l'extraction réalisée pour ces valeurs. Le tableau 6.4 donne les règles communes à plusieurs bases dans cette expérience. La fréquence de la prémisse et la confiance d'une règle dépendent de la base dont elle est issue et il arrive donc qu'une même règle admette des fréquences et des confiances différentes en fonction de sa base d'origine (une même règle peut donc être présentée sur plusieurs lignes, suivant la base). Les bases `Xmax` et `max` − `Xmax` partagent 1 496 règles, `max` − `Xmax` et `median` en ont 768 en commun et `Xmax` et `median` en ont 4 398 en commun. Remarquons qu'une seule et unique règle apparaît dans les trois bases : 8091 19351 → `normal`. Elle apparaît dans les situations numérotées 12, 38 et 84. Malheureusement, l'association des tags 8091 et 19351 a été jugée peu intéressante par les experts. `median` étant la base qui fournit le plus grand nombre de règles, nous concentrons nos efforts sur celle-ci par la suite.

6.2.2 Interprétation des résultats

Une interprétation plus poussée au point de vue biologique est menée dans la thèse de Sylvain Blachon [Bla07].

Règles pertinentes Les biologistes s'intéressent aux ensembles de gènes ayant un profil similaire et non pas au comportement d'un seul gène car il semble naïf de considérer qu'un seul gène est impliqué dans le développement d'une maladie aussi complexe que le cancer. Aussi dans un premier temps, nous éliminons les règles dont la prémisse est constituée d'un seul tag soit 4 368 règles et nous présentons au tableau 6.5 les règles ayant au moins deux tags en prémisse et dont les valeurs de support et de confiance sont les plus élevées.

La règle 4287 4602 19811 → `cancer` (en gras dans le tableau 6.5) semble particulièrement intéressante d'après les biologistes. Dans sept situations (20, 37, 45, 46, 48, 51, 76), l'association des trois tags 4287, 4602 et 19811 conclut sur la classe `cancer`. La confiance de cette règle est égale à $0,875$. Les trois tags impliqués sont décrits dans le tableau 6.1. Deux d'entre eux (les tags 4287 et 4602) sont identifiés comme des protéines ribosomales et le troisième correspond à une protéine transmembranaire. L'intérêt de cette règle réside dans le fait que NIFIE14 (tag 19811) est une protéine découverte récemment et que le rôle des protéines transmembranaires dans le développement du cancer est très étudié. Une perturbation dans le processus de communication cellulaire est souvent invoquée comme une des principales causes de cancer [HF02]. De plus, les protéines transmembranaires sont de plus en plus soupçonnées d'être liées aux tumeurs. Par

communes à `max-Xmax` et `median`					
Prémisse	Conclusion	Exceptions	Fréquence	Confiance	Base
7259 14143	`cancer`	0	4	1	`max − Xmax`
		1	6	0,833	`median`
11695 17436	`cancer`	1	5	0,800	les deux
12719 19258	`cancer`	1	4	0,750	`max − Xmax`
		1	7	0,857	`median`
22218 26894	`cancer`	0	4	1	les deux
6756 26019	`normal`	1	4	0,750	`max − Xmax`
		1	6	0,833	`median`
13954 27489	`normal`	0	4	1	les deux
communes à `Xmax` et `median`					
Prémisse	Conclusion	Exceptions	Fréquence	Confiance	Base
566 11119	`cancer`	1	4	0,750	`Xmax`
		1	5	0,800	`median`
1525 9002	`cancer`	0	4	1	les deux
9739 27441	`cancer`	1	4	0,750	`Xmax`
		1	5	0,800	`median`
11119 21930	`cancer`	0	4	1	les deux
2467 20091	`normal`	1	4	0,750	les deux
20091 27139	`normal`	1	4	0,750	les deux
commune à `Xmax`, `max − Xmax` et `median`					
Prémisse	Conclusion	Exceptions	Fréquence	Confiance	Base
8091 19351	`normal`	1	4	0,750	toutes

TAB. 6.4 – Caractéristiques des règles communes à plusieurs bases pour $\gamma = 4$ et $\delta = 1$.

exemple, une récente étude a montré que l'inhibition de certaines fonctions de ces protéines pourraient constituer un traitement efficace des cellules cancéreuses [GMH+05]. Remarquons également que les sept situations supportant la règle sont relativement homogènes puisque cinq d'entre elles concernent les cellules issues de la vessie et les deux autres viennent respectivement du pancréas et du cerveau. Or, il a déjà été montré que le gène identifiant le tag 4287 est impliqué dans les cancers de la vessie [GLW+04] ce qui tend à valider l'intérêt biologique de cette règle.

Le tableau 6.6 montre également quelques règles extraites à fort support et confiance dans les trois bases pour $\gamma = 5$ et $\delta = 2$.

Tags apparaissant dans de nombreuses règles Quelques tags comme 4602, 8255, 11115 ou 22129 apparaissent clairement dans de nombreuses règles qui concluent sur la classe `cancer`. Nous pensons qu'ils pourraient influer fortement sur le développement de la maladie et les biologistes se sont donc particulièrement intéressés à leur signification. Par exemple, le tag 11115 apparaît dans 28,7 fois plus de règles caractérisant les situations étiquetées cancer que les situations normales lorsque $\gamma = 4$ et $\delta = 1$ (voir le tableau 6.5). Or, ce tag a été identifié comme provenant du gène GPX1 (cf. tableau 6.1) et le niveau d'expression de GPX1 a déjà été supposé corrélé avec les situations cancéreuses [KMD+02, NFED04]. Au contraire, le tag 22129 est souvent présent dans des situations normales (il apparaît dans 22 fois plus de règles concluant sur la classe `normal` que sur `cancer`). De manière générale, nous pensons qu'une étude approfondie des tags essentiellement présents dans la prémisse de règles qui ont la même conclusion pourrait révéler des gènes jouant un rôle particulier dans le développement du cancer.

median				
Prémisse	Conclusion	Exceptions	Fréquence	Confiance
11115 19811	cancer	1	13	0,923
5961 11115	cancer	0	12	1
8279 23600	cancer	1	12	0,917
10960 11115	cancer	1	12	0,917
11115 20766	cancer	1	12	0,917
4602 7259 18882	cancer	1	10	0,900
4602 7259 24686	cancer	1	10	0,900
8255 11115 19811	cancer	1	10	0,900
4602 7259 20461	cancer	1	9	0,889
4602 7259 25202	cancer	1	9	0,889
4602 18882 24686	cancer	1	9	0,889
4287 4602 7818	cancer	1	8	0,875
4287 4602 19811	cancer	**1**	**8**	**0,875**
4602 7259 19734	cancer	1	8	0,875
4602 24686 25202	cancer	1	8	0,875
4602 25128 25202	cancer	1	8	0,875
7259 12667 16807	cancer	1	8	0,875
8255 11115 13642	cancer	0	8	1
8255 11115 26846	cancer	1	8	0,875
8255 19811 26846	cancer	1	8	0,875
22619 25202 26846 27358	cancer	1	5	0,800
16786 26715	normal	1	7	0,857
22129 25356	normal	1	7	0,857
22129 27414	normal	1	7	0,857
22647 25356	normal	1	7	0,857
1722 25202 26715	normal	1	6	0,833

TAB. 6.5 – Exemples de règles extraites de **median** avec $\gamma = 4$ et $\delta = 1$.

Xmax				
Prémisse	Conclusion	Exceptions	Fréquence	Confiance
431 9002	cancer	1	5	0,800
6497 6544	cancer	1	5	0,800
18271 21701	normal	1	5	0,800
max − Xmax				
Prémisse	Conclusion	Exceptions	Fréquence	Confiance
3401 27230	cancer	2	5	0,600
5371 19950	cancer	2	5	0,600
median				
Prémisse	Conclusion	Exceptions	Fréquence	Confiance
4602 24686	cancer	2	17	0,882
8255 11115	cancer	2	15	0,867
4602 7259	cancer	2	14	0,857
8255 19811	cancer	2	14	0,857
16306 16690 24686	cancer	2	9	0,778
7259 24686 25202	cancer	2	8	0,750
8083 8925 19811	normal	2	6	0,667

TAB. 6.6 – Exemples de règles avec $\gamma = 5$ et $\delta = 2$.

6.2.3 Conclusion

Nous avons montré que notre méthode d'extraction des règles de caractérisation δ-fortes est opérante dans les données très larges. Rappelons que celle-ci permet de repousser les limites de la faisabilité des extractions car les algorithmes usuels sont inexploitables dans des données de telles dimensions. L'examen (par les biologistes) des règles obtenues a permis d'isoler une règle qui semble prometteuse. Le fait que cinq situations sur les sept qui supportent cette règle concernent des cellules provenant de prostates nous incite à poursuivre nos investigations de manière plus fine. Une prolongation de ce travail serait de procéder au calcul des règles de caractérisation δ-fortes lorsque les valeurs de classes représentent le type de cancer (ovaires, prostate, cerveau, etc.) pour saisir les caractéristiques propres à chacune des variantes de cette maladie.

Chapitre 7

Apports de MTminer : calcul de bordures et visualisation de clusters

Sommaire

Ce chapitre souligne l'intérêt pratique de notre méthode de calcul des traverses minimales d'un hypergraphe présentée au chapitre 5. Dans la section 7.1, nous montrons l'apport de MTminer pour les algorithmes de calcul de bordure. La section 7.2 décrit l'utilisation de notre méthode lors d'un processus de visualisation de données à partir de clusterings.

7.1 Calcul de bordures lors de la recherche de motifs fréquents

7.1.1 Protocole expérimental

Dans l'état de l'art du chapitre 5, nous avons brièvement évoqué les bordures d'ensembles de motifs. La section 5.1.2 mentionne des méthodes qui calculent directement la bordure positive d'un ensemble de motifs fréquents en utilisant sa bordure négative [GKM+03, SU03]. Par exemple, la méthode [GKM+03] proposée par Gunopulos *et al.* consiste en un parcours en profondeur de l'espace de recherche pour déterminer des motifs fermés maximaux. On teste si la totalité des motifs de la bordure positive a été déterminée en calculant les traverses minimales de l'hypergraphe formé par les complémentaires des motifs contenus dans le bordure positive. Cette opération constitue une étape cruciale de cet algorithme. Dans ce chapitre, l'objectif est de comparer de façon pratique MTminer, Dual [BEGK03] et Thg [KS05] sur ce type de problème. Là encore, les tests ont été effectués avec un processeur Xeon 2.20 GHz avec 3 Go de RAM sous Linux.

Nous effectuons ces expériences sur des benchmarks de l'UCI disponibles à l'url `http://www.ics.uci.edu/~mlearn/MLSummary.html`. Nous utilisons les trois benchmarks suivants : `MUSHROOM` de dimension $8\,124 \times 120$, `LETTER-RECOGNITION` de dimension $20\,000 \times 74$ et `PUMSB` de dimension $49\,046 \times 7\,118$. Tout d'abord, nous commençons par calculer la bordure positive des motifs fréquents. Ensuite nous déterminons les complémentaires des motifs qu'elle contient.

La dernière étape, qui fait l'objet de cette expérimentation, consiste à appliquer les trois prototypes sur ces motifs complémentaires pour obtenir la bordure négative de l'ensemble des motifs fréquents. Les tableaux 7.1, 7.2 et 7.3 donnent leurs performances en secondes sur les trois benchmarks présentés plus haut lorsque le seuil de fréquence γ varie. Ils contiennent également les valeurs des paramètres qui sont impliqués dans la complexité des algorithmes utilisés (cf. section 5.3.3) : le nombre d'hyperarêtes $|\mathcal{E}|$ et la densité de l'hypergraphe $\mathcal{H}$ donné en entrée, la taille du plus grand transversal minimal $t(\mathcal{H})$ et le nombre de traverses minimales $|MinTr(\mathcal{H})|$. Le nombre de sommets de $\mathcal{H}$ est ici fixé : il est égal au nombre d'attributs du benchmark et n'est pas rappelé dans les tableaux.

7.1.2 Résultats

Premier benchmark : `MUSHROOM` Sur ce benchmark (voir le tableau 7.1), MTMINER surpasse DUAL et THG. En effet, MTMINER est en moyenne 191 fois plus rapide que DUAL et 10 fois plus rapide que THG. Globalement, le temps d'exécution de MTMINER augmente avec la taille du plus grand transversal et la taille de la sortie, ce qui confirme la complexité donnée dans l'étude théorique de la section 5.3.3. Il était attendu que la rapidité d'exécution de DUAL dépende essentiellement de la taille de l'entrée ($|\mathcal{E}|$ puisque $|\mathcal{V}|$ est fixé) et de celle de la sortie d'après la complexité donnée à la page 73. Remarquons que les performances les moins bonnes pour les trois méthodes sont réalisées quand γ est égal à 10. Cela correspond à la situation la plus difficile puisque le nombre de transversaux minimaux est très élevé (118 234), certains sont de grande taille (10), le nombre d'hyperarêtes est important et la densité est de 0,782. Quand $\gamma = 1$, la situation est plus favorable puisque DUAL et THG divisent respectivement leur temps d'exécution par 10 et 30. Au contraire, le temps d'exécution de MTMINER diminue à peine puisqu'il passe de 94,98 à 85,89 secondes. En comparaison, pour $\gamma = 50$, le calcul des traverses minimales avec MTMINER ne nécessite qu'une trentaine de secondes. Pourtant, les différents paramètres présentent à peu près les mêmes caractéristiques que dans la situation précédente : la densité et le nombre d'hyperarêtes sont sensiblement les mêmes. Le nombre de traverses minimales et la longueur du plus long transversal sont plus élevés, ce qui devrait ajouter de la difficulté à l'extraction (rappelons que la complexité de notre algorithme dépend de ces deux paramètres). Nous ne sommes pas en mesure d'expliquer ce phénomène à l'heure actuelle.

γ	800	600	400	200	100	50	30	10	1
DUAL	53,52	82,09	278,17	840,89	2 248,50	5 647,58	12 059,95	35 612,74	3 477,25
THG	0,60	1,52	5,10	29,90	117,87	404,11	1 128,39	3 161,14	103,30
MTMINER	0,27	0,58	1,55	4,48	13,48	30,49	48,21	94,98	85,89
densité	0,731	0,736	0,741	0,753	0,763	0,771	0,778	0,782	0,773
$\lvert\mathcal{E}\rvert$	573	918	1 477	3 111	5 776	9 857	15 232	30 809	8 124
$t(\mathcal{H})$	6	6	7	7	8	9	9	10	7
$\lvert MinTr(\mathcal{H})\rvert$	6 244	8 235	16 375	31 331	51 678	77 990	100 573	118 234	22 294

TAB. 7.1 – Performances (temps en secondes) sur le benchmark `MUSHROOM`.

Deuxième benchmark : `LETTER-RECOGNITION` Pour ce benchmark, DUAL est incapable de déterminer les traverses minimales (cf. tableau 7.2), quelle que soit la valeur du seuil de fréquence. La différence entre MTMINER et THG est encore plus saisissante que sur `MUSHROOM` : MTMINER est en moyenne 2 000 fois plus rapide que THG. Par exemple, MTMINER met une seconde à extraire presque 80 000 traverses minimales alors que la même extraction nécessite environ une heure

et demie avec THG. Contrairement à `MUSHROOM`, on ne constate pas de pic au niveau des temps d'exécution suivi d'une diminution lorsque le seuil de fréquence diminue. Cela s'explique par le fait que tous les paramètres sont monotones en fonction du seuil de fréquence : ils décroissent tous strictement à l'exception de la densité. Cette dernière croît alors que pour `MUSHROOM`, ce n'était pas le cas.

γ	3 000	1 000	800	600	400	300	200	100
DUAL	échec	échec	échec	échec	échec	échec	échec	échec
THG	0,21	39,54	106,96	371,01	1 750,11	5 096,74	13 891,36	77 468,80
MTMINER	0	0,13	0,21	0,42	1,09	1,89	4,48	15,36
densité	0,962	0,94	0,937	0,932	0,925	0,921	0,9147	0,905
$\lvert\mathcal{E}\rvert$	347	5 579	8 979	15 779	33 015	52 554	96 355	228 278
$t(\mathcal{H})$	4	7	7	8	9	9	11	11
$\lvert MinTr(\mathcal{H})\rvert$	1 851	16 961	25 298	43 302	79 479	121 307	207 246	453 280

TAB. 7.2 – Performances (temps en secondes) sur le benchmark `LETTER-RECOGNITION`.

Troisième benchmark : `PUMSB` Dans le tableau 7.3, nous constatons que seul MTMINER est en mesure de calculer la bordure négative pour le benchmark `PUMSB`, quelque soit le seuil de fréquence. Pourtant, les valeurs des paramètres $t(\mathcal{H})$, $|\mathcal{E}|$ et $MinTr(\mathcal{H})$ ne sont pas plus élevées que pour les deux benchmarks précédents. Seule la densité est très forte mais nous pensons plutôt que la difficulté de cette tâche, pour DUAL et THG, s'explique par le nombre de sommets de $\mathcal{H}$ qui vaut 7 118 contre 120 pour `MUSHROOM` et 74 pour `LETTER-RECOGNITION`.

γ	48 000	45 000	40 000	35 000
DUAL	échec	échec	échec	échec
THG	échec	échec	échec	échec
MTMINER	0,01	0,28	4,34	24,78
densité	0,9996	0,9993	0,9989	0,9985
$\lvert\mathcal{E}\rvert$	3	144	2 341	10 417
$t(\mathcal{H})$	1	5	9	13
$\lvert MinTr(\mathcal{H})\rvert$	7 120	7 483	14 085	41 020

TAB. 7.3 – Performances (temps en secondes) sur le benchmark `PUMSB`.

7.1.3 Bilan

En résumé, ces expériences montrent que MTMINER est beaucoup plus efficace que les deux autres prototypes testés lors du calcul de bordures négatives d'ensembles de motifs fréquents. Comme les motifs d'une bordure positive contiennent assez peu d'attributs par rapport au nombre total d'attributs de la base de données, leurs complémentaires sont de très longs motifs et les hypergraphes donnés en entrée sont très denses. Nous pensons que cette caractéristique explique l'efficacité de MTMINER. Cette intuition ressortait déjà des expérimentations menées au chapitre 5. De façon plus inattendue, nous avons découvert que MTMINER peut aussi se montrer particulièrement efficace sur des hypergraphes comportant un grand nombre de sommets. Un autre point important réside dans le fait que MTMINER permet de calculer la bordure négative des motifs fréquents extraits de `PUMSB` alors que cela serait impossible autrement.

Par ailleurs, ces expériences montrent que les paramètres impliqués dans la complexité des algorithmes testés ne sont pas indépendants les uns des autres mais que la nature exacte des liens qui les unissent n'est pas triviale. Elles ne permettent pas d'énoncer des généralités sur le comportement de ces paramètres ni d'expliquer les liens subtils qui existent entre eux. Nous pensons qu'une étude approfondie de la complexité de ces algorithmes, non seulement dans le pire cas mais aussi en moyenne, pourrait apporter des éclaircissement sur ces relations. Nous revenons sur ce point dans les perspectives de ce mémoire.

7.2 Visualisation de données catégorisées

MTminer a aussi été fructueusement utilisé dans une méthode de visualisation de données réalisé par Durand *et al.* Pour plus de détails, nous renvoyons le lecteur à l'article [DCS06]. L'enjeu de ce travail est de présenter une visualisation appropriée des données afin d'aider l'utilisateur à choisir et interpréter les résultats de plusieurs clusterings dont les clusters peuvent se recouvrir. L'idée principale est d'attribuer à deux clusters issus de différents clusterings une même couleur si ils sont proches afin de faire ressortir les différences et similarités entre ces clusterings. Dans chaque clustering, chaque objet est représentée par un bâtonnet coloré. La ou les couleur(s) d'un bâtonnet indique(nt) à quel(s) cluster(s) l'objet associé appartient. Par exemple, la figure 7.1 donne la visualisation obtenue sur la base de données géographiques de la zone « Transmanche » (voir l'url `http://atlas-transmanche.certic.unicaen.fr/` pour plus de détails sur le projet Atlas Transmanche). CC1, CC2, CC3 et CC4 représentent les quatre clusterings considérés. On peut voir que les couleurs bleu, vert, jaune, orange et rouge ont été attribuées à 5 regroupements de clusters et qu'un objet peut appartenir à plusieurs clusters simultanément (le bâtonnet qui le représente a alors plusieurs couleurs).

Du point de vue algorithmique, l'étape cruciale est l'attribution de couleurs aux clusters. Cette étape nécessite de déterminer les clusters présentant des similarités pour les regrouper. Or, le nombre de regroupements possibles pour k clusterings contenant chaun n clusters est égal à n^k. Il n'est donc pas possible de les énumérer pour choisir le meilleur regroupement suivant une mesure de similarité. Il est montré dans [DCS06] que l'ensemble des clusterings peut être modélisé par un hypergraphe et que déterminer les traverses minimales de cet hypergraphe conduit à trouver de bons regroupements. La méthode proposée est itérative : après sélection d'une traverse minimale, le processus est répété en ôtant de l'hypergraphe les clusters du regroupement effectué. Cela signifie qu'il est nécessaire de déterminer les traverses minimales à plusieurs reprises ; il faut donc disposer d'une méthode efficace pour cela. L'efficacité de MTminer a été précieuse car elle a permis de ne pas allonger le temps nécessaire à l'affectation des couleurs par rapport à une méthode naïve où pour le regroupement des clusters repose sur un algorithme glouton.

Revenons à la figure 7.1. Les données étudiées contiennent des indicateurs démographiques et économiques variés pour différentes régions françaises et anglaises. La méthode proposée révèle des informations qui n'étaient pas visibles avec la méthode naïve. Par exemple, elle fait émerger le fait que les départements français et anglais éloignés de Paris et de Londres (respectivement coloriés en vert et en bleu) ont des caractéristiques communes telles que le vieillissement de la population. Un autre résultat concerne les départements proches de Paris ou de Londres : ils sont automatiquement regroupés et coloriés en jaune sur la figure. On s'aperçoit notamment qu'ils partagent le fait d'avoir une population jeune.

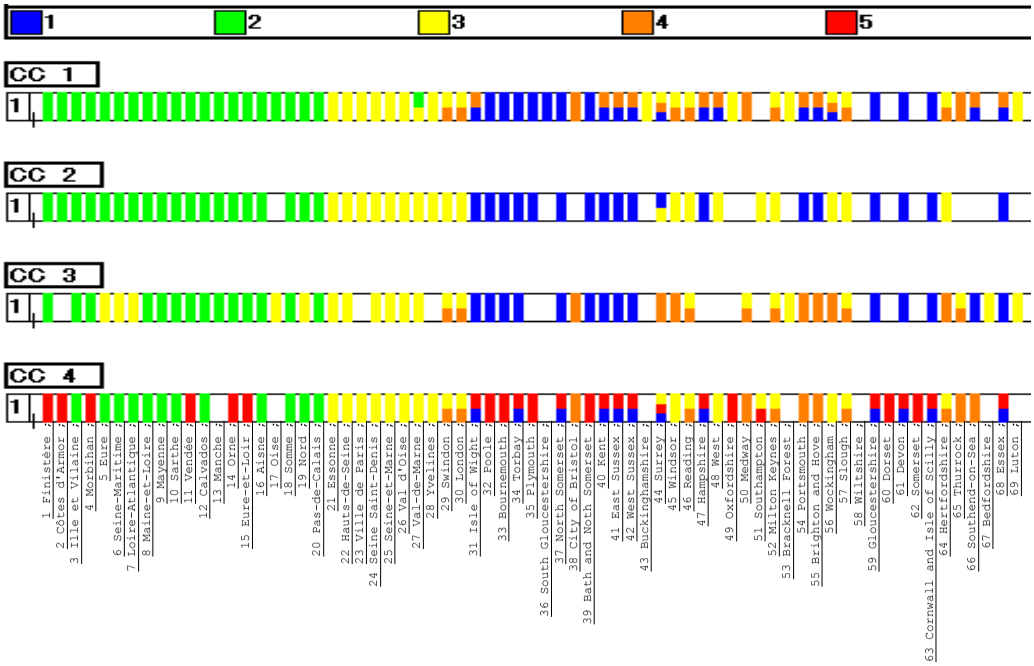

FIG. 7.1 – Visualisation de données géographiques (zone « Transmanche »).

Conclusion

La confrontation de nos méthodes avec des situations concrètes a permis de les éprouver. Au-delà de la faisabilité ou de la rapidité des extractions, ces applications ont validé l'intérêt des motifs minimaux pour des objectifs divers.

Les résultats obtenus dans le chapitre 6 sont cohérents avec la littérature alors qu'ils ont été obtenus de manière différente. Ils ouvrent des perspectives intéressantes pour l'étude de l'influence de gènes dans le développement du cancer et la caractérisation de cancers suivant l'organe affecté. Par ailleurs, alors que la complexité algorithmique du problème des traverses minimales limite fortement leur utilisation en pratique, nous avons montré l'efficacité de notre approche par rapport à deux algorithmes classiques et sa mise en œuvre dans deux applications.

Conclusion et perspectives

Conclusion et perspectives

Bilan

Notre thématique de recherche est centrée sur l'extraction et les usages des motifs minimaux en fouille de données ainsi que dans les hypergraphes. Dans ce travail, nous nous sommes attachés à couvrir un large éventail d'utilisations des motifs minimaux, de la découverte de connaissances dans les données larges aux hypergraphes en passant par la qualité des connaissances extraites lors d'un processus de fouille de données.

Découverte de motifs dans les larges jeux de données

Notre première contribution est relative aux données comportant un grand nombre d'attributs. Nous avons proposé une méthode efficace pour l'extraction des motifs libres et δ-libres dans ces contextes difficiles. L'idée clé est d'exploiter le déséquilibre des dimensions de ce type de données : les objets y sont peu nombreux et il est préférable de travailler à partir de motifs composés d'objets plutôt que d'attributs. L'approche proposée repose sur l'utilisation de l'extension des motifs dont l'intérêt dans le contexte des données larges n'avait, à notre connaissance, jamais été souligné auparavant.

Nous avons également présenté un nouveau critère d'élagage qui résulte de la combinaison des contraintes de fréquence et de δ-liberté. Nous avons identifié les cas où ces deux contraintes ne peuvent être vérifiées simultanément permettant un élagage important, comme les expériences menées le montrent. Ce critère est compatible avec l'utilisation de l'extension et la combinaison de ces deux éléments dans un algorithme par niveaux a rendu possible l'extraction de règles de caractérisation δ-fortes dans les données SAGE.

Un cadre générique pour les mesures d'intérêt

Notre seconde contribution concerne l'utilisation de mesures d'intérêt lors d'un processus de découverte de connaissances. Nous avons présenté un cadre générique en pointant les caractéristiques communes à un grand nombre de mesures usuelles, appelées SBMs. Ce point de vue unifié permet de mettre en évidence de fortes similitudes dans le comportement d'un grand nombre de mesures d'intérêt. Nous en concluons que le choix de l'une ou l'autre des mesures de notre cadre, n'est pas une étape déterminante dans un processus de fouille de données. Un avantage de notre cadre réside dans l'obtention de minorants pour toutes les SBMs.

L'identification de l'ensemble des règles d'association qui optimisent les SBMs fournit un réservoir de règles de bonne qualité relativement à ces mesures d'intérêt. Nous proposons de restreindre l'extraction à des règles à prémisse minimale et à conclusion maximale, appelées

règles optimisées informatives, pour limiter le nombre de règles redondantes. Les règles optimisées informatives sont construites à partir de motifs libres et nous donnons un algorithme pour calculer ces règles. Elles permettent de présenter une synthèse non redondante de toutes les règles qui optimisent l'ensemble des SBMs et de les regénérer en cas de besoin.

Certains résultats plus forts sont obtenus dans le cas particulier des règles de classification. Notre cadre peut être simplifié ainsi que la procédure d'extraction des règles optimisées informatives. De surcroît, celles-ci forment une couverture de l'ensemble des règles optimisant les SBMs.

Des motifs minimaux aux traverses minimales

Notre troisième contribution donne un panorama élargi de l'intérêt des motifs minimaux en approfondissant les liens qui les unissent aux traverses minimales d'un hypergraphe. Il existe de nombreux travaux qui traitent des liens entre fouille de données et hypergraphes mais la plupart d'entre eux considèrent les hypergraphes comme un outil et ont pour objectif l'extraction de motifs. Au contraire, l'originalité de notre approche réside dans le fait d'utiliser les algorithmes d'extraction de motifs pour résoudre un problème portant sur les hypergraphes.

Nous commençons par immerger le domaine des hypergraphes dans celui de la fouille de données et le problème du calcul des traverses minimales est reformulé en termes d'extraction de motifs minimaux. Pour cela, une adaptation de la notion d'extension est nécessaire car elle doit dans ce contexte permettre de caractériser les traverses minimales. Notamment, la définition d'une connexion de Galois appropriée au contexte des hypergraphes est proposée. Nous fournissons une méthode efficace de calcul des traverses minimales d'un hypergraphe basée sur la méthode d'extraction de motifs δ-libres qui constitue notre première contribution. Les expériences menées comparent notre approche à deux autres méthodes connues et établissent qu'elle est particulièrement efficace sur des hypergraphes denses.

Usages des motifs minimaux

Nous avons enfin montré l'intérêt de nos résultats dans divers contextes applicatifs. Dans le domaine de l'analyse du transcriptome, notre méthode de découverte de règles de caractérisation δ-fortes a permis d'obtenir une vue globale des règles. Nous avons également pu pointer une règle significative contenant un gène (la protéine NIFIE14) que les biologistes soupçonnent d'être impliqué dans le développement du cancer. Les situations qui supportent cette règle concernent en majorité la prostate, permettant de mieux cerner l'influence du gène mis en cause.

Par ailleurs, les apports de notre méthode de calcul des traverses minimales d'un hypergraphe sont exposés à travers deux applications. Des expériences montrent comment l'utilisation de notre approche pourrait considérablement améliorer le calcul de la bordure positive d'un ensemble de motifs fréquents. En effet, ce calcul nécessite l'obtention des traverses minimales pour de nombreux hypergraphes denses, ce qui correspond au cas où MTMINER se révèle particulièrement efficace. Ensuite, nous décrivons une méthode de visualisation de données à partir de plusieurs clusterings et basée sur les traverses minimales pour laquelle MTMINER a été fructueusement utilisé.

Perspectives

Nos perspectives de recherche prennent appui sur les travaux menés en extraction de motifs et concernent, de façon plus générale, les liens qui unissent les domaines de la fouille de données

et des hypergraphes.

Découverte d'information dans les données larges

Il existe une grande variété de contraintes pour permettre à tout utilisateur d'exprimer l'intérêt d'un motif. Certaines sont, de même que la δ-liberté, délicates à traiter dans le contexte des données comportant un grand nombre d'attributs. Il semble donc naturel de vouloir étendre l'utilisation de l'extension à ces contraintes. De même, l'obtention de nouveaux critères d'élagage issus de la combinaison de la fréquence avec d'autres contraintes est une piste intéressante. La mise en œuvre de ces deux points permettrait d'élargir le champ des possibilités de découverte d'information dans les données d'expression de gènes et plus généralement, dans les données larges.

Évaluation des règles d'association

Certaines approches [FC03] s'intéressent à l'évaluation des règles d'association par l'utilisation conjointe de plusieurs mesures d'intérêt ; cette problématique de recherche est appelée *optimisation multi-critères*. Cela revient à garantir une valeur seuil pour une combinaison pondérée de plusieurs mesures. Notre cadre pour les mesures d'intérêt est fermé par combinaison linéaire (avec des coefficients positifs) et il permet d'obtenir simultanément des minorants pour un grand nombres de mesures d'intérêt. Nous pensons que cette approche fournit une réponse élégante à l'optimisation multi-critère puisque cette question revient à minorer l'une des mesures de notre cadre.

Lors de la conception d'un cadre pour les mesures d'intérêt, notre démarche a été guidée par l'obtention de minorants pour les SBMs, de manière à garantir une qualité élevée des règles extraites relativement à ces mesures. Nous n'avons pas évoqué la possibilité de majorer les valeurs des SBMs. En effet, les contraintes imposées par le cadre impliquent également l'existence de majorants. Or, ceci pourrait permettre d'encadrer la valeur moyenne des règles pour n'importe laquelle des SBMs. En fait, il serait ainsi possible de modéliser le comportement des règles d'association, avant de réaliser des extractions. Nous pensons que ceci pourrait guider l'utilisateur quant à la définition de seuils de fréquence et du nombre d'exceptions en fonction de la qualité attendue, mais aussi en fonction de la qualité qu'il est possible d'exiger avec ces paramètres.

Utilisation et analyse de notre méthode de calcul des traverses minimales

Nous avons testé notre algorithme de calcul des traverses minimales d'un hypergraphe pour le calcul de la bordure négative d'un ensemble de motifs fréquents à partir de sa bordure positive. Or, ce calcul ne représente qu'une étape des algorithmes d'extraction de motifs fréquents qui parcourent l'espace de recherche en profondeur. Il serait donc intéressant d'intégrer notre méthode de calcul des traverses minimales d'un hypergraphe à de tels algorithmes de manière à évaluer le gain en efficacité apporté par MTMINER pour la totalité de cette procédure. C'est relativement aisé dans le cas de l'algorithme de Gunopulos *et al.* [GKM$^+$03] qui sépare la recherche de bordure en une phase d'extraction de motifs et une phase de génération des traverses minimales mais cela l'est beaucoup moins pour l'agorithme de Satoh et Uno [SU03] dans lequel ces deux étapes sont fusionnées.

Les motifs k-libres [CG03] (différents des motifs δ-libres) constituent une généralisation des motifs libres qui permet de produire des règles concluant sur une disjonction d'attributs. La construction de telles règles nécessite le calcul de traverses minimales de longueur bornée [Rio05].

Puisque l'algorithme de calcul des traverses minimales que nous avons proposé fonctionne suivant le principe des algorithmes par niveaux, il est facilement adaptable au calcul de ces traverses appelées fermetures généralisées.

La complexité de notre algorithme de calcul des traverses minimales d'un hypergraphe dans le pire cas se révèle exponentielle en la taille du plus long transversal minimal. Pourtant, les expériences ont montré que cette méthode était efficace en pratique, en particulier sur des instances très denses. C'est pour expliquer ce résultat que nous souhaiterions, en collaboration avec Loïck Lhote (GREYC), réaliser une étude théorique du comportement de notre algorithme dans le cas moyen. Nous pensons qu'une telle étude refléterait plus finement et de manière plus réaliste, l'efficacité de notre méthode. Pour cela, nous disposons déjà d'outils d'analyse puisque Lhote *et al.* ont proposé une modélisation de bases de données qui débouche sur une estimation du nombre moyen de motifs fréquents dans un jeu de données dans de récents travaux [LRS05]. Ceux-ci seraient un bon point de départ puisque MTminer est basé sur des méthodes de type « extraction de motifs ».

Perspectives générales

Plus généralement, nous pensons que l'étude des liens entre les hypergraphes et les bases de données constitue une perspective de recherche particulièrement riche. Cette étude peut être envisagée sous plusieurs angles. Tout d'abord, d'autres « objets » remarquables dans un hypergraphe peuvent peut-être être identifiés à des motifs vérifiant une contrainte connue en fouille de données. Les deux domaines peuvent donc s'enrichir mutuellement dans l'exploitation de leurs savoir-faire propres. Ensuite, il existe de nombreuses données qu'il est judicieux de représenter sous la forme d'un hypergraphe (des interactions entre protéines, des réseaux mobiles, des segments de texte imbriqués, etc.). La fouille de données structurées a récemment exercé un attrait considérable en ECBD. Plus particulièrement, la fouille de graphes a connu un essort remarquable ces dernières années. Cependant, une de ses limitations réside dans l'obligation de se restreindre à des relations binaires, d'où un fossé entre la représentation utilisée et les processus applicatifs réellement mis en jeu. La fouille d'hypergraphes semble être une voie prometteuse puisqu'elle permettrait de combler ce fossé et de fouiller des structures modélisées par des relations n-aires.

Annexes

Annexe A

Exemple

Pour faciliter la lecture de ce mémoire, nous rappelons au tableau A.1 l'exemple donné au chapitre d'introduction dans le tableau 1 (page 4).

		Gènes (Attributs)							
		a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
Patients (Objets)	o_1	1	0	1	0	1	0	1	0
	o_2	0	1	1	0	1	0	1	0
	o_3	1	0	1	0	1	0	0	1
	o_4	1	0	0	1	0	1	0	1
	o_5	0	1	1	0	0	1	0	1
	o_6	0	1	1	0	1	1	0	1

TAB. A.1 – Un exemple $\mathcal{D}$ de données larges.

Annexe B

Preuves pour le chapitre 4

La section B.1 de cette annexe contient la preuve d'appartenance à l'ensemble des SBMs (cf. chapitre 4) pour le Rule-Interest. On montre à la section B.2 pourquoi le Relative Risk n'est pas une SBM et à quelle condition cette mesure vérifie la propriété P4 de notre cadre.

B.1 Preuve montrant que le Rule-Interest est une SBM

Le Rule-Interest [PS91] est défini de la manière suivante pour une règle d'association $r : X \rightarrow Y$:

$$RI(r) = \frac{\mathcal{F}(XY) \times |\mathcal{D}| - \mathcal{F}(X) \times \mathcal{F}(Y)}{|\mathcal{D}|}$$

La fonction associée et la fonction δ-dépendante sont égales à :

$$\Psi_{RI}(x, y, z) = \frac{z \times |\mathcal{D}| - x \times y}{|\mathcal{D}|}$$

$$\Psi_{RI,\delta}(x, y) = \frac{x \times (|\mathcal{D}| - y) - \delta \times |\mathcal{D}|}{|\mathcal{D}|}$$

$\Psi_{RI}(x, y, z)$ est clairement croissante en z, RI vérifie donc P2'. De plus, elle est décroissante en y puisque x est toujours positif (x est la fréquence de la prémisse de r) donc RI satisfait P3'. $\Psi_{RI,\delta}(x, y)$ est croissante en x car on a toujours $y = \mathcal{F}(Y) \leq |\mathcal{D}|$ donc RI vérifie P4.

Conclusion : Le Rule-Interest est une SBM.

B.2 Condition à laquelle le Relative Risk vérifie P4

Le Relative Risk [GH06] est défini comme suit pour une règle d'association $r : X \rightarrow Y$:

$$RR(r) = \frac{\mathcal{F}(XY)}{\mathcal{F}(X)} \times \frac{|\mathcal{D}| - \mathcal{F}(X)}{\mathcal{F}(Y) - \mathcal{F}(XY)}$$

La fonction associée et la fonction δ-dépendante sont égales à :

$$\Psi_{RR}(x, y, z) = \frac{z}{x} \times \frac{|\mathcal{D}| - x}{y - z}$$

$$\Psi_{RR,\delta}(x,y) = \frac{x-\delta}{x} \times \frac{|\mathcal{D}|-x}{y-x+\delta}$$

Il est aisé de vérifier que $\Psi_{RR}(x,y,z)$ est croissante en z et décroissante en y. Le Relative Risk satisfait donc P2' et P3'. Cependant, $\Psi_{RR,\delta}(x,y)$ n'est pas toujours croissante en x. En effet, $\Psi_{RR,\delta}(x,y)$ peut être réécrite de la manière suivante :

$$\Psi_{RR,\delta}(x,y) = (1-\frac{\delta}{x}) \times (1+\frac{|\mathcal{D}|-y-\delta}{y-x+\delta})$$

$1-\frac{\delta}{x}$ est une fonction croissante en x. $1+\frac{|\mathcal{D}|-y-\delta}{y-x+\delta}$ est croissante en x si et seulement si $|\mathcal{D}|-y-\delta$ est positif ce qui est équivalent à $|\mathcal{D}| \geq y+\delta$. Cette condition est souvent vraie puisque la variable y correspond à la fréquence de la conclusion d'une règle et que δ (c'est le nombre d'exceptions de la règle) est petit en pratique. Cependant, on ne peut assurer que ce soit toujours le cas donc le Relative Risk ne vérifie pas P4.

Conclusion : Le Relative Risk n'est pas une SBM.

Bibliographie

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487–499. Morgan Kaufmann, 1994.

[Azé03] Jérôme Azé. *Extraction de Connaissances à partir de Données Numériques et Textuelles*. PhD thesis, Université de Paris-Sud, France, 2003.

[BA99] Roberto J. Bayardo and Rakesh Agrawal. Mining the most interesting rules. In Brij M. Masand and Myra Spiliopoulou, editors, *proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 145–154, San Diego, California, USA, 1999. ACM Press.

[Bay04] Roberto J. Bayardo. The hows, whys, and whens of constraints in itemset and rule discovery. In Jean-François Boulicaut, Luc De Raedt, and Heikki Mannila, editors, *proceedings of the European Workshop on Inductive Databases and Constraint Based Mining*, volume 3848 of *Lecture Notes in Computer Science*, pages 1–13, Hinterzarten, Germany, 2004. Springer.

[BB00] Jean-François Boulicaut and Artur Bykowski. Frequent closures as a concise representation for binary data mining. In Takao Terano, Huan Liu, and Arbee L. P. Chen, editors, *proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00)*, volume 1805 of *Lecture Notes in Computer Science*, pages 62–73, Kyoto, Japan, 2000. Springer.

[BBJ+02] Céline Becquet, Sylvain Blachon, Baptiste Jeudy, Jean-Francois Boulicaut, and Olivier Gandrillon. Strong-association-rule mining for large-scale gene-expression data analysis : a case study on human sage data. *Genome Biology*, 3(12), 2002.

[BBR00] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Approximation of frequency queries by means of free-sets. In Djamel A. Zighed, Henryk Jan Komorowski, and Jan M. Zytkow, editors, *proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'00)*, volume 1910 of *Lecture Notes in Computer Science*, pages 75–85, Lyon, France, 2000. Springer.

[BBR03] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Free-sets : a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery Journal (DMKD)*, 7(1) :5–22, 2003.

[BEGK03] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Leonid Khachiyan. An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals. In Giuseppe Di Battista and Uri Zwick, editors, *proceedings of the 11th Annual European Symposium on Algorithms (ESA'03)*, volume 2832 of *Lecture Notes in Computer Science*, pages 556–567, Budapest, Hungary, 2003. Springer.

[Ber89] Claude Berge. *Hypergraphs : Combinatorics of Finite Sets*. North-Holland, Amsterdam, 1989.

[BG03] Elena Baralis and Paolo Garza. Majority classification by means of association rules. In Nada Lavrac, Dragan Gamberger, Hendrik Blockeel, and Ljupco Todorovski, editors, *proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2003)*, pages 35–46, Cavtat-Dubrovnik, Croatia, 2003. Springer.

[Bir48] Garett Birkhoff. *Lattice theory*. Number 25 in Colloquium Publications. American Mathematical Society, 1948. Deuxième édition.

[Bla05] Julien Blanchard. *Un système de visualisation pour l'extraction, l'évaluation, et l'exploration interactives des règles d'association*. PhD thesis, Université de Nantes, France, 2005.

[Bla07] Sylvain Blachon. *Exploration des données SAGE par des techniques de fouille de données en vue d'extraire des groupes de synexpression impliqués dans l'oncogénèse*. PhD thesis, Institut National des Sciences Appliquées de Lyon, France, 2007.

[BMR03] James Bailey, Thomas Manoukian, and Kotagiri Ramamohanarao. A fast algorithm for computing hypergraph transversals and its application in mining emerging patterns. In *proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 485–488, Melbourne, Florida, USA, 2003. IEEE Computer Society.

[BMS97] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets : Generalizing association rules to correlations. In Joan Peckham, editor, *proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 265–276, Tucson, Arizona, 1997. ACM Press.

[BPT+00] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In John W. Lloyd, Verónica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luís Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey, editors, *proceedings of the 1st International Conference on Computational Logic (CL'00)*, volume 1861 of *Lecture Notes in Computer Science*, pages 972–986, London, UK, 2000. Springer.

[BR01] Artur Bykowski and Christophe Rigotti. A condensed representation to find frequent patterns. In *proceedings of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'01)*, Santa Barbara, California, USA, 2001. ACM Press.

[BTP+02] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Pascal : un algorithme d'extraction des motifs fréquents. *Technique et Science Informatiques*, 21(1), 2002.

[BVW03] Tom Brijs, Koen Vanhoof, and Geert Wets. Defining interestigness for association rules. *International Journal Information Theories & Applications*, 10(4), 2003.

[CB02] Bruno Crémilleux and Jean-François Boulicaut. Simplest rules characterizing classes generated by δ-free sets. In *Proceedings of the 22nd International Conference on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 33–46, Cambridge, UK, 2002. Springer.

[CCL05] Alain Casali, Rosine Cicchetti, and Lotfi Lakhal. Essential patterns : A perfect cover of frequent patterns. In A. Min Tjoa and Juan Trujillo, editors, *proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'05)*, volume 3589 of *Lecture Notes in Computer Science*, pages 428–437, Copenhagen, Denmark, 2005. Springer.

[CG02] Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)*, volume 2431 of *Lecture Notes in Computer Science*, pages 74–85, Helsinki, Finland, 2002. Springer.

[CG03] Toon Calders and Bart Goethals. Minimal k-free representations of frequent sets. In Nada Lavrac, Dragan Gamberger, Hendrik Blockeel, and Ljupco Todorovski, editors, *proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03)*, volume 2838 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 2003.

[CG05] Toon Calders and Bart Goethals. Depth-first non-derivable itemset mining. In Hillol Kargupta, Jaideep Srivastava, Chandrika Kamath, and Arnold Goodman, editors, *proceedings of the 2005 SIAM International Conference on Data Mining (SDM'05)*, Newport Beach, California, 2005. SIAM.

[CRB04] Toon Calders, Christophe Rigotti, and Jean-François Boulicaut. A survey on condensed representations for frequent sets. In Jean-François Boulicaut, Luc De Raedt, and Heikki Mannila, editors, *proceedings of the European Workshop on Constraint-Based Mining and Inductive Databases*, volume 3848 of *Lecture Notes in Computer Science*, pages 64–80. Springer, 2004.

[Dam06] Peter Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3) :337–350, 2006.

[DCS06] Nicolas Durand, Bruno Crémilleux, and Einoshin Suzuki. Visualizing transactional data with multiple clusterings for knowledge discovery. In Floriana Esposito, Zbigniew W. Ras, Donato Malerba, and Giovanni Semeraro, editors, *proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS'06)*, volume 4203 of *Lecture Notes in Computer Science*, pages 47–57, Bari, Italy, 2006. Springer.

[DL99] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns : Discovering trends and differences. In Brij M. Masand and Myra Spiliopoulou, editors, *proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 43–52, San Diego, California, USA, 1999. ACM Press.

[DL05] Guozhu Dong and Jinyan Li. Mining border descriptions of emerging patterns from dataset pairs. *Knowledge and Information Systems*, 8(2) :178–202, 2005.

[DW02] Klaus Denecke and Shelly L. Wismath. *Universal Algebra and Applications in Theoretical Computer Science*. Chapman and Hall/CRC, 2002.

[EG95] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing archive*, 24(6) :1278–1304, 1995.

[EG02] Thomas Eiter and Georg Gottlob. Hypergraph transversal computation and related problems in logic and ai. In Sergio Flesca, Sergio Greco, Nicola Leone, and Giovambattista Ianni, editors, *proceedings of the 8th European Conference on Logic in Artificial Intelligence (JELIA'02)*, volume 2424 of *Lecture Notes in Computer Science*, pages 549–564, Cosenza, Italy, 2002. Springer.

[ER59] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6 :290–297, 1959.

[FC03] Dominique Francisci and Martine Collard. Multi-criteria evaluation of interesting dependencies according to a data mining approach. In *proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, pages 1568–1574, Canberra, Australia, 2003. IEEE Press.

[FF05] Johannes Fürnkranz and Peter A. Flach. Roc 'n' rule learning-towards a better understanding of covering algorithms. *Machine Learning*, 58(1) :39–77, 2005.

[FK96] Michael L. Fredman and Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3) :618–628, 1996.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge discovery and data mining : Towards a unifying framework. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *proceedings of the 2nd international conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, 1996.

[Fre99] Alex A. Freitas. On rule interestingness measures. *Knowledge Based Systems*, 12(5-6) :309–315, 1999.

[Gan84] Bernhard Ganter. Two basic algorithms in concept analysis. Technical report, Technical University of Darmstadt, Germany, 1984.

[GD86] Jean-Louis Guigues and Vincent Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 95 :5–18, 1986.

[GH06] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining : A survey. *ACM Computing Surveys*, 38(3), 2006.

[GK99] Vladimir Gurvich and Leonid Khachiyan. On generating the irredundant conjunctive and disjunctive normal forms of monotone boolean functions. *Discrete Applied Mathematics*, 96-97 :363–373, 1999.

[GKM+03] Dimitrios Gunopulos, Roni Khardon, Heikki Mannila, Sanjeev Saluja, Hannu Toivonen, and Ram Sewak Sharm. Discovering all most specific sentences. *ACM Transactional Database System*, 28(2) :140–174, 2003.

[GKMT97] Dimitrios Gunopulos, Roni Khardon, Heikki Mannila, and Hannu Toivonen. Data mining, hypergraph transversals, and machine learning. In *proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97)*, pages 209–216. ACM Press, 1997.

[GLW+04] Zhongmin Guo, Jürgen F. Linn, Guojun Wu, Sarah L Anzick, Claus F Eisenberger, Sarel Halachmi, Yoram Cohen, Alexey Fomenkov, Mohammad Obaidul Hoque, Kenji Okami, Gabriel Steiner, James M Engles, Motonabu Osada, Chulso Moon,

Edward Ratovitski, Jeffrey M Trent, Paul S Meltzer, William H Westra, Lambertus A Kiemeney, Mark P Schoenberg, David Sidransky, and Barry Trink. $Cdc_{91}l_1$ (pig-u) is a newly discovered oncogene in human bladder cancer. *Nature Medicine*, 10 :374–381, 2004.

[GMB85] Hector Garcia-Molina and Daniel Barbará. How to assign votes in a distributed system. *Journal of the ACM*, 32(4) :841–860, 1985.

[GMH+05] Jeremy R. Graff, Ann M. McNulty, Kimberly Ross Hanna, Bruce W. Konicek, Rebecca L. Lynch, Spring N. Bailey, Crystal Banks, Andrew Capen, Robin Goode, Jason E. Lewis, Lillian Sams, Karen L. Huss, Robert M. Campbell, Philip W. Iversen, Blake Lee Neubauer, Thomas J. Brown, Luna Musib, Sandaruwan Geeganage, and Donald Thornton. The protein kinase cβ-selective inhibitor, enzastaurin (ly317615.hcl), suppresses signaling through the akt pathway, induces apoptosis, and suppresses growth of human colon cancer and glioblastoma xenografts. *Cancer Research*, 65(6) :7462–7469, 2005.

[GMS97] Dimitrios Gunopulos, Heikki Mannila, and Sanjeev Saluja. Discovering all most specific sentences by randomized algorithms. In Foto N. Afrati and Phokion G. Kolaitis, editors, *proceedings of the 6th International Conference on Database Theory (ICDT'97)*, volume 1186 of *Lecture Notes in Computer Science*, pages 215–229. Springer, 1997.

[Gui00] Sylvie Guillaume. *Traitement des données volumineuses, mesures et algorithmes d'extraction de règles d'association et de règles ordinales.* PhD thesis, Université de Nantes, France, 2000.

[GYNS06] Ghada Gasmi, Sadok Ben Yahia, Engelbert Mephu Nguifo, and Yahia Slimani. $\mathcal{IGB}$: une nouvelle base générique informative des règles d'association. *Information, Interaction et Intelligence (I3)*, 6(1) :31–67, 2006.

[Hag07] Matthias Hagen. Lower bounds for three algorithms for the transversal hypergraph generation. In *proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG'07)*, Lecture Notes in Computer Science, Dornburg near Jena, Germany, 2007. Springer. To appear.

[HBC07] Céline Hébert, Alain Bretto, and Bruno Crémilleux. A data mining formalization to improve hypergraph minimal transversal computation. *Fundamenta Informaticae*, 79, 2007. To appear.

[HC05] Céline Hébert and Bruno Crémilleux. Mining frequent δ-free patterns in large databases. In Achim G. Hoffmann, Hiroshi Motoda, and Tobias Scheffer, editors, *proceedings of the 8th International Conference on Discovery Science (DS'05)*, volume 3735 of *Lecture Notes in Computer Science*, pages 124–136, Singapore, Singapore, 2005. Springer.

[HC06a] Céline Hébert and Bruno Crémilleux. Optimized rule mining through a unified framework for interestingness measures. In A Min Tjoa and Juan Trujillo, editors, *proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'06)*, volume 4081 of *Lecture Notes in Computer Science*, pages 238–247, Krakow, Poland, 2006. Springer.

[HC06b] Céline Hébert and Bruno Crémilleux. Obtention de règles optimisant un ensemble de mesures. In *proceedings of the Conférence francophone sur l'apprentissage automatique (CAp'06)*, pages 1–16, Trégastel, France, 2006. Presses Universitaires de Grenoble.

[HC07] Céline Hébert and Bruno Crémilleux. A unified view of objective interestingness measures. In Petra Perner, editor, *proceedings of the 5th International Conference on Machine Learning and Data Mining (MLDM´07)*, Lecture Notes in Computer Science, Leipzig, Germany, 2007. Springer. To appear.

[HF02] Karen M. Hajra and Eric R. Fearon. Cadherin and catenin alterations in human cancer. *Genes, Chromosomes and Cancer*, 34(3) :255–268, 2002.

[HH99] Robert J. Hilderman and Howard J. Hamilton. Knowledge discovery and interestingness measures : A survey. Technical report, University of Regina, 1999. CS 99-04.

[HH03] Robert J. Hilderman and Howard J. Hamilton. Measuring the interestingness of discovered knowledge : A principled approach. *Intelligent Data Analysis*, 7(4) :347–382, 2003.

[HK03] Christoph Helma and Stefan Kramer. A survey of the predictive toxicology challenge 2000-2001. *Bioinformatics*, 19(10) :1179–1182, 2003.

[HKKM98] Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Hypergraph based clustering in high-dimensional data sets : A summary of results. *IEEE Data Engineering Bulletin*, 21(1) :15–22, 1998.

[HT05] S. Hirano and S. Tsumoto. Guide to the hepatitis data. In *Discovery Challenge on hepatitis data co-located with the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*, pages 120–124, Porto, Portugal, October 2005.

[HYS05] Tarek Hamrouni, Sadok Ben Yahia, and Yahya Slimani. Prince : An algorithm for generating rule bases without closure computations. In A. Min Tjoa and Juan Trujillo, editors, *proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'05)*, volume 3589 of *Lecture Notes in Computer Science*, pages 346–355, Copenhagen, Denmark, 2005. Springer.

[JR04] Baptiste Jeudy and François Rioult. Database transposition for constrained (closed) pattern mining. In Bart Goethals and Arno Siebes, editors, *proceedings of the 3rd International Workshop on Knowledge Discovery inInductive Databases (KDID'03)*, volume 3377 of *Lecture Notes in Computer Science*, pages 89–107, Pisa, Italy, 2004. Springer.

[KBEG05] Leonid Khachiyan, Endre Boros, Khaled M. Elbassioni, and Vladimir Gurvich. A new algorithm for the hypergraph transversal problem. In *proceedings of the 11th International Computing and Combinatorics Conference (COCOON'05)*, volume 3595 of *Lecture Notes in Computer Science*, pages 767–776, Kunming, China, 2005. Springer.

[KMD+02] R. N. Korotkina, G. N. Matskevich, A. Sh. Devlikanova, A. A. Vishnevskii, A. G. Kunitsyn, and A. A. Karelin. Activity of glutathione-metabolizing and antioxidant enzymes in malignant and benign tumors of human lungs. *Bulletin of Experimental Biology and Medicine*, 133(6) :606–608, 2002.

[KMR+94] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In *proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 401–407, Gaithersburg, Maryland, 1994. ACM Press.

[Kry02] Marzena Kryszkiewicz. Concise representations of association rules. In David J. Hand, Niall M. Adams, and Richard J. Bolton, editors, *proceedings of the ESF Exploratory Workshop Pattern Detection and Discovery*, volume 2447 of *Lecture Notes in Computer Science*, pages 92–109, London, UK, 2002. Springer.

[KS05] Dimitris J. Kavvadias and Elias C. Stavropoulos. An efficient algorithm for the transversal hypergraph generation. *Journal of Graph Algorithms and Applications*, 9(2) :239–264, 2005.

[LFZ99] Nada Lavrac, Peter Flach, and Blaz Zupan. Rule evaluation measures : a unifying view. In *proceedings of the 9th international Workshop on Inductive Logic Programming (ILP'99)*, volume 1634 of *Lecture notes in artificial intelligence*, pages 174–185, Bled, Slovenia, 1999. Springer-Verlag.

[LHM98] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In Gregory Piatetsky-Shapiro Rakesh Agrawal, Paul E. Stolorz, editor, *proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 80–86, New York City, New York, USA, 1998. AAAI Press.

[LRS05] Loïck Lhote, François Rioult, and Arnaud Soulet. Average number of frequent (closed) patterns in bernoulli and markovian databases. In Jiawei Han, Benjamin W. Wah, Vijay Raghavan, Xindong Wu, and Rajeev Rastogi, editors, *proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 713–716, Houston, Texas, USA, 2005. IEEE Computer Society.

[LSW97] Brian Lent, Arun N. Swami, and Jennifer Widom. Clustering association rules. In W. A. Gray and Per-Åke Larson, editors, *proceedings of 13th International Conference on Data Engineering (ICDE'97)*, pages 220–231, Birmigham, UK, 1997. IEEE Computer Society.

[LT04] S. Lallich and O. Teytaud. Evaluation et validation de l'intérêt des règles d'association. *Revue des Nouvelles Technologies de l'Information*, pages 193–218, 2004.

[Lux91] Michael Luxenburger. Implications partielles dans un contexte. *Mathématiques et Sciences Humaines*, 113 :35–55, 1991.

[McG05] Ken McGarry. A survey of interestingness measures for knowledge discovery. *The Knowledge Engineering Review*, 20 :39–61, 2005.

[Mit82] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2) :203–226, 1982.

[MM95] John A. Major and John J. Mangano. Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems*, 4(1), 1995.

[MT97] Heikki Mannila and Hannu Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3) :241–258, 1997.

[NFED04] MA. Nasr, MJ. Fedele, K. Esser, and AM. Diamond. Gpx-1 modulates akt and p70s6k phosphorylation and gadd45 levels in mcf-7 cells. *Free Radical Biology and Medicine*, 37(2) :187–195, 2004.

[NLHP98] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained association rules. In Laura M. Haas and Ashutosh Tiwary, editors, *proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 13–24. ACM Press, 1998.

[PBTL99a] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *proceedings of the 7th International Conference on Database Theory (ICDT'99)*, volume 1540 of *Lecture Notes in Computer Science*, pages 398–416, Jerusalem, Israel, 1999. Springer.

[PBTL99b] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1) :25–46, 1999.

[PCT+03] Feng Pan, Gao Cong, Anthony K. H. Tung, Jiong Yang, and Mohammed J. Zaki. Carpenter : Finding closed patterns in long biological datasets. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 637–642, Washington, DC, USA, 2003. ACM.

[PHM00] Jian Pei, Jiawei Han, and Runying Mao. Closet : An efficient algorithm for mining frequent closed itemsets. In *proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[PNSL06] Marie Plasse, Ndeye Niang, Gilbert Saporta, and Laurent Leblond. Une comparaison de certains indices de pertinence des règles d'association. In *actes des 6èmes Journées Francophones « Extraction et Gestion des Connaissances » (EGC'06)*, pages 561–568, 2006.

[PS91] Gregory Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules.* AAAI/MIT Press, 1991.

[RBCB03] François Rioult, Jean-François Boulicaut, Bruno Crémilleux, and Jérémie Besson. Using transposition for pattern discovery from microarray data. In Mohammed J. Zaki and Charu C. Aggarwal, editors, *proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD03)*, pages 73–79, San Diego, USA, 2003. ACM Press.

[RC03] François Rioult and Bruno Crémilleux. Condensed representations in presence of missing values. In Michael R. Berthold, Hans-Joachim Lenz, Elizabeth Bradley, Rudolf Kruse, and Christian Borgelt, editors, *proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA'03)*, volume 2810 of *Lecture Notes in Computer Science*, pages 578–588, Berlin, Germany, 2003. Springer.

[Rio05] François Rioult. *Extraction de connaissances dans les bases de données comportant des valeurs manquantes ou un grand nombre d'attributs.* PhD thesis, Université de Caen, France, 2005.

[RMCW01] Tannishtha Reya, Sean J. Morrison, Michael F. Clarke, and Irving L. Weissman. Stem cells, cancer, and cancer stem cells. *Nature*, 414(6859) :105–111, 2001.

[Sou06] Arnaud Soulet. *Un cadre générique de découverte de motifs sous contraintes fondées sur des primitives.* PhD thesis, Université de Caen, France, 2006.

[SS98] Saswati Sarkar and Kumar N. Sivarajan. Hypergraph models for cellular mobile communication systems. *IEEE Transactions on Vehicular Technology*, 47(2) :460 – 471, 1998.

[ST96] Abraham Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6) :970–974, 1996.

[SU03] Ken Satoh and Takeaki Uno. Enumerating maximal frequent sets using irredundant dualization. In Gunter Grieser, Yuzuru Tanaka, and Akihiro Yamamoto, editors, *proceedings of the 6th International Conference on Discovery Science (DS'03)*, volume 2843 of *Lecture Notes in Computer Science*, pages 256–268, Sapporo, Japan, 2003. Springer.

[Suz03] Einoshin Suzuki. Undirected discovery of interesting exception rules. *International Journal of Pattern Recognition and Artificial Intelligence*, 16 :1065–1086, 2003.

[Sza06] Laszlo Szathmary. *Méthodes symboliques de fouille de données avec la plate-forme Coron*. PhD thesis, Université Henri Poincaré, Nancy 1, France, 2006.

[TKS02] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *proceedings of the eighth ACM SIGKDD international conference on Knowledge Discovery and Data mining (KDD'02)*, pages 32–41, Edmonton, Alberta, Canada, 2002. ACM Press.

[VLL04] Benoît Vaillant, Philippe Lenca, and Stéphane Lallich. A clustering of interestingness measures. In Einoshin Suzuki and Setsuo Arikawa, editors, *proceedings of the 7th International Conference on Discovery Science (DS'04)*, volume 3245 of *Lecture Notes in Computer Science*, pages 290–297, Padova, Italy, 2004. Springer.

[VMP+05] Benoît Vaillant, Patrick Meyer, Elie Prudhomme, Stéphane Lallich, Philippe Lenca, and Sébastien Bigaret. Mesurer l'intérêt des règles d'association. In *actes des 5èmes Journées Francophones « Extraction et Gestion des Connaissances » (EGC'05)*, pages 421–426, 2005.

[VZVK95] Victor E. Velculescu, Lin Zhang, Bert Vogelstein, and Kenneth W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235) :484–487, 1995.

[Wah04] Magnus Wahlström. Exact algorithms for finding minimum transversals in rank-3 hypergraphs. *Journal of Algorithms*, 51(2) :107–121, 2004.

[Wil82] Rudolf Wille. *Ordered sets*. Reidel, Dordrecht-Boston, 1982.

[Zak00a] Mohammed J. Zaki. Generating non-redundant association rules. In *proceedings of the 6th ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD'00)*, pages 34–43, Boston, Massachusetts, USA, 2000. ACM Press.

[Zak00b] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3) :372–390, 2000.

[Zak04] Mohammed J. Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9(3) :223–248, 2004.

[ZG03] Mohammed J. Zaki and Karam Gouda. Fast vertical mining using diffsets. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 326–335, Washington, DC, USA, 2003. ACM.

[ZH02] Mohammed J. Zaki and Ching-Jiu Hsiao. Charm : An efficient algorithm for closed itemset mining. In Robert L. Grossman, Jiawei Han, Vipin Kumar, Heikki Mannila, and Rajeev Motwani, editors, *proceedings of the 2th SIAM international conference on Data Mining (SDM'02)*, pages 33–43, Arlington, USA, 2002. SIAM.

Résumé : La découverte et l'interprétation de motifs et de règles sont deux tâches centrales en extraction de connaissances dans les bases de données. Ce mémoire traite de l'extraction et des usages de motifs minimaux à la fois en fouille de données et dans le domaine des hypergraphes. D'une part, nous proposons une méthode efficace pour la découverte de motifs δ-libres dans les données larges, malgré les difficultés algorithmiques inhérentes à ce type de données. Cette méthode repose sur l'utilisation de l'extension des motifs et d'un nouveau critère d'élagage. D'autre part, nous nous intéressons à la qualité des règles d'association et nous présentons un cadre générique qui permet de mieux comprendre les similarités et différences entre mesures. Il montre que de nombreuses mesures (appelées SBMs pour Simultaneously Bounded Measures) ont des comportements proches. Ce résultat permet de garantir des valeurs minimales pour toutes les SBMs et la production de règles de qualité par rapport à l'ensemble de ces mesures. Enfin, l'apport des méthodes de type « extraction de motifs » pour d'autres domaines est mis en évidence. Nous montrons que notre approche de découverte de motifs dans les données larges est exploitable pour calculer efficacement les traverses minimales d'un hypergraphe, un problème réputé comme particulièrement difficile. Différentes applications, notamment en biologie, montrent l'intérêt pratique de nos méthodes.

Mots-clés : Fouille de données, générateurs minimaux, mesures d'intérêt, hypergraphes, traverses minimales.

Title: Mining and using minimal patterns in data mining and hypergraphs.

Abstract: Pattern discovery is a significant field of Knowledge Discovery in Databases. This work deals with mining and using minimal generators (also called free or key patterns). First, we propose an efficient algorithm for mining δ-free patterns in large databases. This is a difficult task due to the huge search space. We present a new approach based on pattern extension and a new pruning criterion. Second, we provide a unified view of objective interestingness measures. We design a framework capturing the main features of interestingness measures and we prove that a large set of usual measures, called SBMs, behave in a similar way. We also give an algorithm to efficiently mine non-redundant rules simultaneously optimizing all the SBMs by using the free patterns. Finally, we deepen the relationships between data mining and hypergraphs. We show how to exploit the key ideas of our extension-based method for efficiently computing the minimal transversals of a hypergraph which is known as a very hard problem. Experiments prove that our methods are very efficient in practice and useful for various applications.

Keywords: Knowledge Discovery in Databases, Data Mining, minimal generators, interestingness measures, hypergraphs, minimal transversals.

Discipline : Informatique

Laboratoire : Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen - UMR 6072, Université de Caen/Basse-Normandie, France