Extraction de connaissances dans les bases de donn'ees comportant des valeurs manquantes ou un grand nombre d'attributs

François Rioult

Département d'Informatique
UFR Sciences

École doctorale SIMEM

Extraction de connaissances dans les bases de données comportant des valeurs manquantes ou un grand nombre d'attributs

THÈSE

présentée et soutenue publiquement le 24 novembre 2005

pour l'obtention du

Doctorat de l'Université de Caen Basse-Normandie

(spécialité Informatique)

par

François Rioult

Composition du jury

Directeurs :	Bruno Crémilleux	Professeur	Université de Caen Basse-Normandie
	Étienne Grandjean	Professeur	Université de Caen Basse-Normandie
Rapporteurs :	Dominique Laurent	Professeur	Université de Cergy-Pontoise
	Gerd Stumme	Professeur	Université de Kassel (Allemagne)
Examinateurs :	Christophe Rigotti	Maître de conférences - H.D.R.	INSA de Lyon ISTI-C.N.R. de Pise (Italie)
	Jean-Daniel Zucker	Professeur	Université de Paris XIII

Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen — CNRS UMR 6072

Mis en page avec la classe thloria.

À mon père,
ce chercheur que j'ai peu connu.

Remerciements

En premier lieu, j'aimerais exprimer toute ma gratitude et ma reconnaissance à Bruno Crémilleux qui a dirigé ce travail et m'a permis de m'épanouir professionnellement. Ce manuscrit doit beaucoup à ses patientes relectures. Il m'a toujours chaleureusement accueilli et mené sur le chemin de l'autonomie. À son contact, j'ai beaucoup profité de sa finesse d'analyse, tant lors de l'élaboration de réflexions scientifiques que pour la gestion des aspects politiques inhérents à toute activité de recherche.

Je suis particulièrement heureux que ce travail ait été mené sous la responsabilité d'Étienne Grandjean. Il m'a initié aux passionnantes problématiques de la complexité en informatique et a su guider mes réflexions scientifiques à chacun de nos entretiens.

Ces recherches ont été financées par l'Unité IRM du CHU de Caen, le comité de la Manche de la Ligue contre le Cancer et le Conseil Régional de Basse-Normandie. Elles n'auraient pu aboutir sans la disponibilité de Jean-Marc Constans, responsable de l'Unité IRM du CHU de Caen, qui malgré la charge de son emploi du temps a toujours été présent et constructif. J'associe à ces remerciements Michel-Henry Amar, responsable de l'unité de recherche clinique du centre de lutte contre le cancer François Baclesse à Caen.

Toute ma gratitude va à Dominique Laurent et Gerd Stumme, qui ont patiemment relu et rapporté ce manuscrit. Par sa présence, Gerd Stumme me permet de concrétiser scientifiquement la relation privilégiée que j'entretiens depuis de nombreuses années avec son pays, l'Allemagne.

Je remercie Jean-Daniel Zucker et Christophe Rigotti d'avoir accepté d'être membres du jury. Christophe Rigotti a patiemment suivi et encouragé mes efforts depuis leurs débuts.

Une partie de ce travail est le résultat d'une collaboration avec Arnaud Soulet, Loïck Lhote et Frédérick Houben du GREYC, ainsi qu'avec Baptiste Jeudy du laboratoire EURISE de l'Université de Saint-Étienne. Je les remercie pour les relations tant amicales que scientifiques que nous entretenons.

Enfin, je souhaite exprimer ma reconnaissance à l'ensemble du personnel du GREYC et du département d'Informatique de l'Université de Caen Basse-Normandie. Patrice Enjalbert, Brigitte Vallée et Khaldoun Zreik m'ont permis de revenir avec bonheur au sein du monde universitaire. Jean-Jacques Hébrard a su m'orienter avec justesse et confiance vers la fouille de données. Merci également à Alain Bretto, Régis Carin, Jerzy Karczmarczuk, François Kauffman, Jacques Madeleine, Anne Nicolle, Françoise Quiquemelle, André Sesbouë et à toute l'équipe d'administration système : Davy Gigan, Vincent Lenouvel, Philippe May, Hélène Rousset et Véronique Robert.

Je remercie tout spécialement Pierre Renaux, Céline Hébert, Nicolas Durand, Frédérick Houben, Antoine Widlöcher ainsi que Bruno Zanuttini pour leur présence quotidienne et soutenue, amicale et scientifique, pendant l'élaboration et la rédaction de mon travail.

Ce travail n'aurait pas pu voir le jour sans la patience et la compréhension de ma famille. Je remercie affectueusement ma femme Sylvie Huguenin, entre autres pour ses nombreuses et minutieuses relectures et mes enfants pour leur indulgence...

Table des matières

Table des figures

Liste des tableaux

Index thématique des notations utilisées

Le signe d'union $\cup$ est omis. La notation XY est préférée à $X \cup Y$.

$\mathcal{A}$		l'ensemble des attributs (ou items)
a	$\in \mathcal{A}$	un attribut (ou item) de $\mathcal{A}$
m	$= \lvert\mathcal{A}\rvert$	le nombre d'attributs
$\mathcal{O}$		l'ensemble des objets (ou transactions)
o	$\in \mathcal{O}$	un objet (ou transaction)
n	$= \lvert\mathcal{O}\rvert$	le nombre d'objets
R	$\subseteq \mathcal{A} \times \mathcal{O}$	une relation binaire
r	$= (\mathcal{A}, \mathcal{O}, R)$	une base de données
$\mathcal{L}_{\mathcal{A}}$	$= 2^{\mathcal{A}}$	ensemble (langage) des motifs d'attributs
$\mathcal{L}_{\mathcal{O}}$	$= 2^{\mathcal{O}}$	ensemble (langage) des motifs d'objets
A	$\subseteq \mathcal{A}$	un motif d'attributs
E		un motif d'attributs utilisé comme paramètre
X	$\subseteq \mathcal{A}$	un motif d'attributs, désignant la prémisse d'une règle
Y	$\subseteq \mathcal{A}$	un motif d'attributs, désignant la conclusion d'une règle
$X\overline{Y}$		un motif généralisé
O	$\subseteq \mathcal{O}$	un motif d'objets
Z	$= X \cup Y$	un motif d'attributs utilisé pour construire une règle
$X \rightarrow Y$		une règle d'association
$X \rightarrow \vee Y$		une règle d'association généralisée
r_X		la base de données restreinte aux objets contenant le motif X
$supp(X)$		l'ensemble des objets contenant le motif X
$\mathcal{F}(X)$	$= \lvert supp(X)\rvert$	la fréquence du motif X
γ	$\leq n$	un seuil minimum de fréquence
δ	$\leq n$	un nombre d'exceptions tolérées pour une règle d'association
k	$\leq m$	une profondeur de règle d'association généralisée

$mv()$		l'opérateur de modélisation des valeurs manquantes
$mv(r)$	$= (\mathcal{A}, \mathcal{O}, mv(R))$	une base de données avec valeurs manquantes, issue de r complète
$mv(r)^\circ$		la base $mv(r)$ dont les valeurs manquantes ont été neutralisées
$Des(X)$		ensemble des objets désactivés pour au moins un attribut de X
$Des(X \rightarrow \vee Y)$		ensemble des objets désactivés pour $X \rightarrow \vee Y$
$Des(\wedge Y)$		ensemble des objets désactivés pour tous les attributs de Y
${}^-r$	$= (\mathcal{A}, \mathcal{O}, {}^-R)$	la base opposée de r (${}^-R$ est la relation opposée de R)
${}^t r$	$= (\mathcal{O}, \mathcal{A}, {}^t R)$	la base transposée de r
$\mathcal{C},\ q$	$:\ \mathcal{L}_\mathcal{A} \rightarrow \{0,1\}$	une contrainte ou un prédicat sur un motif d'attributs
$\mathcal{C}_{\gamma\text{-freq}},\ \mathcal{C}_{\mathsf{close}}$		contraintes de fréquence, de fermeture
$\mathcal{C}_{\subseteq E}$		contrainte de sous ensemble de E
$\mathcal{C}_{\supseteq E}$		contrainte de sur ensemble de E
S	$\subseteq \mathcal{L}_\mathcal{A}$	un ensemble de motifs d'attributs satisfaisant une contrainte
$\overline{S}$		l'ensemble des complémentaires de S
$Tr(S)$		les traverses minimales de S
$Tr_k(S)$		les traverses minimales de S de longueur bornée par k
$\mathcal{B}d^+(S)$		l'ensemble des maximaux de S
$\mathcal{B}d^-(S)$	$= Tr(\overline{S})$	l'ensemble des minimaux de $\mathcal{L}_\mathcal{A} \backslash S$
f	$:\ \mathcal{A} \rightarrow \mathcal{O}$	l'opérateur d'intension de Galois
g	$:\ \mathcal{O} \rightarrow \mathcal{A}$	l'opérateur d'extension de Galois
h	$= f \circ g$	l'opérateur de fermeture de Galois sur les attributs
h'	$= g \circ f$	l'opérateur de fermeture de Galois sur les objets
c	$= (X, T)$	un concept, où X et T sont des motifs fermés d'attributs et d'objets
${}^t f$	$= g$	l'opérateur d'intension dans ${}^t r$
${}^t g$	$= f$	l'opérateur d'extension dans ${}^t r$
${}^t c$	$= (T, X)$	un concept transposé, où T et X sont des motifs fermés d'objets et d'attributs
${}^t \mathcal{C}$		la contrainte transposée de $\mathcal{C}$

Introduction générale

Contexte

L'Extraction de Connaissances dans les Bases de Données (E.C.B.D.) ou *fouille de données* (data-mining en anglais) est une discipline récente, à l'intersection des domaines des bases de données, de l'intelligence artificielle, de la statistique, des interfaces homme/machine et de la visualisation. À partir de données collectées par des experts, il s'agit de proposer des connaissances nouvelles qui enrichissent les interprétations du champ d'application, tout en fournissant des méthodes automatiques qui exploitent cette information.

L'ECBD est classiquement décrite comme un processus interactif de préparation des données (sélection de descripteurs, constitution d'une table, discrétisation), d'extraction de connaissances à l'aide d'algorithmes de calcul, de visualisation et d'interprétation des résultats, lors d'interactions avec l'expert [Fayyad *et al.*, 1996] (voir figure 1). Les méthodes d'exploration proposent des solutions aux problèmes de recherche d'associations, de classification supervisée et non supervisée.

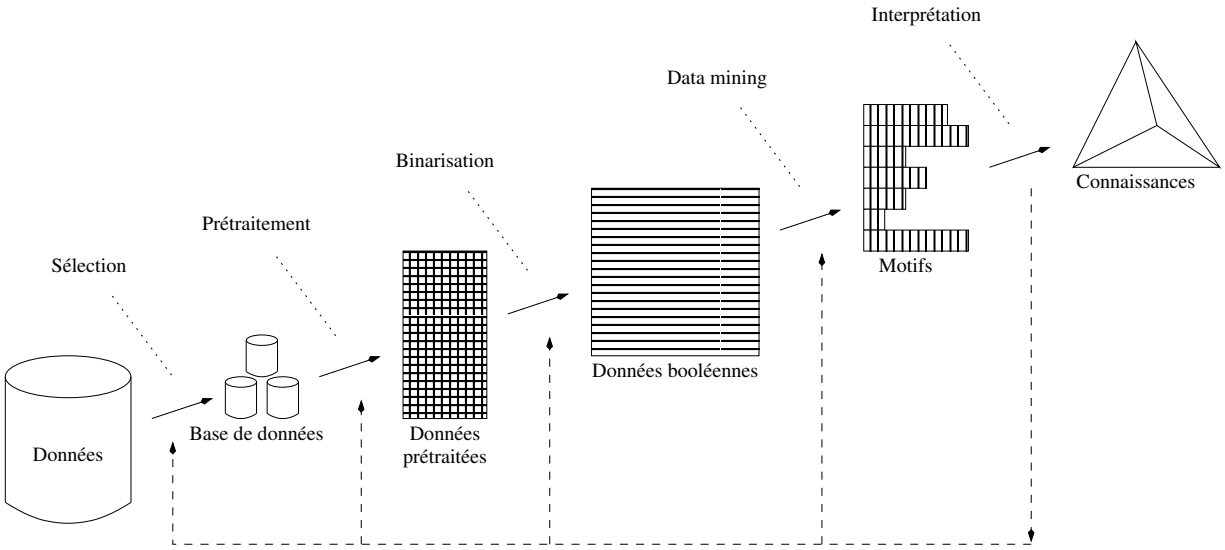

FIG. 1 – Processus d'extraction de connaissances.

Fréquemment exprimée sous forme de règles, la connaissance extraite requiert la mise au point d'algorithmes efficaces pour prendre en compte les difficultés algorithmiques ou liées aux caractéristiques du problème. Les bases de données utilisées comprennent la description de millions d'objets par des milliers d'attributs et l'espace de recherche est de taille exponentielle en nombre d'attributs. Plusieurs problèmes NP-difficiles se cachent en particulier derrière la recherche des motifs fréquents (ensembles d'attributs communs à plusieurs objets), étape préalable à la construction de règles associant des motifs.

Dans ce mémoire, nous focalisons plus particulièrement notre intérêt sur la mise au point d'algorithmes d'extraction de motifs dans des contextes difficiles (valeurs manquantes, grand nombre d'attributs) et sur leurs usages dans l'application de méthodes d'exploration.

Motivations

La fouille de données arrive maintenant à maturité sur les contextes d'extraction classiques pour lesquels les algorithmes ont été mis au point à l'origine de la discipline. Les bases de données commerciales, qui décrivent les achats réalisés par des millions de clients sur des milliers de références sont désormais parfaitement exploitées par les spécialistes et les méthodes fondées sur les règles d'association connaissent des développements de plus en plus vastes. Ces techniques se popularisent dans les domaines médicaux, économiques ou industriels.

Les technologies se sont également affinées et il est aujourd'hui courant d'extraire des motifs qui satisfont des contraintes particulières ou des règles aux propriétés bien identifiées afin d'éviter la profonde redondance des résultats élémentaires. Les spécialistes du domaine se tournent maintenant vers l'obtention d'un résumé de la connaissance utile, sous forme de représentations condensées, plutôt que l'exploitation de l'intégralité des résultats obtenus. Ces représentations apportent un double avantage. D'une part elles sont plus faciles à extraire que la connaissance de base qu'elles synthétisent, d'autre part leurs développements pour les méthodes d'exploration des données sont nombreux. Néanmoins, certains domaines d'application particuliers peinent à bénéficier de ces évolutions, du fait de contraintes structurelles particulières. Dans ce travail, nous nous intéressons à deux de ces problèmes : les données incomplètes, entachées de valeurs manquantes, et les données très « larges », comportant un grand nombre d'attributs devant le nombre d'objets.

Le premier problème est classique dans le domaine des bases de données. Celles-ci proviennent généralement de processus réels d'acquisition, concernant par exemple des données médicales humaines ou des résultats de sondages. Dans ce cadre, il n'est pas toujours possible d'obtenir une mesure relative à un examen qui n'a pas été pratiqué (par exemple quand le patient n'est pas en état de le supporter) ; ou la réponse à une question posée, car les sondés n'ont pas toujours une opinion à exprimer (ou ne le souhaitent pas) ni la patience de le faire. Ces données sont

donc régulièrement entachées de valeurs manquantes. Si la communauté des bases de données a produit de nombreux travaux sur le sujet [Dyreson, 1997], les contributions dans le domaine de la fouille de données sont plus rares.

Le deuxième problème concerne les bases de données aux dimensions inhabituelles, comportant nettement plus d'attributs que d'objets. Cette configuration rend difficile voire impossible l'application des algorithmes classiques, conçus pour traiter les grandes quantités d'objets. Par exemple, le domaine scientifique émergent de l'analyse du génome produit des données sur des dizaines de milliers de gènes, mais les expériences biologiques qui déterminent la séquence de gènes sont coûteuses donc peu nombreuses. Si les techniques de fouille de données se sont largement démocratisées, elles peinent à prendre en compte les bases de données comportant un grand nombre d'attributs.

Contributions

Notre travail utilise la notion de motif *k-libre*, une propriété centrale pour le calcul de représentations condensées ou la construction de règles d'association généralisées. La notion de règle d'association généralisée *informative* est définie. La prémisse d'une telle règle est un motif k-libre et sa conclusion est un élément de la fermeture généralisée de la prémisse.

Pour faire face aux valeurs manquantes, nous préconisons un mode de calcul original pour déterminer si un motif est k-libre. À l'aide de la notion de désactivation partielle et temporaire d'objet, nous mettons en évidence une relation entre le support d'un motif dans la base incomplète et son support dans la base complète dont elle est issue. Cette relation permet de caractériser la k-liberté d'un motif lorsqu'il y a des valeurs manquantes. Elle est importante pour la recherche d'associations généralisées informatives car la prémisse d'une règle est validée par une définition consistante avec la base complète. La conclusion de ces règles est obtenue en utilisant la base opposée. Les valeurs manquantes étant laissées invariantes par transformation de la base en base opposée, nous combinons cette propriété avec la consistance des prémisses afin de proposer des conclusions pertinentes pour les règles d'association généralisées informatives. Nous analysons l'intérêt de cette pratique.

Pour les données comportant un grand nombre d'attributs devant le nombre d'objets, nous introduisons la transposition de base de données et montrons que les concepts d'une base coïncident avec les concepts de la base transposée. Cette utilisation simple de la connexion de Galois offre des conditions plus praticables pour appliquer directement les classiques algorithmes de calcul de concepts. Sous cette forme, les extractions de motifs fréquents, dans des bases de données génomiques par exemple, se révèlent possibles.

Nous proposons aussi un cadre complet de transposition de contraintes, afin de calculer les motifs contraints dans les larges jeux de données. La méthode consiste à extraire les motifs fermés

dans la base transposée, selon une contrainte relative aux objets, équivalente à la contrainte originale sur les attributs. Les motifs satisfaisant la contrainte originale et qui ne sont pas fermés sont regénérés.

Pour mener à bien ce travail, nous avons développé un ensemble d'outils logiciels qui permettent de couvrir l'intégralité d'un processus de fouille de données. Des applications menées avec des propriétaires de données montrent l'intérêt pratique de notre travail.

Organisation du mémoire

La première partie de ce mémoire dresse un état de l'art des techniques d'extraction de connaissances et de leur difficulté. Cet état de l'art permet de mieux comprendre les notions développées par la suite.

Nous présentons au chapitre 1 un cadre formel générique pour l'obtention des motifs fréquents et plus généralement des motifs qui satisfont une contrainte antimonotone, qui définissent les fondements du calcul des règles d'association. Pour cela, nous détaillons les classiques algorithmes par niveaux et les algorithmes de recherche en profondeur.

Le chapitre 2 explique comment les représentations condensées permettent de pallier les difficultés algorithmiques de l'extraction des motifs fréquents. Les motifs fermés et la connexion de Galois sont introduits, ainsi que les motifs libres et δ-libres.

Le chapitre 3 généralise le principe des motifs libres en présentant les motifs non dérivables ou k-libres. Ces motifs libres généralisés expriment l'absence de corrélation entre les attributs qui les composent et constituent les prémisses des règles généralisées informatives.

Le chapitre 4 termine cette première partie en contribuant à l'étude de la complexité de l'extraction de motifs. Une analyse en moyenne du nombre de motifs fréquents ou fermés est décrite, qui s'attache au dénombrement dans le cas général plutôt que dans le pire des cas. Ce chapitre propose une vision nouvelle de la difficulté des tâches d'extraction de motifs.

La deuxième partie présente notre travail sur l'obtention de motifs k-libres et de règles d'associations généralisées informatives depuis des bases de données incomplètes. Cette partie commence par le chapitre 5, où l'état de l'art sur le traitement des valeurs manquantes détermine notre point de vue et nos objectifs précis.

Au chapitre 6, nous définissons un mode de calcul original pour exhiber les motifs k-libres d'une base contenant des données manquantes. Nous montrons que cette méthode est consistante puisqu'elle permet à partir de calculs menés dans une base incomplète, d'obtenir des motifs qui ont cette propriété dans la base complète dont la base d'étude est issue.

Nous améliorons la portée de ce résultat au chapitre 7 en montrant son apport pour la construction de règles d'association généralisées correctes. Celles-ci sont issues d'extractions me-

nées à la fois dans la base et son opposée. La comparaison des règles obtenues dans les deux contextes permet de construire des règles d'association dont la prémisse est un motif k-libre dans toute base complète d'origine et la conclusion est pertinente en présence de valeurs manquantes.

La troisième partie de ce document étudie l'extraction des motifs fréquents dans les contextes comportant un grand nombre d'attributs. Dans le chapitre 8, nous montrons que cette extraction est facilitée par l'utilisation conjointe de la transposition de données et de la connexion de Galois.

Dans le chapitre 9, nous étendons ce principe à l'extraction de motifs sous contraintes. La définition d'un cadre formel pour la transposition de contraintes permet des extractions sous contraintes dans des contextes comportant un grand nombre d'attributs grâce à une extraction contrainte relaxée dans le contexte transposé.

La dernière partie de notre travail est consacrée aux applications de nos résultats dans le cadre de coopérations avec des experts du domaine médical et biologique. Le chapitre 10 introduit les usages des motifs k-libres en terme de méthodes d'exploration de données. Nous y présentons le calcul des motifs émergents forts, les règles de caractérisation de classe et de classification.

Le chapitre 11 décrit les outils logiciels que nous avons mis au point pour expérimenter et vérifier nos propriétés. Nous détaillons les extracteurs de motifs réalisés et les utilitaires qui constituent nos chaînes de traitement des données.

Le chapitre 12 relate nos expérimentations sur des données incomplètes relatives à la maladie de Hodgkin, un cancer des organes lymphatiques (ganglions). Les résultats concernent la mise en évidence de règles d'association généralisées correctes dans des données comportant un grand nombre de valeurs manquantes.

Pour terminer cette partie, nous relatons au chapitre 13 une étude de données d'expression de gènes, provenant du séquençage de situations biologiques humaines. Dans ces données comportant un grand nombre d'attributs, nous recherchons à l'aide des motifs émergents forts des associations de gènes caractéristiques d'une situation cancéreuse.

Le dernier chapitre dresse le bilan de notre travail. Les principaux résultats obtenus sont résumés et discutés. Nous proposons également quelques perspectives.

Première partie

Extraction de connaissances dans les bases de données

Introduction

Cette partie se concentre sur les difficultés d'ordre algorithmique de l'extraction de connaissances dans des bases de données et plus spécifiquement l'extraction de motifs fréquents, qui sont le pivot de nombreuses méthodes de fouille de données.

Dans cette partie préliminaire, nous décrivons un cadre formel général introduit par Mannila et Toivonen [Mannila et Toivonen, 1997] pour l'expression des algorithmes d'extraction des motifs fréquents et plus généralement sous contrainte antimonotone. Nous avons choisi cette méthode pour présenter une large variété de travaux relatifs à l'extraction de motifs car ce cadre permet de qualifier la difficulté algorithmique du problème et d'unifier les principales méthodes. Le chapitre 1 réserve pour cela une place importante à la présentation d'outils formels.

Nous approfondissons cette étude des algorithmes d'extraction en montrant que l'usage de représentations condensées de motifs réduit fortement la difficulté initiale du problème. En effet, celles-ci sont plus efficaces à extraire que les collections entières de motifs. De plus, elles permettent d'offrir un résumé des motifs intéressants et de leurs propriétés et proposent une vision synthétique de ces grandes collections de motifs. Les représentations condensées se révèlent précieuses pour la conception et le développement de méthodes de fouille de données. Le chapitre 2 introduit les motifs fermés, libres (ou clés) puis δ-libres, qui sont les plus classiques.

Le chapitre 3 est réservé à la description des motifs libres généralisés (ou k-libres), introduits par Calders et Goethals [Calders et Goethals, 2002]. À notre connaissance, ces motifs sont pour le moment peu utilisés et uniquement pour l'obtention des motifs fréquents. Une part importante de ce mémoire est consacrée à ces motifs en s'intéressant à leur extraction en présence de valeurs manquantes (partie II) et à leurs usages en production de règles d'association généralisées à des fins de caractérisation ou de classification.

Enfin, nous terminons cette partie par une contribution à l'étude de la complexité des tâches d'extraction de motifs fréquents ou fermés (chapitre 4), sous l'angle de l'analyse en moyenne du nombre de motifs satisfaisants.

Chapitre 1

Découverte de motifs fréquents

Ce chapitre présente un état de l'art sur l'extraction des motifs fréquents dans les bases de données. Cet état de l'art est non seulement nécessaire pour introduire et détailler les difficultés de l'extraction de connaissances, mais également pour comprendre la suite de notre travail.

Le cadre formel que nous reprenons ici a été introduit par Heikki Mannila et Hannu Toivonen [Mannila et Toivonen, 1997] pour la découverte de motifs fréquents dans une base de données. Nous avons choisi cette présentation car elle permet d'unifier la typologie des algorithmes connus selon le mode de production des motifs candidats, qui est la seule réelle difficulté de ce problème. Dans ce chapitre, nous nous concentrons sur la contrainte de fréquence car c'est la contrainte la plus classique qui parle aux experts des données. Elle fournit un premier niveau de connaissance et permet de nombreux usages, typiquement les règles d'association. Mais les algorithmes détaillés par la suite fonctionnent avec toute contrainte antimonotone et ne sont pas restreints à la découverte des motifs fréquents.

Après avoir rappelé quelques définitions sur les bases de données, nous exposons le cadre de Mannila et Toivonen. Puis nous détaillons les différentes instances de l'algorithme `Guess & Correct`, selon que les candidats sont produits par niveau (section 1.3.1), ou en profondeur (section 1.3.2). Enfin nous examinons la stratégie `Divide & Conquer` (section 1.3.3) et brièvement l'extraction sous contrainte (section 1.4).

1.1 Définitions

Les bases de données considérées ici sont de simples tables contenant l'information, éventuellement construites par jointures à partir de plusieurs relations. L'exemple du tableau 1.1 répertorie les valeurs de trois *attributs* multi-valués X_1, X_2 et X_3 pour 8 *objets* d'étude, appelés également n-uplets. Dans cet exemple, les deux premiers attributs X_1 et X_2 sont de type symbolique ou qualitatif car leur domaine de définition est discret. *A contrario*, le dernier attribut X_3 est numérique ou quantitatif.

	attributs		
objets	X_1	X_2	X_3
o_1	$+$	$\rightarrow$	0.2
o_2	$-$	$\rightarrow$	0
o_3	$+$	$\rightarrow$	$0,1$
o_4	$+$	$\leftarrow$	$0,4$
o_5	$-$	$\rightarrow$	$0,6$
o_6	$-$	$\rightarrow$	$0,5$
o_7	$+$	$\leftarrow$	1
o_8	$-$	$\leftarrow$	$0,8$

TAB. 1.1 – Exemple d'une base de données au format attribut/valeur.

Il peut arriver, dans certaines configurations, que des données soient manquantes (*i.e.* quand la valeur d'un attribut est inconnue). Nous renvoyons le lecteur à la partie II de ce document qui présente en détail notre contribution sur le sujet.

Ce mémoire se concentre sur l'extraction de *motifs* ensemblistes, où un motif est un ensemble d'attributs booléens ou *items*. Cela nécessite de discrétiser les attributs numériques, afin de disposer de données booléennes. Il sort du cadre de ce travail de discuter précisément des méthodes de discrétisation qui permettent d'obtenir de tels contextes booléens à partir d'attributs multi-valués ou continus. Le lecteur intéressé se référera à [Grzymala-Busse, 2002] pour un catalogue des méthodes élémentaires, à [Zighed *et al.*, 1999] pour un examen plus détaillé dans un cadre supervisé, à [Srikant et Agrawal, 1996] pour une approche dédiée à la recherche d'associations quantitatives et à [Ganter et Wille, 1989, Ganter et Wille, 1999] pour l'analyse des concepts formels. Disons simplement que cette étape de prétraitement des données est difficile dans le cas d'attributs numériques : il faut regrouper ensemble des valeurs différentes qui expriment la même information, ou définir des intervalles. Les connaissances des experts se révèlent indispensables pour effectuer les bons choix lors de cette opération délicate. Dans la partie IV où nous relatons nos résultats de fouilles sur des applications concrètes, nous exposerons au cas par cas les méthodes de discrétisation utilisées.

Revenons à notre exemple et supposons que cette étape de discrétisation fournit ici pour X_3 trois attributs a_5, a_6 et a_7 correspondant aux intervalles $[0-0,3],]0,3-0,7],]0,7-1]$. En recodant également les valeurs des attributs symboliques (a_1 pour « + », a_2 pour « - », a_3 pour « $\rightarrow$ », a_4 pour « $\leftarrow$ »), nous obtenons (tableau 1.2) une matrice booléenne qui indique pour chaque objet les attributs qu'il contient. Ce format est usuellement qualifié de *transactionnel*. Dans ces contextes booléens, un attribut est souvent appelé *item* et un objet *transaction*.

	attributs						
objets	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_1	×		×		×		
o_2		×	×		×		
o_3	×		×		×		
o_4	×			×		×	
o_5		×	×			×	
o_6		×	×			×	
o_7	×			×			×
o_8		×		×			×

TAB. 1.2 – Exemple d'une base de données au format transactionnel.

Nous donnons ci-dessous des définitions élémentaires pour la fouille de données.

Définition 1 (Contexte formel) *Une base de données booléennes r est notée sous la forme d'un contexte formel $(\mathcal{A}, \mathcal{O}, R)$ où $\mathcal{A} = \{a_1 \dots a_m\}$ est l'ensemble des attributs, $\mathcal{O} = \{o_1 \dots o_n\}$ celui des objets et R une relation binaire entre $\mathcal{A}$ et $\mathcal{O}$. R indique quels attributs a sont recensés dans les objets o et on notera de façon équivalente $R(a, o) = present$ pour aRo et $R(a, o) = absent$ pour $\neg aRo$.*

Notre travail porte sur l'extraction de motifs d'attributs dans les bases de données, mais nous parlons également de motifs d'objets dans ce document. Lorsqu'aucune précision n'est indiquée, un *motif* est un motif d'attributs.

Définition 2 (Motif)

Un motif d'attributs *est un sous ensemble de $\mathcal{A}$ (ou un élément du langage $\mathcal{L}_{\mathcal{A}} = 2^{\mathcal{A}}$).*

Un motif d'objets *est un sous ensemble de $\mathcal{O}$ (ou un élément du langage $\mathcal{L}_{\mathcal{O}} = 2^{\mathcal{O}}$).*

Pour alléger les écritures, les motifs seront notés sous forme de chaîne plutôt que sous forme d'ensembles (*i.e.* a_1a_2 au lieu de $\{a_1, a_2\}$). Dans un contexte formel, nous considérons indifféremment un objet o comme un élément de $\mathcal{O}$ ou comme un motif d'attributs : $o = \{a \in \mathcal{A} \mid R(a, o) = present\}$. Les notions de support et de fréquence d'un motif sont introduites comme suit :

Définition 3 (Support, fréquence) *Un objet $o \in \mathcal{O}$ supporte le motif d'attributs X (ou X est présent dans o) si $X \subseteq o$ (ou $\forall a \in X, R(a, o) = present$). Le support $supp(X)$ d'un motif*

X d'attributs est l'ensemble des objets qui le supportent. Sa fréquence $\mathcal{F}(X)$ est le cardinal du support[1].

La notion de motif *fréquent* est centrale pour la fouille de données :

Définition 4 (Motif fréquent) *Un motif d'attribut est fréquent (ou γ-fréquent) si sa fréquence dépasse un seuil γ fixé par l'utilisateur.*

Sur notre exemple, les objets o_4 et o_7 contiennent le motif a_1a_4, son support est o_4o_7. Sa fréquence vaut 2, et si le seuil de fréquence est fixé à 1 ou 2, ce motif est fréquent.

Nous définissons maintenant les règles d'association :

Définition 5 (Règle d'association) *Soit r une base de données et $Z \subseteq \mathcal{A}$ un motif d'attributs. Une règle d'association basée sur Z est une expression $X \rightarrow Y$ avec $X \subsetneq Z$ et $Y = Z \backslash X$. X est la prémisse, Y est la conclusion. La fréquence de la règle est celle de XY. La confiance, notée $conf(X \rightarrow Y)$, est la proportion d'objets contenant X qui contiennent aussi Y [Agrawal et Srikant, 1994] : $conf(X \rightarrow Y) = \mathcal{F}(X \cup Y)/\mathcal{F}(X)$.*

Une règle exacte dans r est une règle de confiance 1, i.e. nous notons $\models_r X \rightarrow Y$ quand $\mathcal{F}(X \cup Y) = \mathcal{F}(X)$.

Sur notre exemple, la règle $a_1a_3 \rightarrow a_5$ est exacte (les fréquences de a_1a_3 et de $a_1a_3a_5$ sont égales et valent 2). La règle $a_2a_3 \rightarrow a_6$ a une confiance de 2/3 ($\mathcal{F}(a_2a_3) = 3$ et $\mathcal{F}(a_2a_3a_6) = 2$).

Même si dans la pratique les possesseurs de bases sont plus intéressés par les règles présentes dans les données que par les motifs fréquents, celles-ci sont dépendantes de l'obtention de ces motifs. Le calcul de ces règles n'est pas difficile lorsque l'on connaît les motifs fréquents. L'extraction des motifs fréquents est clairement identifiée dans la communauté fouille de données comme l'étape algorithmiquement difficile préalable à la formation de règles [Rioult, 2004a] et nous y consacrons une large part dans la présentation qui suit.

1.2 Le cadre de Mannila et Toivonen

Nous avons choisi de décrire l'extraction des motifs fréquents à l'aide du formalisme unificateur développé par Heikki Mannila et Hannu Toivonen [Mannila et Toivonen, 1997]. En effet, les algorithmes qui s'acquittent de cette tâche utilisent les mécanismes de ce formalisme dans leur grande majorité : il s'agit d'un problème de recherche, dont l'espace à parcourir est progressivement élagué grâce aux propriétés du problème.

[1] Dans la littérature, le terme support est parfois utilisé pour désigner le nombre d'occurrences d'un motif (ce que nous avons appelé la fréquence), tandis que la fréquence est utilisée pour exprimer en pourcentage la fraction d'occurrences relativement au nombre total d'objets.

Nous décrivons donc l'espace de recherche associé à une base de données. Puis nous caractérisons la contrainte d'extraction souhaitée, dont les propriétés fournissent un critère d'élagage de l'espace de recherche. Après avoir détaillé la procédure de génération des nouveaux candidats, nous pouvons décrire l'algorithme `Guess & Correct` qui calcule les motifs d'une base de données satisfaisant une contrainte et déterminer sa complexité.

1.2.1 Espace de recherche

L'espace de recherche $\mathcal{L}_{\mathcal{A}}$ des motifs est le langage construit sur l'alphabet $\mathcal{A} = \{a_1, \ldots, a_7\}$ ($\mathcal{L}_{\mathcal{A}} = 2^{\mathcal{A}}$). Il se représente naturellement sous la forme d'un treillis d'inclusion (figure 1.1). En haut, partant de l'ensemble vide, la deuxième ligne contient les singletons, puis la troisième contient les paires, la quatrième les triplets, etc. Pour les premiers et derniers niveaux, cette quantité est faible mais elle grandit rapidement jusqu'au milieu du treillis où on rencontre le plus de motifs. Un losange est donc traditionnellement utilisé pour cette représentation car pour m attributs, il y a $\binom{m}{l}$ motifs de longueur l. Pour notre exemple, les objets issus du modèle attribut/valeur sont de longueur limitée à 3, donc il n'y a pas de motif de longueur supérieure à 3 dans ces données. En revanche, le treillis théorique descend jusqu'aux 7 motifs de longueur 6 et le dernier de longueur 7.

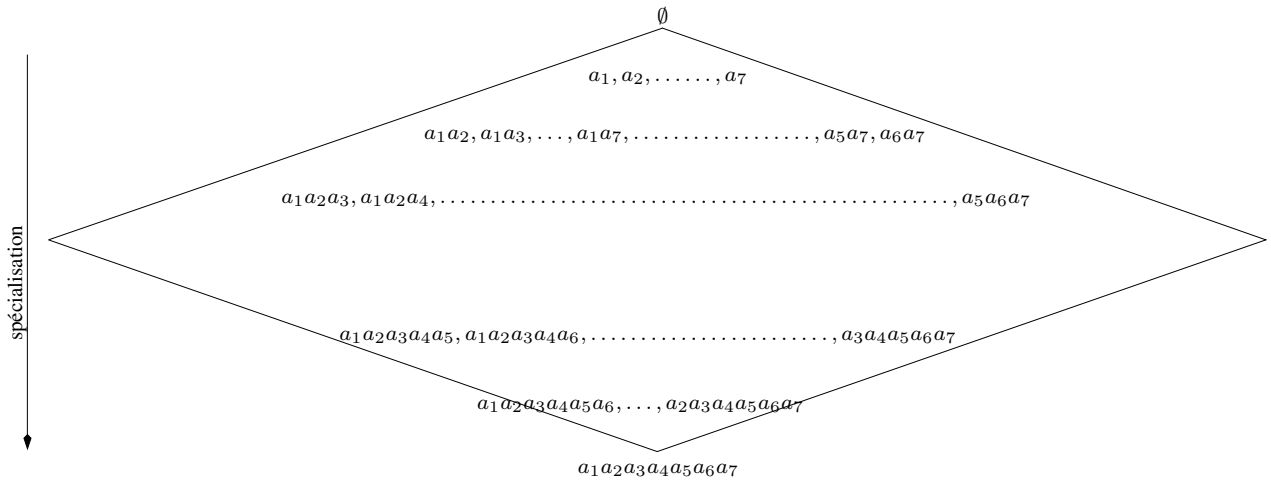

FIG. 1.1 – Treillis des motifs de la base exemple.

Le lien d'inclusion entre deux motifs définit une relation de spécialisation [Mitchell, 1980] et on parlera d'un motif plus spécifique ou plus général qu'un autre pour qualifier l'inclusion dans un sens ou l'autre. En effet, plus un motif contient d'attributs, moins il y a d'objets qui contiennent ce motif. La fraction de l'échantillon qu'il désigne est d'autant plus spécifique. Inversement, un motif contenant moins d'attributs désigne une population plus générale. Dans la suite, spécialiser un motif indiquera qu'on lui ajoute des attributs, le généraliser indiquera qu'on lui retire des

attributs.

La notion de treillis n'est pas fortuite car la recherche de motifs vérifiant une contrainte revient fondamentalement à un parcours de l'ensemble des parties de l'ensemble des attributs. Dans un treillis, tout couple de motifs possède une plus petite borne inférieure et une plus grande borne supérieure au sens de l'inclusion. Pour notre problème, le motif vide et le motif « plein » contenant tous les attributs de l'alphabet fournissent des bornes triviales. La relation de spécialisation, calquée sur l'inclusion, définit un pré-ordre sur le langage et plus précisément un treillis. Nous verrons à la section 2.2 l'utilisation de l'un des opérateurs de fermeture associés à cette relation.

1.2.2 Contrainte antimonotone : élagage de l'espace de recherche

Dans les applications réelles, les bases de données comportent couramment des milliers d'attributs. L'espace de recherche étant de taille exponentielle (pour m attributs binaires, il y a 2^m motifs différents), il est illusoire de le parcourir exhaustivement et nous devons avoir une stratégie. Celle-ci repose sur une propriété de la contrainte que les motifs recherchés doivent vérifier.

La fréquence d'un motif a une propriété remarquable : elle ne peut que diminuer quand on spécialise le motif. Sur notre exemple tableau 1.2, a_1a_4 a une fréquence de 2 tandis que $a_1a_4a_6$ n'est présent qu'une fois. La spécialisation diminue la proportion de l'échantillon concerné. Inversement, quand on généralise un motif, sa fréquence ne peut qu'augmenter. Ainsi, les généralisations d'un motif fréquent sont fréquentes également. Cette propriété est qualifiée d'*antimonotone* par rapport à la relation de spécialisation, par opposition à une propriété *monotone* :

Définition 6 (Monotonie) *Une contrainte binaire $q(X, r)$ (i.e. prédicat d'arité 2), qualifiant une propriété du motif X de $\mathcal{L}_\mathcal{A}$ relativement à une base de données r, est monotone suivant la spécialisation si et seulement si*

$$\forall X, Y \in \mathcal{L}_\mathcal{A},\ X \subseteq Y \Rightarrow (q(X, r) \Rightarrow q(Y, r)). \tag{1.1}$$

Définition 7 (Antimonotonie) *Une contrainte $q(X, r)$ est antimonotone suivant la spécialisation si et seulement si*

$$\forall X, Y \in \mathcal{L}_\mathcal{A},\ X \subseteq Y \Rightarrow (q(Y, r) \Rightarrow q(X, r)). \tag{1.2}$$

La contrainte de fréquence est antimonotone. Il existe beaucoup d'autres types de contraintes, pour la plupart monotones ou antimonotones, ou décomposables en une conjonction de contraintes monotones et antimonotones [De Raedt *et al.*, 2002]. On trouve par exemple des contraintes sur la longueur d'un motif, des contraintes syntaxiques indiquant qu'un motif en contient un autre ou est contenu dans un autre motif. Lorsqu'une valuation est associée aux attributs (par exemple

le prix d'un produit), des contraintes d'agrégat peuvent exprimer que le prix total des articles d'un motif est supérieur à un seuil, ou qualifient le prix maximum, la moyenne, etc. Les grand traits de l'extraction sous contrainte quelconque (*i.e.* non nécessairement (anti)monotone) sont rappelés section 1.4. Nous nous focalisons dans ce chapitre sur la contrainte de fréquence, mais les algorithmes détaillés dans la suite fonctionnent avec toute contrainte antimonotone.

L'antimonotonie fournit deux critères d'élagage pour un parcours optimisé de l'espace de recherche (voir figure 1.2) :

Critère 1 *L'utilisation de la contraposée de l'équation 1.2 correspond à l'usage le plus courant : si un motif X ne satisfait pas la contrainte, ses spécialisations ne la vérifient pas non plus. Lors du parcours de l'espace de recherche, on élaguera les branches issues de X.*

Critère 2 *L'équation 1.2 indique que tous les sous-ensembles d'un motif X qui vérifie la contrainte la vérifient également. De fait, si l'un des sous-ensembles d'un motif ne vérifie pas la contrainte, X ne peut pas la satisfaire.*

À partir d'un ensemble de motifs valides qui vérifient la contrainte, ces deux critères permettent de construire l'ensemble de leurs spécialisations qui vérifient *a priori* [Agrawal *et al.*, 1993] la contrainte. On désignera par le terme de *candidat* un motif qui vérifie les deux critères. Il est produit à partir de motifs qui vérifient la contrainte. En partant des singletons et en produisant petit à petit les candidats, il sera ainsi possible de progresser dans le treillis. Il faut ensuite tester la validité de la contrainte pour chaque motif en examinant les données. Muni de ces nouveaux motifs valides, on produit les nouveaux candidats et on itère le procédé.

La figure 1.2 détaille les effets des deux critères d'élagage lors de la production des candidats. Elle présente une situation où un motif Y est spécialisé en $Y \cup \{a_1\}$ et $Y \cup \{a_2\}$. Supposons que dans la base, $Y \cup \{a_1\}$ satisfait la contrainte mais pas $Y \cup \{a_2\}$ et que les candidats sont produits en parcourant l'ensemble des motifs valides dans un ordre précis. Le critère 2 empêchera ainsi la spécialisation de $Y \cup \{a_1\}$ en $Y \cup \{a_1, a_2\}$ car il contient un sous-ensemble $Y \cup \{a_2\}$ qui ne vérifie pas la contrainte. Ensuite, le critère 1 interdit toute spécialisation de $Y \cup \{a_2\}$.

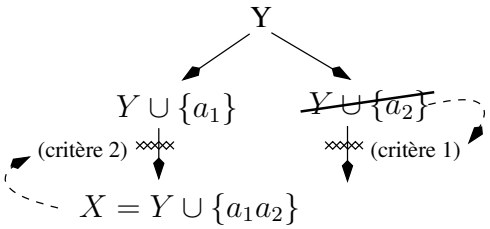

FIG. 1.2 – Critères d'élagage.

S'il est naturel d'interdire la génération de candidats depuis un motif non valide, il est moins immédiat de vérifier les sous-ensembles des candidats potentiels et cela complique l'algorithme. Cette étape est même qualifiée dans [Han *et al.*, 2000] de *goulet d'étranglement* de cette approche par génération de candidat. Dans [Bayardo *et al.*, 1999], l'algorithme DENSE-MINER utilise seulement le critère 1 et l'efficacité reste bonne. Malgré cela, l'usage simultané des deux critères et les économies importantes d'espace élagué ont fait le succès des approches à base de contraintes antimonotones. Même si ce sont les vérifications de contraintes dans la base qui sont les plus coûteuses en temps, la quantité de mémoire nécessaire au stockage des motifs valides, pour pouvoir effectuer la génération des candidats, peut rapidement dépasser les capacités usuellement disponibles. Aussi d'autres approches n'utilisant pas la génération de candidats doivent elles être envisagées (voir section 1.3.3 avec l'algorithme `Divide & Conquer`).

Nous restons cependant attaché aux méthodes à base de génération des candidats car elles permettent d'aborder une large variété d'algorithmes. Le concept de *bordure* est introduit et il permet de mettre en perspective cette génération selon des problèmes classiques de l'algorithmique.

1.2.3 Bordures positive et négative

Cette section précise l'étape de génération qu'il est nécessaire de mener pour explorer l'espace de recherche. À partir de motifs de même taille, cette génération est une chose aisée : il suffit de fusionner les motifs $Y \cup \{a_1\}$ et $Y \cup \{a_2\}$, qui partagent le préfixe commun Y et diffèrent d'un seul attribut ; le candidat potentiel sera $Y \cup \{a_1, a_2\}$. En revanche, ce procédé ne peut traiter des générateurs de longueurs hétérogènes ou dont le préfixe commun diffère de plus d'un attribut. Cette configuration spécifique des parcours en profondeur de l'espace de recherche sera détaillée section 1.3.2 et désignée par le terme d'*algorithme en profondeur*. *A contrario*, le terme d'*algorithme par niveaux* qualifie les méthodes utilisant des générateurs d'égale longueur et différant d'un seul attribut.

En vertu de l'antimonotonie de la contrainte, la recherche des éléments qui la vérifient peut s'assimiler à la découverte de la *frontière* qui sépare le treillis en deux parties : les motifs valides d'un côté, les motifs invalides de l'autre. Cette séparation en deux parties est possible car la contrainte est préservée par généralisation. Nous utiliserons par la suite le terme de bordure, proche du terme anglais *border* utilisé dans [Mannila et Toivonen, 1997], qui désigne l'union des bordures positive et négative. La bordure positive contient les motifs valides qui jouxtent la frontière : ce sont les maximaux valides au sens de l'inclusion, c'est-à-dire les plus longs. La bordure négative contient les motifs invalides qui sont de l'autre côté de la frontière : ce sont les minimaux invalides, ou les plus courts.

Le problème de génération des candidats est donc formalisé de la façon suivante : on dispose

d'un ensemble S de motifs valides (vérifiant la contrainte de recherche antimonotone, *i.e.* clos par sous-ensembles) et les candidats à produire doivent être plus spécifiques que les motifs de S, tout en vérifiant les deux critères d'élagage. Les sous-ensembles des candidats produits doivent par conséquent être dans S et la bordure négative de S rassemble les motifs qui satisfont cette propriété :

Définition 8 (Bordure négative) *Soit S un ensemble quelconque de motifs de $\mathcal{L}_\mathcal{A}$ satisfaisant une contrainte antimonotone q. La bordure négative de S, notée $\mathcal{B}d^-(S)$ est l'ensemble des motifs X de $\mathcal{L}_\mathcal{A}$ tels que*

$$X \in \mathcal{B}d^-(S) \iff X \in \mathcal{L}_\mathcal{A} \backslash S \text{ et } \forall\, Y \subsetneq X,\ q(Y,r) \tag{1.3}$$

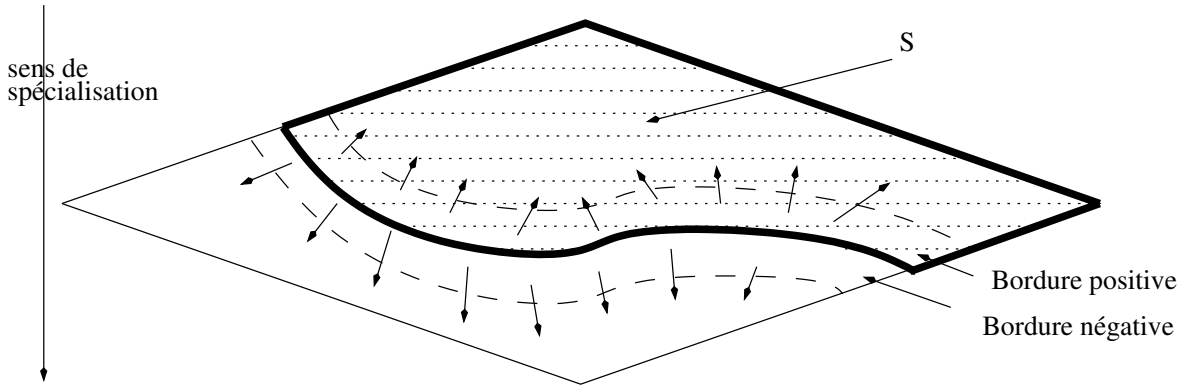

FIG. 1.3 – Bordures positive et négative.

La vision duale de la bordure négative est la bordure positive $\mathcal{B}d^+(S)$, qui rassemble les maximaux de S.

Définition 9 (Bordure positive) *Soit S un ensemble quelconque de motifs de $\mathcal{L}_\mathcal{A}$ satisfaisant une contrainte antimonotone q. La bordure positive de S, notée $\mathcal{B}d^+(S)$ est l'ensemble des motifs X de $\mathcal{L}_\mathcal{A}$ tels que*

$$X \in \mathcal{B}d^+(S) \iff q(X,r) \wedge (\forall Y,\ X \subsetneq Y \Rightarrow \neg q(Y,r)) \tag{1.4}$$

La figure 1.3 illustre ces deux notions. Sur le treillis, la bordure négative se représente comme les minimums de la zone complémentaire. La dualité exprime le fait que la bordure positive est elle-même la bordure négative de $\mathcal{L}_\mathcal{A}\backslash S$, quand on change la relation de spécialisation pour une relation de généralisation : $\mathcal{B}d^-(S)$ rassemble les minimaux de $\mathcal{L}_\mathcal{A}\backslash S$.

1.2.4 Traverses minimales d'hypergraphe

L'algorithme par niveaux doit régulièrement générer des candidats et les produit à l'aide de la bordure négative de l'ensemble temporaire des motifs valides. Nous présentons rapidement la notion d'hypergraphe, qui répond à ces attentes avec le calcul des traverses minimales.

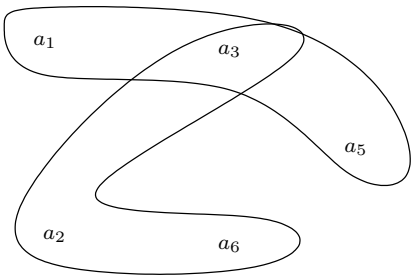

FIG. 1.4 – Exemple d'hypergraphe.

Si un graphe est une collection d'arêtes reliant ensemble deux sommets, un hypergraphe contient des hyperarêtes reliant un nombre quelconque de sommets. C'est une structure idéale pour représenter les motifs comme des hyperarêtes reliant les attributs. Par exemple, l'ensemble de motifs $\mathcal{E} = \{\{a_1, a_3, a_5\}, \{a_2, a_3, a_6\}\}$ sur l'alphabet $\mathcal{V} = \{a_1 \dots a_7\}$ donnera lieu à l'hypergraphe $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. Cet hypergraphe est tracé à la figure 1.4 comme on le ferait pour un graphe. Cette représentation est cependant peu lisible lorsqu'il y a de nombreuses hyperarêtes et on lui préférera une forme tabulaire.

Une *traverse* pour un hypergraphe est une hyperarête qui intersecte toutes les hyperarêtes de l'hypergraphe (le terme *hitting set* est employé en anglais). [Mannila et Toivonen, 1997] montre que si l'hypergraphe $H_{\overline{S}}$ représente l'ensemble des complémentaires des motifs de S par rapport à l'ensemble des attributs, alors une traverse pour $H_{\overline{S}}$ est un motif de $\mathcal{L}_{\mathcal{A}} \backslash S$. La figure 1.5 schématise ce procédé. Elle présente à gauche un ensemble de motifs en traits pleins, les complémentaires figurent à côté en traits pointillés. Un exemple de traverse prend un attribut dans chaque complémentaire. Si l'ensemble des traverses décrit $\mathcal{L}_{\mathcal{A}} \backslash S$ [Mannila et Toivonen, 1997], les *traverses minimales* indiquent les minimaux de $\mathcal{L}_{\mathcal{A}} \backslash S$, soit la bordure négative de S [Demetrovics et Thi, 1995, Mannila et Räihä, 1986]. En notant $Tr(H)$ l'ensemble des traverses minimales de H, cette propriété est la suivante :

Propriété 1 (Calcul de bordure par traverses minimales)

$$\mathcal{B}d^-(S) = Tr(\overline{S}) \tag{1.5}$$

Donnons un exemple : supposons que $S = \{a_1a_3, a_2a_3a_6\}$ (figure 1.5, à droite). Sur un alphabet allant jusqu'à a_7, l'ensemble des complémentaires est $\overline{S} = \{a_2a_4a_5a_6a_7, a_1a_4a_5a_7\}$. Les

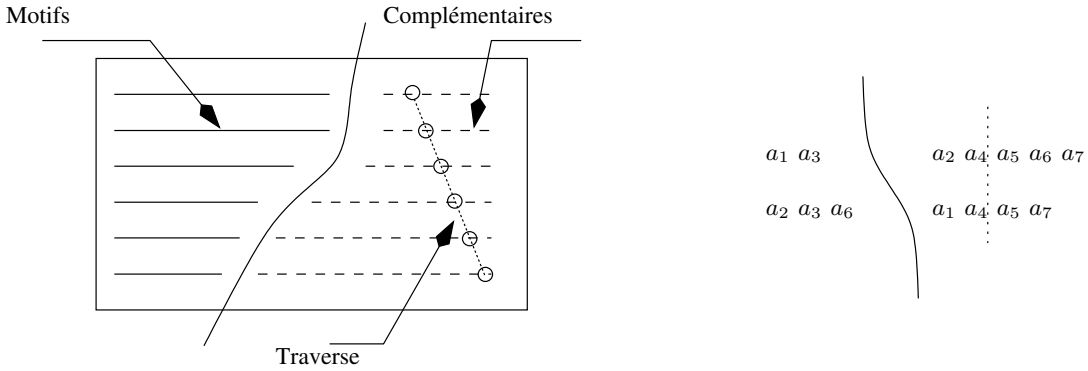

FIG. 1.5 – Traverses d'un hypergraphe.

traverses minimales les plus courtes sont a_4 (représentée sur la figure), a_5 et a_7 ; puis celles de longueur 2 sont a_1a_2 et a_1a_6. Au cours de la recherche des motifs fréquents, si S est l'ensemble temporaire des motifs maximaux, les candidats générés seront dans l'ensemble $\{a_4, a_5, a_7, a_1a_2, a_1a_6\}$. Ce cas est typique de la production de candidats à partir de générateurs de longueurs hétérogènes.

Il est intéressant de noter qu'un problème classique comme celui des traverses minimales est au cœur des algorithmes d'extraction de motif. Mais il s'agit bien plus qu'une élégante formalisation du problème de la recherche de bordure négative à l'aide de la théorie des hypergraphes [Berge, 1989]. Le problème de l'énumération des traverses minimales est en effet renommé [Eiter et Gottlob, 1995] et largement étudié. De nature NP car la vérification d'une solution s'effectue en temps polynomial, sa complexité exacte n'est pas connue. On dispose en revanche d'algorithmes quasi polynomiaux [Boros *et al.*, 2003, Eiter *et al.*, 2002], mais le problème reste difficile. À l'annexe 1, nous détaillons quelques algorithmes de calcul des traverses minimales. Nous verrons également au chapitre 3 que la résolution de ce problème contribue encore à la fouille de données en permettant de calculer les fermetures généralisées des motifs k-libres.

La génération de candidats de longueur hétérogène reste donc un challenge, c'est pourquoi certains algorithmes en profondeur utilisent plutôt une heuristique, tel `maxminer` [Bayardo, 1998]. La phase de génération s'en trouve simplifiée et accélérée, même si elle fournit plus de candidats que nécessaire. Nous reviendrons sur ce point à la section 1.3.2.

1.2.5 L'algorithme `Guess & Correct`

Nous disposons désormais de tous les outils formels nécessaires pour présenter l'algorithme générique `Guess & Correct` [Mannila et Toivonen, 1997], qui pour parcourir l'espace de recherche alterne des phases de production de candidats puis d'examen de la base pour tester ces candidats.

Les données d'entrée de cet algorithme sont la base de données r et une contrainte q antimonotone que doivent vérifier les motifs obtenus. On supposera également disposer d'un ensemble de motif S qui constitue une solution préliminaire potentielle au problème, sans présager de la validité de tous ses motifs. S provient par exemple d'un échantillonnage réalisé sur la base [Toivonen, 1996] et certains motifs issus d'une fraction de la base peuvent y avoir une fréquence plus élevée que dans l'ensemble de la base.

`Guess & Correct` procède, comme son nom l'indique, en deux étapes (voir algorithme 1) :

CORRECT : la solution provisoire S est corrigée par suppression de tous les motifs de S non valides,

GUESS : la solution est progressivement complétée pour intégrer les motifs valides manquants.

Données : une base de données $r = (\mathcal{A}, \mathcal{O}, R)$, une contrainte q antimonotone validant les motifs de $\mathcal{A}$ après examen de r, un ensemble S de motifs de $\mathcal{A}$. $\mathcal{C}and$ stocke les candidats et $\mathcal{E}$ mémorise les motifs déjà examinés.
Résultat : l'ensemble des motifs vérifiant q.
/*`phase` *correct* */
$\mathcal{C}and = \mathcal{B}d^+(S)$;
$\mathcal{E} = \emptyset$;
tant que $\mathcal{C}and$ *n'est pas vide* **faire**
  $\mathcal{E} = \mathcal{E} \cup \mathcal{C}and$;
  $S = S \backslash \{X \in \mathcal{C}and \ t.q. \ \neg q(X, r)\}$;
  $\mathcal{C}and = \mathcal{B}d^+(S) \backslash \mathcal{E}$;
fin
/*`phase` *guess* */
$\mathcal{C}and = \mathcal{B}d^-(S) \backslash \mathcal{E}$;
tant que $\mathcal{C}and$ *n'est pas vide* **faire**
  $\mathcal{E} = \mathcal{E} \cup \mathcal{C}and$;
  $S = S \cup \{X \in \mathcal{C}and \ t.q. \ q(X, r)\}$;
  $\mathcal{C}and = \mathcal{B}d^-(S) \backslash \mathcal{E}$;
fin
retourner S;

Algorithme 1 – `Guess & Correct`.

La plupart du temps, la solution temporaire S est vide au départ de l'algorithme. La phase *correct* n'est pas effectuée dans ce cas. Puis, au début de la phase *guess*, S ne contient que l'ensemble vide. Le complémentaire de l'ensemble vide est l'alphabet $\mathcal{A}$ et ses traverses minimales sont les singletons de $\mathcal{L}_\mathcal{A}$. Ainsi, `Guess & Correct` peut être directement initialisé avec les

candidats singletons.

Notons que `Guess & Correct` ne précise pas la façon dont sont produits les candidats : suivant des longueurs homogènes ou hétérogènes. Si la solution initiale est vide, les candidats auront les mêmes longueurs à chaque étape de *guess*, tandis qu'ils seront hétérogènes si on utilise une technique d'extension progressive ou si les motifs proviennent d'un échantillonnage.

1.2.6 Complexité

Précisons maintenant la complexité des méthodes issues de `Guess & Correct`. Nous introduisons pour cela la *théorie* relative à une contrainte q, qui contient l'ensemble des motifs du langage des attributs qui satisfont q dans une base de données r. Cet ensemble est noté $Th(\mathcal{L}_{\mathcal{A}}, q, r)$. Dans la littérature, le paramètre intéressant développé pour cette complexité est le nombre d'accès à la base de données. En effet, il est notoire que les accès dans les fichiers constituent généralement le facteur limitant, par rapport à un accès à la mémoire centrale.

C'est cette difficulté qui a motivé la technique d'extraction, dont le domaine d'application initial était les bases de transactions collectées lors de chaque achat de millions de consommateurs. Il était alors impératif de trouver des algorithmes dont la complexité soit faible relativement à ce paramètre. La taille de l'espace de recherche est quant à elle exponentielle en nombre d'attributs et pour ce type de problème commercial, ce nombre d'attributs peut être très élevé, de l'ordre de quelques milliers. Néanmoins les transactions sont habituellement courtes, de l'ordre d'une centaine d'attributs car les clients achètent peu de produits au regard de ceux disponibles. L'élagage étant très puissant, la taille en nombre d'attributs devient donc secondaire dans ces données peu denses, sauf si ce sont fréquemment les mêmes attributs qui reviennent.

Pour déterminer la théorie $Th(\mathcal{L}_{\mathcal{A}}, q, r)$, MANNILA et TOIVONEN ont montré [Mannila et Toivonen, 1997] que l'on testera lors de l'algorithme par niveaux (détaillé section 1.3.1) au moins autant de fois la contrainte q sur r qu'il y a de motifs dans les bordures positive et négative de $Th(\mathcal{L}_{\mathcal{A}}, q, r)$. Même si ce résultat ne fournit pas de précision sur la complexité en fonction des paramètres de taille (nombre m d'attributs et n d'objets), il précise que l'on ne fera pas l'économie de tester les motifs de la bordure négative : ces motifs délimitent la frontière des recherches.

Pour mieux cerner la complexité de l'algorithme d'extraction des motifs fréquents, nous indiquons le lien entre cette difficulté et la caractéristique des problèmes NP-difficiles. Pour cela, [Gunopulos *et al.*, 1997b, Purdom *et al.*, 2004] ont montré que prouver l'existence d'un motif fréquent d'une longueur précise dans une base de données est un problème NP-complet car il se réduit à un problème de clique bipartite équilibrée. Le problème du comptage du nombre de motifs fréquents est de fait #P-difficile (réduction au calcul du nombre d'affectations satisfaisantes d'une formule 2CNF) [Gunopulos *et al.*, 1997b]. Cette difficulté souligne le fait qu'il est illusoire

de vouloir produire les motifs fréquents d'une façon déterministe : il faudra parcourir le treillis méthodiquement et avec retour en arrière, par exemple depuis l'ensemble vide si l'on ne dispose pas de plus amples informations.

Il reste néanmoins que le nombre de motifs fréquents est limité dans les bases de données aléatoires [Agrawal *et al.*, 1996]. On dispose même de bornes serrées sur le nombre de candidats [Geerts *et al.*, 2001]. Dans la pratique, la longueur d'un motif fréquent maximum excède rarement la vingtaine d'attributs, même pour des seuils très faibles. Si l'on considère la variante de `Guess & Correct` qui découpe le treillis en niveaux (détaillée section 1.3.1), l'algorithme effectuera un nombre de passes sur la base égal à la longueur maximale des motifs. Chacune de ces passes examine chaque objet un par un et y vérifie la présence de tous les motifs candidats. L'opération est donc linéaire en taille de la base et cet argument est souvent mis en avant dans les publications.

La vision classique de la complexité des tâches d'extraction est donc liée aux nombre d'accès à la base de données et on connaît la taille de l'espace de recherche dans le pire des cas. Dans le chapitre 4, nous livrons un point de vue original sur cette difficulté puisque nous réalisons une analyse en moyenne du nombre de motifs fréquent, en fonction de la taille de la base. Nous nous focalisons donc sur la taille de l'ensemble solution cherché, plutôt que sur celle de l'espace de recherche.

1.3 Typologie des algorithmes

`Guess & Correct` fournit une méthode générique de recherche des motifs vérifiant une contrainte antimonotone, mais la complexité du calcul de la bordure négative utilisé pour générer les candidats reste importante. En revanche, il est possible d'aménager cette étape de génération des candidats et l'*algorithme par niveaux* exploite cette possibilité en contraignant la recherche sur les tranches horizontales successives du treillis, constituées par les motifs de même longueur. *A contrario*, l'*algorithme en profondeur* spécialise autant que possible les candidats et travaille avec des générateurs de longueurs hétérogènes. Nous avons choisi cette typologie car nous avons remarqué que la grande majorité des algorithmes connus utilise l'une ou l'autre approche. La différence entre *par niveaux* et *en profondeur* n'est pas ici relative à la façon de parcourir l'espace de recherche (largeur ou profondeur d'abord) mais révélatrice de la procédure employée pour générer les candidats.

1.3.1 L'algorithme par niveaux

C'est la plus simple application de `Guess & Correct` car les motifs générateurs de candidats ont tous la même longueur. À partir des singletons fréquents, on produit les paires candidates, dont il faudra vérifier la fréquence sur la base. À l'aide des paires fréquentes, les triplets sont générés, puis testés dans la base et le procédé est itéré. La génération des candidats se fait donc

à partir de motifs fréquents d'égale longueur, par simple fusion à préfixe commun (voir figure 1.2), sans nécessiter le recours à des techniques coûteuses comme les traverses minimales.

L'algorithme par niveaux fournit une solution à l'étape *guess* de `Guess & Correct` (voir algorithme 1) et ne modifie pas l'étape *correct*. Cependant, on calculera la bordure négative de $S \backslash \mathcal{E}$, plutôt que celle de S dans son intégralité ($\mathcal{E}$ est l'ensemble des motifs qui ont été examinés).

La littérature prodigue de nombreuses instances de l'algorithme par niveaux, dont Apriori est le pionnier [Agrawal *et al.*, 1993]. Pour optimiser les performances, les astuces sont nombreuses. La vérification de la contrainte utilisera par exemple un parcours vertical de la base, considérée comme un ensemble d'objets, ou horizontal, comme un ensemble d'attributs, voire une approche hybride [Hipp *et al.*, 2000b]. Quand les dimensions le requièrent, on peut échantillonner [Toivonen, 1996] pour fournir un ensemble de candidats potentiels, diviser la base en groupes d'objets ou d'attributs [Ahmed *et al.*, 2003]. Toutes ces méthodes partagent l'utilisation de générateurs de même longueur, seules diffèrent les approches de vérification de contrainte pour un candidat.

Nous donnons tableau 1.3 pour l'exemple du tableau 1.2 la suite des motifs candidats $\mathcal{C}and_i$ et des motifs fréquents $\mathcal{S}_i$ de longueur i avec leur fréquence, pour un seuil de deux objets. Les motifs de la bordure positive sont soulignés.

$\mathcal{C}and_1$	$\mathcal{S}_1$	$\mathcal{C}and_2$	$\mathcal{S}_2$	$\mathcal{C}and_3$	$\mathcal{S}_3$
a_1	a_1 (4)	toutes	a_1a_3 (2)	$a_1a_3a_5$	$\underline{a_1a_3a_5}$ (2)
a_2	a_2 (4)	les	$\underline{a_1a_4}$ (2)	$a_2a_3a_6$	$\underline{a_2a_3a_6}$ (2)
a_3	a_3 (5)	paires	a_1a_5 (2)		
a_4	a_4 (3)	de	a_2a_3 (3)		
a_5	a_5 (3)	a_1a_2	a_2a_6 (2)		
a_6	a_6 (3)	à	a_3a_5 (3)		
a_7	a_7 (2)	a_6a_7	a_3a_6 (2)		
			$\underline{a_4a_7}$ (2)		

Tab. 1.3 – Exécution de l'algorithme par niveaux.

Dans la pratique, sur toutes les paires candidates, peu sont fréquentes. Lorsque les attributs booléens dérivent d'attributs multi-valués ou continus, ce phénomène est amplifié par la binarisation des données, qui rend incompatibles certaines paires. Dans notre exemple, a_1a_2 et a_3a_4 proviennent des mêmes attributs multi-valués X_1 et X_2, ils ne peuvent donc être fréquents. Ces candidats sont malgré tout générés et seul un examen de la base permet de les éliminer si on ne dispose pas de connaissances préliminaires sur le domaine.

On constate en revanche que la génération des candidats de longueur 3 est très efficace car

tous les motifs candidats à cette étape sont fréquents. En effet, $a_1a_3a_4$ n'est pas généré à partir de a_1a_3 et a_1a_4 car son sous-ensemble a_3a_4 n'est pas fréquent : le critère 2 s'applique et $a_1a_3a_4$ ne peut être fréquent. En pratique, le second critère d'élagage est très efficace car il limite fortement l'espace de recherche, même s'il est coûteux à vérifier, tant en temps de calcul qu'en espace mémoire pour stocker les éléments temporaires.

1.3.2 Les algorithmes en profondeur

Principe

L'exemple typique d'algorithme en profondeur est fourni par GUNOPULOS *et al.* dans [Gunopulos *et al.*, 1997b], qui utilise pleinement le concept de bordure négative, sous le nom d'éléments orthogonaux minimaux. Il s'agit d'un algorithme randomisé, dans la mesure où chaque motif fréquent découvert est aléatoirement spécialisé par extension progressive, attribut par attribut, tant que la contrainte est vérifiée. Le motif ainsi obtenu est maximal car la contrainte est antimonotone et le critère 1 s'applique. Il est ajouté à l'ensemble temporaire des maximaux.

Décrivons le fonctionnement de cet algorithme sur le jeu de données de la table 1.2, en considérant un seuil de fréquence minimum toujours égal à 2 objets. Le tableau 1.4 donne les versions successives des bordures découvertes par l'algorithme. Ne disposant au départ d'aucune information, $\mathcal{B}d^+$ ou l'ensemble des maximaux, contient l'ensemble vide. La bordure négative correspondante est l'ensemble des singletons (cf. section 1.2.5). Ayant vérifié que l'un des motifs de cette bordure est bien fréquent (ici, le motif a_1), on lui ajoute d'autres attributs choisis aléatoirement, tant que cette spécialisation reste fréquente. Ce faisant, on obtient directement un élément de la bordure positive, ici a_1, a_3, a_5. La bordure négative qui suit en tient compte, car ni a_1 ni a_3 ni a_5 n'y figurent. Le nouveau motif fréquent aléatoirement spécialisé est a_2, qui fournit $a_2a_3a_6$. Le procédé est itéré jusqu'à ce qu'aucun candidat de la bordure négative ne soit fréquent.

Complexité

Le calcul de la bordure négative, effectué à l'aide des traverses minimales de l'hypergraphe des complémentaires, fournit un remarquable résultat. Les deux critères d'élagage de l'espace de recherche (voir section 1.2.2), mis en évidence par l'antimonotonie de la contrainte, sont pleinement exploités même sur des motifs de longueur hétérogène. Cette solution élégante à un problème difficile a cependant un coût algorithmique élevé. La spécialisation aléatoire fournit de façon appropriée les motifs maximaux, mais dans la pratique le calcul de la bordure est très pénalisant car quasi polynomial. Pour $m = |\mathcal{A}|$ attributs et $b = |\mathcal{B}d^+| + |\mathcal{B}d^-|$, ce coût est en $mb^{o(log^2 b)}$ [Fredman et Kachiyan, 1996]. *A contrario*, lors d'une génération de candidats d'égale longueur (algorithme par niveaux), on restera sur un coût polynomial en m.

Gunopulos *et al.* [Gunopulos *et al.*, 1997a] ont fourni une borne supérieure sur le nombre d'accès à la base pour vérifier la contrainte pendant l'algorithme en profondeur. Cette borne a été récemment améliorée par Satoh et Uno [Uno et Satoh, 2003] : $|\mathcal{B}d^-| + m \cdot |\mathcal{B}d^+|$ au lieu de $|\mathcal{B}d^-| \cdot |\mathcal{B}d^+| + m^2 \cdot |\mathcal{B}d^+|$. La différence d'ordre de grandeur tient au fait que l'algorithme introduit par Gunopulos recalcule la bordure négative à chaque motif maximal découvert, sans réutiliser le fait que les motifs précédemment découverts sont toujours valides. La méthode de Satoh et Uno calcule quant à elle de manière incrémentale les traverses infréquentes, ce qui est bien le but final du calcul de bordure négative. Les traverses fréquentes trouvées pendant ce calcul sont spécialisées et leur complémentaire est injecté dans l'hypergraphe de départ. La bordure positive est ainsi construite motif par motif à l'intérieur d'un seul calcul incrémental de traverses minimales. De nombreuses itérations sont économisées et le stockage temporaire de la bordure négative est évité. Cette approche est très différente de celle que nous décrivons, car elle intègre l'utilisation de contraintes à un calcul incrémental de traverses minimales.

L'économie réalisée peut certes être nuancée par la complexité théorique plus importante du calcul de traverses utilisé, exponentielle en m contre usuellement quasi polynomiale en taille des bordures [Fredman et Kachiyan, 1996]. Satoh et Uno indiquent que les instances de problèmes rencontrées en fouille de données se satisfont de ces stratégies.

$\mathcal{B}d_1^+$	$\mathcal{B}d_1^-$	$\mathcal{B}d_2^+$	$\mathcal{B}d_2^-$	$\mathcal{B}d_3^+$	$\mathcal{B}d_3^-$	$\mathcal{B}d_4^+$	$\mathcal{B}d_4^-$	$\mathcal{B}d_5^+$	$\mathcal{B}d_5^-$
$\emptyset$	a_1	$\underline{a_1a_3a_5}$	a_2	$\underline{a_1a_3a_5}$	a_1a_2	$\underline{a_1a_3a_5}$	a_1a_2	$\underline{a_1a_3a_5}$	a_1a_2
	a_2		a_4	$\underline{a_2a_3a_6}$	a_1a_6	$\underline{a_2a_3a_6}$	a_1a_4	$\underline{a_1a_4}$	a_1a_6
	a_3		a_6		a_2a_5	$\underline{a_4a_7}$	a_1a_6	$\underline{a_2a_3a_6}$	a_1a_7
	a_4		a_7		a_4		a_1a_7	$\underline{a_4a_7}$	a_2a_4
	a_5				a_5a_6		a_2a_4		a_2a_5
	a_6				a_7		a_2a_5		a_2a_7
$\overline{\mathcal{B}d_1^+}$		$\overline{\mathcal{B}d_2^+}$		$\overline{\mathcal{B}d_3^+}$		$\overline{\mathcal{B}d_4^+}$	a_2a_7	$\overline{\mathcal{B}d_5^+}$	a_3a_4
a_1a_2,		$a_2a_4a_6a_7$		$a_2a_4a_6a_7$		$a_2a_4a_6a_7$	a_3a_4	$a_2a_4a_6a_7$	a_3a_7
a_3a_4,				$a_1a_4a_5a_7$		$a_1a_4a_5a_7$	a_3a_7	$a_2a_3a_5a_6a_7$	a_4a_5
$a_5a_6a_7$						$a_1a_2a_3a_5a_6$	a_4a_5	$a_1a_4a_5a_7$	a_4a_6
							a_4a_6	$a_1a_2a_3a_5a_6$	a_5a_6
							a_5a_6		a_5a_7
							a_5a_7		a_6a_7
							a_6a_7		

Tab. 1.4 – Exécution de l'algorithme en profondeur.

[Petit *et al.*, 2004] propose une version différente de la construction de $\mathcal{B}d^-$: c'est la bordure négative qui est étendue par niveaux, à partir de l'alphabet complet, en choisissant une relation

de généralisation plutôt que de spécialisation. Un dernier calcul de traverses permet d'obtenir la bordure positive souhaitée.

1.3.3 L'algorithme `Divide & Conquer`

Le dernier type d'algorithme étudié est fondé sur la notion de base conditionnelle [Han *et al.*, 2000]. Il ne peut pas être modélisé par l'algorithme `Guess & Correct` car il ne repose ni sur le principe de la génération des candidats ni sur celui des bordures.

Une base conditionnelle relativement à un préfixe donné X est une projection qui contient tous les objets qui supportent X. Ainsi, la découverte des motifs fréquents de préfixe X est effectuée dans la base conditionnelle de préfixe X. On produira des candidats basés sur X auxquels on ajoute l'un des singletons a_i fréquents de la base conditionnelle. Il reste à appeler récursivement l'algorithme sur la base conditionnelle de $X \cup \{a_i\}$. Bien sûr, la base étudiée diminue à chaque étape, ce qui fait le succès de cette approche.

`Divide & Conquer` est de nature récursive : le problème initial est morcelé (divide) puis l'algorithme est appelé sur le sous-problème (conquer). Il n'y a pas de phase de génération des candidats comparable à celle de `Guess & Correct`, car l'algorithme restreint de lui-même son espace de recherche par l'utilisation de bases conditionnelles. De fait, il est inutile de conserver en mémoire une trace des résultats obtenus et l'algorithme est peu consommateur de cette ressource. De plus, ordonner les attributs suivant leur fréquence permet d'améliorer considérablement les performances puisque chaque étape peut réutiliser les informations fournies par les étapes précédentes [Han *et al.*, 2000]. Les auteurs introduisent enfin une structure d'arbre préfixe de représentation des bases conditionnelles dont l'algorithme tire pleinement profit.

La technique de projection en sous-bases évite les générations de candidats et est utilisée dans d'autres contributions : H-MINE [Pei *et al.*, 2001b] des mêmes auteurs, qui exploite une structure compilée de la base pour répondre efficacement aux nombreux tests de contrainte ; le lecteur peut également se référer à [Agarwal *et al.*, 2001] et [Bykowski et Rigotti, 2003]. Nous citerons enfin [Wang *et al.*, 2003] qui extrait des motifs fermés sur ce principe, et [Pan *et al.*, 2003] qui réalise l'extraction des motifs fermés dans des données biologiques très larges.

1.4 Extraction de motifs sous contrainte

Nous nous sommes jusqu'à maintenant concentré sur l'extraction de motifs sous contrainte antimonotone en illustrant les méthodes à l'aide de la contrainte de fréquence. Dans cette section, nous élargissons notre point de vue à d'autres types de contraintes et nous dressons une rapide synthèse historique de l'extraction de motifs sous contrainte.

Les premiers travaux du domaine [Ng *et al.*, 1998] utilisent les propriétés de monotonie d'une grande variété de contraintes d'agrégat (somme, min, max, count, moyenne) et définissent une

contrainte succincte, composée à la fois d'une contrainte antimonotone et d'une contrainte monotone [Jeudy, 2002]. Lors de l'extraction sous contrainte, une question cruciale concerne la possibilité de réaliser des élagages à l'aide de la contrainte, on dit « pousser » la contrainte à l'intérieur de l'algorithme, afin d'obtenir des élagages plus puissants [Boulicaut et Jeudy, 2000]. Lorsque la contrainte est antimonotone, cette opération est triviale car les algorithmes connus exploitent cette propriété. Pour les contraintes ne disposant pas de ces propriétés, un post-traitement qui sélectionne les motifs valides est une solution plus simple que la création d'un algorithme dédié. À noter qu'un critère de convertibilité est introduit dans [Pei *et al.*, 2001a] : selon un ordre judicieux des attributs, certaines contraintes retrouveront des propriétés de monotonie. Cependant, les compositions de contraintes convertibles ne sont pas nécessairement convertibles.

Ces approches se limitent donc aux propriétés de monotonie et le cadre des bases de données inductives [Imielinski et Mannila, 1996, Boulicaut *et al.*, 1999] formalise la décomposition d'une contrainte complexe en plusieurs contraintes élémentaires qui ont de bonnes propriétés [De Raedt *et al.*, 2002, Lee et De Raedt, 2003, Giacometti *et al.*, 2002]. On retrouve ici l'analogie avec les notions d'espace des versions de Mitchell [Mitchell, 1980, Sebag, 1994], qui caractérise un ensemble de motifs suivant deux contraintes. [De Raedt et Kramer, 2001] propose un algorithme qui extrait cet espace par niveaux. Une autre méthode permettant de travailler simultanément suivant une contrainte antimonotone et une contrainte monotone est décrite dans [Bucila *et al.*, 2002].

Les approches récentes introduites dans [Bonchi *et al.*, 2003b, Bonchi *et al.*, 2003a] réalisent simultanément un élagage de l'espace de recherche et de la base de données pour effectuer la recherche sous contrainte. Enfin, Arnaud Soulet (GREYC) a développé un cadre théorique général qui utilise la conservation de contraintes selon des intervalles définis par les opérateurs de fermeture [Soulet et Crémilleux, 2005]. Il a pour cela défini des opérateurs de manipulation des intervalles et de réécriture des contraintes. L'idée clé est ici de pouvoir exprimer et d'exploirer de façon simple de très nombreuses contraintes avec un même formalisme.

1.5 Conclusion

Les algorithmes d'extraction des motifs fréquents ont été initialement motivés par l'analyse de données marketing. Celles-ci comportent des millions de clients comme objets, des milliers de référence pour attributs, mais ces données sont peu denses et peu corrélées. Cette particularité explique le succès des approches à la `A-priori` (algorithme par niveaux) :

- la complexité est fonction du nombre d'objets de la base, où le nombre d'accès aux données est très pénalisant. `A-priori` effectue un nombre restreint de passes sur la base entière ;
- la génération des candidats de longueur égale est simple. Des structures de données fondées sur des arbres de préfixe, qui stockent les motifs, résolvent aisément ce problème.

L'intérêt des algorithmes en profondeur est moins évident quand on cherche à reproduire ces facteurs de succès. Les accès à la base sont généralement effectués pour chaque candidat testé, ce qui n'est pas le cas d'`A-priori`, qui en une seule passe sur la base calcule efficacement la fréquence de tous les candidats d'une longueur donnée. D'autre part, la génération des candidats est plus complexe quand on ne s'intéresse directement qu'aux maximaux. Mais l'intérêt de cette approche réside dans la condensation du résultat qu'elle propose : les maximaux fréquents sont calculés directement. À partir de ceux-ci, on peut déduire tous les motifs fréquents[2]. Les techniques pour la recherche des motifs fréquents maximaux s'inscrivent dans un cadre formel, avec l'utilisation de problèmes classiques en théorie de la complexité. Un intérêt supplémentaire des approches en profondeur réside dans le fait qu'elles construisent la solution de manière incrémentale ; il n'est pas utile d'attendre les dernières opérations de l'algorithme pour connaître les motifs les plus longs. Un contexte *any time* en bénéficiera.

Cependant, si l'on considère les bornes fournies par Mannila et Toivonen sur le nombre d'accès à la base de données pour tester *un* candidat, celui-ci n'est pas comparable avec le nombre d'accès à la base nécessaire pour tester *plusieurs* candidats. En effet, l'opération la plus coûteuse matériellement consiste à lire un objet dans un fichier. Une fois stocké en mémoire, c'est moins coûteux en temps de tester plusieurs candidats également disponibles en mémoire relativement à cet objet. Ainsi, dans l'approche d'extraction en profondeur, chaque spécialisation progressive d'un candidat nécessite un accès à la base. Par niveaux, les phases d'accès à la base sont factorisées pour vérifier en une passe tous les candidats de la même taille. Dans les grandes bases de données de millions de transactions marketing, cet argument compromet les stratégies en profondeur, d'autant plus que le calcul de traverses minimales est difficile.

Les besoins actuels en fouille de données se rencontrent pour des données de tout type : des données marketing bien sûr, mais également des données médicales, où les objets résument les caractéristiques des patients. Les dimensions de ces bases sont plus raisonnables, de l'ordre du millier d'objets et de la centaine d'attributs, mais les données sont beaucoup plus denses et corrélées. Le nombre d'accès à la base n'est plus un facteur déterminant car celle-ci tient en mémoire. Les problèmes algorithmiques changent et les méthodes doivent être adaptées à ces contextes.

Il existe même des données aux dimensions *pathologiques*, quand la tendance nombre d'objets/nombre d'attributs s'inverse. C'est le cas de données biologiques issues de l'analyse du génome, où les objets codent les résultats de coûteuses expériences (donc peu nombreuses, quelques dizaines) sur des attributs qui représentent des milliers de gènes. Dans cette configuration, les approches à la `A-priori` échouent. D'autres solutions sont à mettre en œuvre. Elles sont détaillées dans la partie III consacrée à la transposition de bases de données.

[2] On ne connaît cependant pas leur fréquence exacte.

Les performances réelles des algorithmes sont finalement difficiles à évaluer, tant la diversité des implémentations est grande. Les possibilités d'optimisation sont nombreuses et la littérature est très riche [Hipp *et al.*, 2000a]. Mais la comparaison des performances pratiques est quasi impossible car les auteurs n'utilisent pas systématiquement les mêmes bases de test ni les mêmes implémentations des algorithmes de référence. Cette situation a justifié la création d'un atelier (le FIMI[3]) associé à la conférence ICDM, dédié à une synthèse fine des méthodes de recherche des motifs fréquents.

Les récents progrès sur l'extraction des motifs fréquents se situent autour de la notion de représentation condensée. En effet, certains motifs possèdent une propriété remarquable : à partir de ceux-ci, il est possible de regénérer la collection complète des motifs fréquents. Ces motifs constituent une représentation condensée des motifs fréquents dont le prochain chapitre fait l'objet.

[3]Frequent Itemset Mining Implementations, http://fimi.cs.helsinki.fi/. Sur la page de Bart Goethals (http://www.cs.helsinki.fi/u/goethals/), on peut trouver de nombreuses implémentations des algorithmes classiques.

Chapitre 2

Représentations condensées de motifs fréquents

Le nombre de motifs présents dans une base de données est généralement très grand, il est courant d'en obtenir plusieurs millions. Leur utilisation repose classiquement sur une démarche de sélection par la fréquence afin de ne conserver que les plus représentatifs, mais aussi parce que c'est une façon pragmatique efficace d'élaguer l'espace de recherche. Néanmoins cette approche s'avère insuffisante dans de nombreux cas pratiques et montre ses limites lorsque l'espace de recherche est important ou la base très dense et corrélée.

Les représentations condensées de motifs [Mannila et Toivonen, 1996] fournissent une solution au problème de l'extraction de motifs fréquents en proposant un résumé des motifs fréquents, tout en assurant de pouvoir reconstruire l'ensemble des motifs fréquents si nécessaire. Dans certains cas, une approximation sur la fréquence est autorisée. Le terme de représentation condensée (ou ϵ-adéquate) est formellement défini dans [Mannila et Toivonen, 1996]. De façon intuitive, disons qu'une représentation condensée utilise des motifs particuliers de la structure qu'elle synthétise pour savoir si un motif est fréquent.

Ces représentations offrent plusieurs avantages par rapport aux méthodes traditionnelles :

1. la concision de l'ensemble des motifs obtenus, tout en permettant la regénération de chaque motif fréquent ;
2. l'efficacité des algorithmes d'extraction qui fournissent les représentations ;
3. les nouveaux usages possibles des motifs dans le cadre de méthodes d'exploration (cf. chapitre 10).

Nous commençons ce chapitre (section 2.1) en donnant des exemples intuitifs de représentations condensées. Puis la représentation condensée à base de motifs fermés est détaillée section 2.2), ainsi que la connexion de Galois. Nous terminons cette présentation à la section 2.3 avec l'exemple des motifs libres et la section 2.4 pour les motifs δ-libres.

2.1 Exemples de représentations condensées

Pour mieux comprendre la notion représentation condensée, donnons un exemple historique et rappelons nous des tables de trigonométrie ou de logarithmes. Elles fournissent les valeurs de ces fonctions pour des valeurs particulières. Sur l'extrait de la table des *cosinus* (cf. table 2.1), on peut par exemple calculer une valeur approchée de $\cos 0,153$, qui se situe entre celle de $\cos 0,15$ et $\cos 0,16$, soit entre $0,987$ et $0,989$ (une valeur plus précise fournie par une calculatrice indique $0,98831831$). Ainsi, pour obtenir une valeur approchée à la question posée, il suffit de connaître les réponses pour certaines données particulières.

angle	cosinus
...	...
0,14	0,990
0,15	0,989
0,16	0,987
...	...

TAB. 2.1 – Représentation condensée pour le calcul des cosinus.

Revenons à la fréquence d'un motif. Les motifs fréquents et leur fréquence forment une représentation condensée de l'ensemble de *tous* les motifs munis de leur fréquence. Soit γ le seuil de fréquence minimal. On utilise l'approximation suivante :

$$\mathcal{F}_{rep}(X) = \begin{cases} 0 & \text{si } X \text{ n'est pas fréquent } (\mathcal{F}(X) \leq \gamma - 1) \\ \text{sa fréquence calculée} & \text{si } X \text{ est fréquent.} \end{cases}$$

Cette représentation est $(\gamma - 1)$-adéquate, car l'erreur sur la valeur recherchée est bornée par $(\gamma - 1)$. Une définition analogue pour la fréquence d'un motif fournit une représentation $(\gamma/2)$-adéquate :

$$\mathcal{F}_{rep}(X) = \begin{cases} \gamma/2 & \text{si } X \text{ n'est pas fréquent } (\mathcal{F}(X) \leq \gamma - 1) \\ \text{sa fréquence calculée} & \text{si } X \text{ est fréquent} \end{cases}$$

Les motifs maximaux fréquents introduits à la section 1.3.2 fournissent un autre exemple simple de représentation condensée des motifs fréquents. L'antimonotonie assure en effet que tout sous-ensemble d'un motif satisfaisant la contrainte la vérifie également ; à partir des motifs fréquents maximaux, on peut dériver tous leurs sous-ensembles, qui sont aussi fréquents. Dans ce cas, l'ensemble des maximaux fréquents est une représentation des motifs pour répondre à la question : le motif est-il fréquent ? Il est en revanche impossible de connaître précisément sa fréquence.

2.2 Motifs fermés

Dans cette section, nous commençons par donner la définition intuitive des motifs fermés, liée à l'extraction des motifs fréquents. Si on examine attentivement le treillis des motifs, on s'aperçoit qu'il est structuré à l'aide de classes d'équivalence. Ces classes regroupent les motifs qui ont le même support, *i.e.* qui sont supportés par les mêmes objets. De fait, ils ont la même fréquence et un préfixe commun. La figure 2.1 illustre ce phénomène sur la base d'exemple de la table 1.2 page 13. Elle représente les motifs munis de leur fréquence, générés par un algorithme qui recherche les motifs 2-fréquents. Il y a 17 motifs, mais seulement 11 classes d'équivalence. Les motifs d'une classe ayant tous la même fréquence que l'unique motif maximal de la classe, on peut se contenter de ce motif pour indiquer la fréquence des autres motifs de la classe. Les 17 motifs fréquents peuvent donc être résumés par les 11 motifs maximaux des classes d'équivalence.

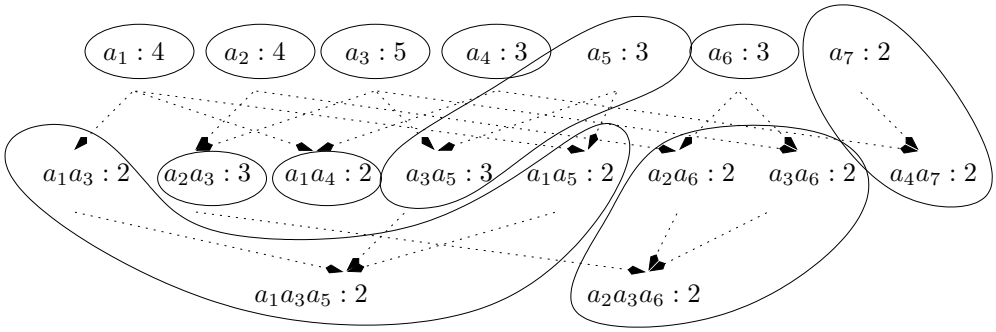

FIG. 2.1 – Classes d'équivalence des supports.

2.2.1 Connexion de GALOIS

Les motifs maximaux des classes d'équivalence sont appelés motifs *fermés*, en référence à l'opérateur de fermeture de GALOIS. En effet, il existe deux opérateurs de *connexion* de GALOIS sur un contexte booléen $r = (\mathcal{A}, \mathcal{O}, R)$:

Définition 10 (Connexion de GALOIS) *On définit les opérateurs suivants :*

$$f(O) = \{a \in \mathcal{A} \mid \forall o \in O, R(a, o) = present\}$$

$$g(A) = \{o \in \mathcal{O} \mid \forall a \in A, R(a, o) = present\}$$

f représente l'ensemble de tous les attributs communs à un groupe d'objets O (on parle d'intension) et g l'ensemble des objets qui possèdent tous les attributs de A (extension). Le couple (f, g) définit la[4] *connexion de* GALOIS *[Birkhoff, 1967] entre $\mathcal{A}$ et $\mathcal{O}$; $h = f \circ g$ et $h' = g \circ f$*

[4] Il serait plus exact de parler d'*une* connexion de GALOIS. Cependant, cette connexion particulière est très classique en fouille de données.

sont les opérateurs de la fermeture de Galois.

Le terme de « connexion » est motivé par le fait que la relation binaire R connecte chaque attribut à chaque objet et vice-versa. Les opérateurs de Galois connectent un motif d'attribut (resp. d'objets) avec un motif d'objet (resp. d'attributs).

Les motifs fermés sont les motifs qui sont invariants par l'opérateur de fermeture ($A = h(A)$). L'association d'un motif fermé d'attributs et de son extension (un motif fermé d'objets) forme un concept :

Définition 11 (Motif fermé, concept) *Un motif A d'attributs est fermé ssi $h(A) = A$. Un motif O d'objets est fermé ssi $h'(O) = O$. Un concept [Wille, 1982] (A, O) associe deux fermés A d'attributs et O d'objets, tels que $A = f(O)$ (ou $O = g(A)$).*

La figure 2.2 représente notre exemple de contexte et souligne le concept $(a_3a_5, o_1o_2o_3)$. En effet, $g(a_3a_5) = o_1o_2o_3$ et $f(o_1o_2o_3) = a_3a_5$.

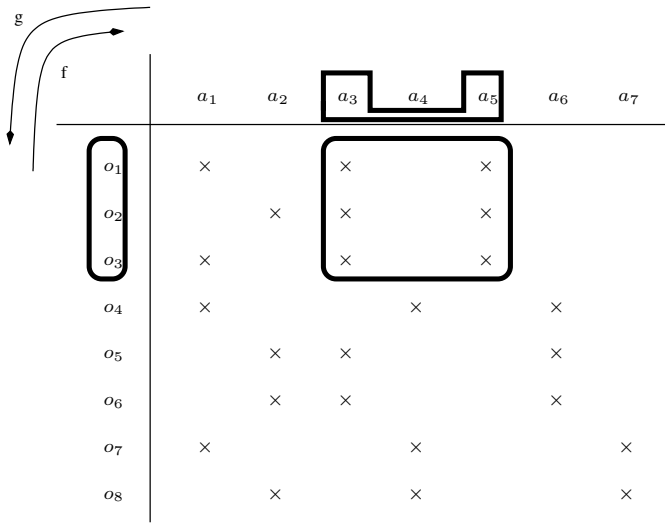

Fig. 2.2 – Connexion de Galois.

Les opérateurs f et g satisfont les propriétés suivantes :

Propriété 2 (opérateurs de Galois)

- *f et g sont décroissantes relativement à l'ordre d'inclusion : si $A \subseteq B$ alors $g(B) \subseteq g(A)$ (resp. $f(B) \subseteq f(A)$) ;*
- *si A est un motif d'attributs et O un motif d'objets, alors $g(A)$ est un motif fermé d'objets et $f(O)$ un motif fermé d'attributs ;*
- *$A \subseteq h(A)$ (extensivité de l'opérateur de fermeture).*

Les opérateurs h et h' sont des opérateurs de fermeture, car ils possèdent les propriétés suivantes [Stadler et Stadler, 2002] (pour tous motifs A et B) :

Propriété 3 (opérateur de fermeture)

- extensivité : $A \subseteq h(A)$;
- idempotence : $h(h(A)) = h(A)$;
- isotonie : $A \subseteq B \Rightarrow h(A) \subseteq h(B)$.

Plusieurs définitions équivalentes coexistent pour la notion de motif fermé : c'est le point fixe de l'opérateur de fermeture, ou bien également le maximum de la classe d'équivalence des supports, ou encore un motif dont toutes les spécialisations sont moins fréquentes que lui. Précisons aussi que le motif fermé qui englobe un motif X est constitué par l'intersection de tous les objets contenant X.

En ce qui concerne l'extraction des motifs fréquents, la propriété essentielle des motifs fermés est qu'un motif quelconque a la même fréquence que le motif fermé de sa classe d'équivalence. Ceci confère aux motifs fermés fréquents la propriété de constituer **une représentation condensée exacte des motifs fréquents**. Le qualificatif *exact* souligne le fait que la valeur de fréquence fournie n'est pas approximée. La fréquence d'un motif quelconque est celle plus petit motif fermé qui le contient. Sur notre exemple, le motif fermé de a_2a_6 est $a_2a_3a_6$, dont la fréquence vaut 2 : la fréquence de a_2a_6 vaut 2 également.

2.2.2 Algorithmes d'extraction de motifs fermés

Les algorithmes qui permettent de calculer les motifs fermés ou les concepts présents dans les bases de données sont nombreux et anciens, ainsi que les études qui les comparent. Les concepts sont utilisés depuis longtemps en apprentissage automatique et il existe de nombreuses approches de classification à partir de ces couples, supervisée [Simon, 2000, Mephu Nguifo et Njiwoua, 2000] comme non supervisée [Durand et Crémilleux, 2002]. La littérature concernant les algorithmes d'extraction de l'intégralité des motifs fermés est donc riche [Ganter, 1984, Guénoche, 1993, Godin *et al.*, 1995, Fu et Mephu Nguifo, 2003]. [Kuznetsov et Obiedkov, 2002] en dresse un état de l'art et détaille les composants des algorithmes historiques, selon qu'ils sont incrémentaux, trient les attributs, utilisent du hachage, du partitionnement, des structures d'arbres, etc. Les complexités sont indiquées et des comparaisons sont effectuées sur des données aléatoires.

L'étude [Fu et Mephu Nguifo, 2004, Fu, 2005] effectue des comparaisons sur les bases de données fournies par l'UCI [Blake et Merz, 1998] et propose sa propre amélioration de l'algorithme de Ganter [Ganter, 1984], le plus rapide lors de ces tests. Fu étudie également l'impact de la transposition de la base de données sur ces algorithmes et nous reviendrons sur cet aspect au chapitre 8.

Cependant, les dimensions actuelles des bases de données n'autorisent pas toujours le calcul de cette intégralité et des approches plus récentes font appel aux élagages selon la fréquence [Pasquier, 2000, Bastide, 2000, Pei *et al.*, 2000, Wang *et al.*, 2003, Zaki et Hsiao, 2002, Stumme *et al.*, 2002, Wang *et al.*, 2003]. Pour les bases de données aux dimensions pathologiques (grand nombre d'attributs devant le nombre d'objets), nous proposons au chapitre 8 une technique d'extraction qui utilise la transposition de base de données.

Pour nos expériences, nous utilisons un extracteur de motifs libres (les motifs libres sont les générateurs minimaux des classes d'équivalence, voir section suivante) qui indique simultanément la fermeture des motifs découverts. Les motifs fermés sont ainsi disponibles. Lors de tests sur des données particulièrement volumineuses et pour extraire *tous* les motifs fermés, nous avons utilisé l'extracteur de Takeagi UNO [Uno *et al.*, 2003] disponible sur `fimi.cs.helsinki.fi/src/`. L'avantage de notre prototype est que nous pouvons intervenir de nombreuses façons sur ses modes de calcul. Entre autres, il permet de calculer l'extension et donc les concepts complets, ce qui n'est pas le cas des algorithmes classiques.

2.3 Motifs libres

Revenons maintenant sur la structure des classes d'équivalence des supports. Si chaque motif maximal est un motif fermé, les motifs minimaux sont appelés motifs *générateurs* ou *clés* [Bastide *et al.*, 2000, Bastide *et al.*, 2002, Stumme *et al.*, 2002] ou motifs *libres* [Boulicaut *et al.*, 2000]. Nous utilisons la dénomination « libres » par souci d'homogénéité avec les travaux de CALDERS et GOETHALS [Calders et Goethals, 2003], exposés au chapitre suivant et largement réutilisés dans ce mémoire. À la figure 2.1, on constate par exemple qu'une classe regroupe les motifs a_7 et a_4a_7. Cela signifie que l'attribut a_4 est toujours présent avec l'attribut a_7. Un algorithme d'extraction de motifs peut exploiter cette propriété pour élaguer l'espace de recherche : il évitera de produire des candidats qui contiennent simultanément ces deux attributs et se limitera aux motifs libres.

La propriété de minimalité des motifs libres dans les classes d'équivalence souligne le fait que ces motifs ne contiennent pas de corrélation intrinsèque. Leur caractérisation est la suivante :

Définition 12 (Motif libre) *Un motif Z est libre s'il n'existe sur Z aucune règle d'association exacte $X \rightarrow Y$, avec $X \subsetneq Z$ et $Y = Z \backslash X$.*

Cette définition signifie que la fréquence d'un motif libre ne peut être déterminée à l'aide de celle de ses sous-ensembles (il n'existe pas de règle pour déduire cette fréquence). Pour connaître la fréquence, il est donc nécessaire d'accéder à la base de données. En revanche, la fréquence d'un motif non libre peut être calculée sans accéder aux données : elle est égale à celle du plus grand motif libre qu'il contient. Lors de la recherche des motifs fréquents, on pourra faire l'économie

d'examiner la base de données pour calculer la fréquence des motifs non libres. À ce titre, l'ensemble des motifs libres et fréquents, muni de leur bordure négative, est une **représentation condensée exacte des motifs fréquents**.

Il faut de plus souligner que la propriété de liberté est antimonotone. Les algorithmes décrits au chapitre précédent sont donc parfaitement réutilisables pour extraire ces motifs, ce qui n'est pas le cas des motifs fermés qui requièrent un algorithme dédié. Les deux critères de génération des candidats sont utilisables tels quels, mais le test de liberté est néanmoins plus complexe à exploiter que celui de fréquence.

Pour mettre en œuvre ces principes, la technique consiste, lors du parcours de la base, à repérer les attributs qui sont toujours présents avec le candidat testé, de manière à calculer sa fermeture. Dans notre exemple, le motif a_2a_3 est libre. D'autre part, l'attribut a_3 est dans chaque objet qui contient le motif a_2a_6 : sa fermeture est $h(a_2a_6) = a_2a_3a_6$. De même, $h(a_3a_6) = a_2a_3a_6$. Lors de la production des candidats, l'algorithme envisagera le cas du motif $a_2a_3a_6$, généré par a_2a_3 et a_2a_6. Cependant, l'attribut a_3 de la fermeture du générateur a_2a_6 figure dans l'autre générateur : le candidat $a_2a_3a_6$ ne sera pas produit et l'espace de recherche élagué. En quelque sorte, cet algorithme a toujours *un coup d'avance* par rapport à `A-priori`, dans la mesure où il profite du parcours de la base de données pour calculer simultanément les fréquences et les fermetures. Les informations obtenues lui permettent des élagages plus puissants de l'espace de recherche que les deux classiques critères de l'antimonotonie. Le chapitre suivant montrera la généralisation de ce principe pour les motifs k-libres.

Pour extraire les motifs libres, leurs fermetures doivent être calculées. [Boulicaut *et al.*, 2003] indique pour cela une méthode efficace (algorithme `matched`), qui utilise les informations présentes dans la base de données. L'algorithme `Titanic` [Stumme *et al.*, 2002] de calcul du treillis de concepts procède d'une façon différente en utilisant la compatibilité de la mesure de fréquence avec l'opérateur de fermeture. Ce principe est mis à profit dans [Hamrouni *et al.*, 2005] pour construire simultanément le treillis des motifs fermés et la base générique de règles.

2.4 Motifs δ-libres

Cette notion est introduite pour compléter la présentation des représentations condensées. La suite de ce mémoire n'y fera pas appel, sauf au chapitre 13.4 sur nos perspectives.

Les motifs δ-libres permettent de relaxer la contrainte de liberté, en considérant non plus des règles exactes pour calculer les fréquences, mais des règles dont le nombre d'exceptions est borné par δ. On définit pour cela les règles δ-fortes :

Définition 13 (Règle d'association δ-forte) *Une règle δ-forte sur $Z = X \cup Y$ est une règle d'association de la forme $X \rightarrow Y$ qui admet au plus δ exceptions [Boulicaut* et al.*, 2000], soit*

$\mathcal{F}(X) - \mathcal{F}(X \cup Y) \leq \delta$.

La confiance d'une telle règle est au moins égale à $1 - (\delta/\mathcal{F}(X))$. Quand δ est nul, la confiance atteint le maximum possible et on retrouve les règles exactes. Les règles fortes présentent un double intérêt : une efficacité d'extraction améliorée et une meilleure capture des phénomènes globaux dans des données réelles. Un motif δ-libre est défini comme suit :

Définition 14 (Motif δ-libre) *Un motif Z est δ-libre s'il n'existe sur Z aucune règle δ-forte $X \rightarrow Y$ (avec $X \subsetneq Z$ et $Y = Z \backslash X$).*

Par exemple, a_2a_3 est 0-libre car aucune règle construite avec des sous-ensembles de a_2a_3 n'est exacte. Autrement dit, ces règles admettent au moins une exception (les seules règles possibles seraient $\emptyset \rightarrow a_2a_3$, $a_2 \rightarrow a_3$ et $a_3 \rightarrow a_2$). Si $\delta = 1$, a_2a_3 n'est pas 1-libre car la règle $a_2 \rightarrow a_3$, admettant une exception, est vérifiée par les données. D'un point de vue technique, les règles δ-fortes peuvent être construites à partir des motifs δ-libres, qui en constituent les parties gauches ou prémisses [Boulicaut *et al.*, 2003].

Comme pour les motifs libres (quand $\delta = 0$), la collection des motifs δ-libres et fréquents est une **représentation condensée des motifs fréquents** [Boulicaut et Bykowski, 2000]. Cependant, lorsque $\delta > 0$, ce n'est pas une représentation exacte, car la fréquence d'un motif quelconque est approchée par celle du plus grand motif δ-libre qu'il contient, avec une erreur bornée. Dans [Boulicaut *et al.*, 2000], il est souligné qu'en pratique l'erreur est faible. Enfin, les motifs δ-libres ont des propriétés remarquables pour construire des règles δ-fortes avec une confiance élevée ou de caractérisation de classe [Crémilleux et Boulicaut, 2002].

2.5 Discussion

Au début de ce chapitre, nous avons précisé que les représentations condensées apportent une vision concise des motifs fréquents et que les algorithmes utilisés sont plus efficaces que ceux qui extraient directement les motifs fréquents. Cependant, ces deux points sont liés à la taille des classes d'équivalence. Il est par exemple envisageable qu'une base de données ne contienne aucune corrélation, auquel cas le pouvoir de concision et d'efficacité de représentations à base de motifs fermés ou libres s'effondre.

D'un point de vue algorithmique, ces techniques auront d'autant plus d'intérêt que les données de la base sont corrélées. Dans le cas contraire, l'effort logiciel nécessaire pour migrer de l'extraction classique des motifs fréquents à l'extraction des motifs fermés ou libres ne se justifie pas. Nous reviendrons sur cet aspect au chapitre 4, où nous étudions l'analyse en moyenne du nombre de motifs fermés dans des bases de données aléatoires.

Chapitre 3

Motifs libres généralisés (motifs k-libres)

La notation k-libre ne doit pas être confondue avec celle de δ-libre, introduite à la section 2.4. δ indique un nombre d'exceptions pour une règle d'association, k qualifie la profondeur d'une règle d'association généralisée. Dans toute la suite de notre travail, seule la notion de motif k-libre est développée.

Nous réservons ce chapitre à l'étude des motifs libres généralisés ou *motifs k-libres*, car ces techniques sont très récentes dans la communauté fouille de données [Calders et Goethals, 2003] et jouent un rôle central dans nos travaux. La partie II de ce document est consacrée aux motifs k-libres en présence de valeurs manquantes ainsi que leur utilisation pour la production de règles d'association généralisées informatives.

Lors de la définition des motifs libres « classiques », nous avons donné l'intuition que leur extraction consiste à prendre *un coup d'avance* en parcourant le treillis : l'examen des bases de données fournit non seulement la fréquence des candidats, mais également leur fermeture. Cette fermeture est utilisée comme une source de corrélations entre attributs et permet de réaliser des économies sur l'espace de recherche. Nous montrons ici la généralisation de ce procédé, qui permet de prendre k coups d'avance pour rendre plus efficace l'extraction des motifs.

Cette présentation détaille à la section 3.1 le calcul de la fréquence d'un motif généralisé à l'aide du principe d'inclusion-exclusion. La section 3.2 développe les règles d'association généralisées, puis la section 3.3 définit les motifs k-libres. Nous proposons les notions de règles d'association généralisées informatives et de fermeture généralisée à la section 3.4. Enfin, la section 3.5 conclut ce chapitre avec la présentation de l'algorithme d'extraction.

3.1 Fréquence d'un motif généralisé

Jusqu'à présent, nous avons travaillé avec des motifs *classiques*, définis comme des sous-ensembles de l'ensemble $\mathcal{A}$ des attributs booléens. Nous étudions désormais des *motifs généralisés*, qui contiennent des attributs booléens, mais également des négations d'attributs booléens. Par exemple, $Z = a_1\overline{a_2}a_3$ est un motif généralisé, que l'on écrira comme l'union d'une partie *positive* $X = a_1a_3$ et d'une partie *négative* $\overline{Y}$ où $Y = a_2$. Un objet o supporte $Z = X \cup \overline{Y}$ si $X \subseteq o$ et $Y \cap o = \emptyset$ (o supporte la partie positive du motif, mais ne contient aucun attribut de la partie négative). Dans la suite de l'exposé, nous omettrons le signe *union* et écrirons $X\overline{Y}$ et $X\overline{a_1}$ à la place de $X \cup \overline{Y}$ et $X \cup \overline{a_1}$.

Donnons rapidement l'intuition qui justifie l'emploi des motifs généralisés. La fréquence de $Z = X\overline{Y}$ est fondamentale : si elle est nulle, cela signifie que l'un des attributs de Y est toujours présent avec X. Un algorithme d'extraction de motifs tirera profit de cette information car elle permet de déduire la fréquence de XY à l'aide de celles de ses sous-ensembles. Dans ce cas, le motif XY ne sera pas conservé par l'algorithme, car sa fréquence n'apporte pas d'information supplémentaire.

Voyons maintenant comment exprimer la relation entre la fréquence de XY et celle de ses sous-ensembles. Le principe d'inclusion-exclusion [Jaroszewicz et Simovici, 2002] est appliqué sur $\mathcal{F}(X, \overline{Y})$. Il permet des égalités de type Bonferroni assimilables à des développements limités.

1. au premier niveau, quand Y contient un attribut : $\mathcal{F}(X\overline{a_1}) = \mathcal{F}(X) - \mathcal{F}(Xa_1)$
2. avec 2 attributs : $\mathcal{F}(X\overline{a_1a_2}) = \mathcal{F}(X) - \mathcal{F}(Xa_1) - \mathcal{F}(Xa_2) + \mathcal{F}(Xa_1a_2)$
3. avec 3 attributs : $\mathcal{F}(X\overline{a_1a_2a_3}) = \mathcal{F}(X) - \mathcal{F}(Xa_1) - \mathcal{F}(Xa_2) - \mathcal{F}(Xa_3) + \mathcal{F}(Xa_1a_2) + \mathcal{F}(Xa_1a_3) + \mathcal{F}(Xa_2a_3) - \mathcal{F}(Xa_1a_2a_3)$
4. ...

Au premier niveau, la fréquence de $X\overline{a_1}$, si elle est nulle, caractérise le fait que a_1 est toujours présent avec X. Au cours du chapitre précédent, cette propriété a été identifiée comme l'appartenance de a_1 à la fermeture de X. La nullité de $\mathcal{F}(X\overline{a_1})$ détermine la possibilité d'effectuer des élagages de l'espace de recherche car la fréquence de Xa_1 peut directement se déduire de celle de son sous-ensemble X.

L'expression la plus générale est la suivante :

$$\begin{aligned}\mathcal{F}(X\overline{Y}) &= \sum_{\emptyset \subseteq J \subseteq Y} (-1)^{|J|}\mathcal{F}(XJ) \\ &= \sum_{\emptyset \subseteq J \subsetneq Y} (-1)^{|J|}\mathcal{F}(XJ) + (-1)^{|Y|}\mathcal{F}(XY)\end{aligned}$$

Conformément à [Calders et Goethals, 2002], la somme $\sum_{\emptyset \subseteq J \subsetneq Y} (-1)^{|J|} \mathcal{F}(XJ)$ sera désormais notée $\sigma(X, Y)$. Nous obtenons donc l'expression finale pour la fréquence de XY en fonction de celle de ses sous-ensembles et de $freq(X\overline{Y})$:

$$\mathcal{F}(XY) = -(-1)^{|Y|}\sigma(X, Y) + (-1)^{|Y|}\mathcal{F}(X\overline{Y})$$

La fréquence de $X\overline{Y}$ étant toujours positive, l'équation précédente permet de déterminer, suivant la parité de la longueur de Y, une borne inférieure ou supérieure de la fréquence de XY :

– si Y est de longueur impaire, c'est une borne supérieure :

$$\mathcal{F}(XY) \leq \sigma(X, Y) \tag{3.1}$$

– si Y est de longueur paire, c'est une borne inférieure :

$$\mathcal{F}(XY) \geq -\sigma(X, Y) \tag{3.2}$$

L'argument développé par Calders et Goethals est le suivant : avant d'examiner les données, calculons à l'aide des équations précédentes les bornes sur cette fréquence, selon toute décomposition de Z en XY. Les meilleurs bornes sont conservées. Lorsqu'elles sont égales, elles indiquent donc la fréquence exacte sans avoir besoin d'examiner la base de données : il existe une *règle de déduction* [Calders et Goethals, 2002] pour la fréquence de Z à partir de celle de ses sous-ensembles. Lorsque les bornes ne sont pas égales, l'examen de la base de données est nécessaire permet d'obtenir la fréquence exacte des motifs. Si elle coïncide avec l'une des bornes précalculées, le motif n'est pas conservé car sa fréquence n'apporte pas de connaissance nouvelle.

D'un point de vue formel, la nullité de $\mathcal{F}(X\overline{Y})$ détermine s'il y a une corrélation entre X et Y. Dans la suite de cette présentation, elle donne lieu à la définition des règles d'association généralisées.

3.2 Règles d'association généralisées

La formalisation de règles d'association généralisées, ou règles *disjonctives*, permet de qualifier les relations entre les motifs, en s'appuyant sur la fréquence de $X\overline{Y}$.

Définition 15 (Règle d'association généralisée) *Étant donnée une profondeur k ($k \in \mathbb{N}^*$), une règle d'association généralisée basée sur $Z = XY$ est une expression de la forme $X \rightarrow \vee Y$ où X et Y sont deux motifs et $|Y| \leq k$. La fréquence de la règle $X \rightarrow \vee Y$ est le nombre d'objets contenant X et au moins un attribut de Y, que nous notons $\mathcal{F}(X \rightarrow \vee Y)$.*

Notons qu'à la différence des règles d'associations « classiques » (cf. définition 5 page 14), la fréquence d'une règle généralisée $X \rightarrow \vee Y$ n'est pas celle de XY. D'autre part, les règles

classiques représentent un cas particulier des règles d'association généralisées. En considérant $k = 1$ et $Y = y_1 \dots y_{|Y|}$, une règle classique $X \rightarrow Y$ donne lieu à $|Y|$ règles de profondeur 1. Il s'agit des règles $X \rightarrow y_1, \dots, X \rightarrow y_{|Y|}$.

La fréquence de $X\overline{Y}$ représente le nombre d'exceptions à une règle généralisée :

Proposition 1

$$\mathcal{F}(X\overline{Y}, r) = \mathcal{F}(X, r) - \mathcal{F}(X \rightarrow \vee Y, r).$$

Preuve : $\mathcal{F}(X \rightarrow \vee Y, r)$ est le nombre d'objets contenant X et au moins un attribut de Y. Si on y ajoute le nombre d'objets contenant X mais aucun attribut de Y ($\mathcal{F}(X\overline{Y}, r)$), on obtient le nombre d'objets contenant X ($\mathcal{F}(X, r)$). □

Cette propriété caractérise l'exactitude d'une règle $X \rightarrow \vee Y$:

Définition 16 (Règle d'association généralisée exacte) *Soit $X \rightarrow \vee Y$ une règle d'association généralisée de fréquence non nulle. Elle est exacte dans une base de données r si tout objet de r contenant la prémisse X contient au moins un attribut de la conclusion Y. Nous noterons*

$$\models_r X \rightarrow \vee Y \iff \mathcal{F}(X\overline{Y}, r) = 0 \text{ et } \mathcal{F}(X) \neq 0. \tag{3.3}$$

Un point crucial pour les associations généralisées est que leur conclusion est une disjonction. Il est possible d'obtenir des expressions de sémantique équivalente en déplaçant des attributs de la conclusion vers la prémisse ou vice-versa, sous réserve de prendre leur négation. $\models_r X \rightarrow a \vee Y$ équivaut donc à $\models_r X\overline{a} \rightarrow \vee Y$. Par leur formalisme, les règles d'associations généralisées offrent des mécanismes simples de transfert d'attribut que n'autorisent pas les règles d'associations classiques. Nous les utiliserons à la section 10.3 pour mettre en évidence des règles de classification positives qui concluent sur une valeur de classe et des règles négatives qui concluent sur la négation d'une valeur de classe.

3.3 Liberté généralisée (k-liberté)

Le chapitre précédent a introduit la liberté d'un motif Z pour qualifier l'absence de règle d'association classique entre ses sous ensembles. Un motif *libre généralisé* (ou k-libre) est défini selon le même principe, relativement à une règle d'association généralisée :

Définition 17 (Motif k-libre) *Z est k-libre dans une base de données r si et seulement si il n'existe aucune règle d'association généralisée basée sur Z exacte dans r, soit :*

$$\forall XY = Z,\ |Y| \leq k \Rightarrow \mathcal{F}(X\overline{Y}) \neq 0$$

En utilisant les bornes définies précédemment (équations 3.1 et 3.2), un motif Z est k-libre si et seulement si sa fréquence est différente des bornes calculées[5] en restreignant la longueur de ses sous-ensembles Y à k. Dans ce cas, les motifs non k-libres voient concorder leur fréquence avec la prédiction obtenue par les équations 3.1 et 3.2 sur les fréquences des sous-ensembles. Ces motifs n'apportent aucune connaissance nouvelle et les éviter permet de réaliser des économies sur l'espace de recherche.

De ce point de vue, les motifs k-libres sont *non dérivables* [Calders et Goethals, 2002] et structurent le treillis lors du calcul des fréquences. Leur fréquence ne dérive pas d'une combinaison de celles de ses sous-ensembles : il est impossible d'économiser un accès à la base de données si l'on souhaite connaître cette fréquence avec précision. À l'image des nombres premiers pour la multiplication, ces motifs k-libres résistent à toute décomposition de leur fréquence. Mais quand on les connaît, on peut regénérer tous les motifs fréquents et leur fréquence, bien qu'ils soient bien moins nombreux : l'ensemble des motifs k-libres fréquents et de leur bordure constitue **une représentation condensée des motifs fréquents** [Calders et Goethals, 2002].

La k-liberté est de plus une propriété antimonotone et permet l'application des algorithmes décrits au chapitre 1. Le gain en efficacité de l'extraction et de la condensation est déjà connu pour $k = 1$ (les motifs libres classiques), mais décuplé dès que $k \geq 2$ (les motifs sont qualifiés de *disjunction free sets* [Bykowski et Rigotti, 2001]). De plus, le nombre de motifs k-libres décroît lorsque k augmente, pour stagner rapidement. À titre indicatif, nous verrons à la section 3.5 que la valeur $k = 5$ suffit pour détecter toutes les corrélations sur MUSHROOM, un jeu de test classique de l'UCI [Blake et Merz, 1998]. Il n'y en a pas de plus profonde.

3.4 Règles d'association généralisées informatives

Nous définissons ici les règles d'association généralisées *informatives*. Elles permettent d'adapter le concept des classiques règles d'association informatives exactes [Bastide *et al.*, 2002] ou non redondantes [Bastide *et al.*, 2000]. Dans le cadre classique, ces règles ont une prémisse qui est un motif 1-libre X et leur conclusion est $h(X)\backslash X$ (h est l'opérateur de fermeture de GALOIS). Elles forment une couverture de l'ensemble des règles d'associations exactes.

Dans le cadre des associations généralisées, nous cherchons des règles non redondantes dont la prémisse est un motif k-libre. Prenons l'exemple de deux règles exactes $X \rightarrow \vee Y$ et $X \rightarrow \vee Y'$, avec $Y \subsetneq Y'$. Celle qui conclut sur Y' n'apporte pas d'information supplémentaire. En effet, la conclusion est une disjonction entre ses attributs, donc la redondance est évitée en choisissant la conclusion minimale. Selon ce principe, nous construisons et définissons la fermeture généralisée d'un motif k-libre de la façon suivante :

[5] C'est cette définition précise qui est utilisée dans [Calders et Goethals, 2003].

Définition 18 (Fermeture généralisée) *Soit X un motif k-libre. La fermeture généralisée $\mathcal{FG}_k(X)$ est l'ensemble des Y minimaux, de longueur inférieure à k, tels que la règle $X \rightarrow \vee Y$ est exacte.*

Reconnaissons que le mot « fermeture » est abusif et peut prêter à confusion avec le terme consacré d'opérateur de fermeture, utilisé dans la théorie des treillis et des hypergraphes (cf. section 2.2.1). Cependant, il ne nous semble pas opportun d'utiliser l'expression « presque-fermeture », réservée au cas des δ-libres. Cette notion étant centrale dans la suite de l'exposé, nous utiliserons ce terme lorsqu'il n'y a pas d'ambiguïté.

Le calcul de la fermeture généralisée est lié à celui des traverses minimales de longueur bornée, car elle rassemble les motifs minimaux qui intersectent avec les objets contenant la prémisse. Si $Tr_k(S)$ est l'ensemble des traverses minimales de S de longueur inférieure à k, nous obtenons la propriété suivante :

Proposition 2 *Soit X un motif k-libre. La fermeture généralisée de X est*

$$\mathcal{FG}_k(X) = Tr_k(\{o \backslash X \mid o \in \mathcal{O} \text{ et } X \subseteq o\})$$

Preuve : La proposition est une reformulation de la définition 18 à l'aide du principe de traverses minimales de longueur bornée. □

Muni de la notion de fermeture, nous pouvons définir les règles d'association généralisées informatives :

Définition 19 (Règle d'association généralisée informative) *Une règle d'association généralisée informative est une règle d'association généralisée $X \rightarrow \vee Y$ où X est un motif k-libre et $Y \in \mathcal{FG}_k(X)$.*

Notons que par construction, ces règles sont exactes car leur conclusion est un élément de la fermeture.

Les règles informatives classiques ont la propriété d'être les règles exactes à prémisse minimale et à conclusion non redondante (*i.e.* maximale). Pour les règles généralisées informatives, la non redondance est assurée par construction de conclusions minimales, ce qui leur confère la meilleure généralité possible. Elles seront utiles au chapitre 7 pour calculer des règles fiables dans les bases de données incomplètes, et au chapitre 10 pour la construction de règles de classification.

L'exemple des données de la table 1.2 page 13 illustre ces notions. Calculons la fermeture généralisée de a_1 pour $k = 2$. Privés de a_1, l'ensemble des objets $o_1o_3o_4o_7$ qui le contiennent fournit l'hypergraphe $\{a_3a_5, a_3a_5, a_4a_6, a_4a_7\}$. Les traverses minimales sont $a_3a_4, a_3a_6a_7, a_4a_5, a_5a_6a_7$. Pour $k = 2$, on ne garde que les traverses de longueur inférieure à 2 : la fermeture généralisée est $\{a_3a_4, a_4a_5\}$. Les règles généralisées informatives de prémisse a_1 sont donc $a_1 \rightarrow a_3 \vee a_4$ et $a_1 \rightarrow a_4 \vee a_5$.

3.5 Algorithme d'extraction des règles généralisées informatives

3.5.1 Algorithme `k-miner`

Nous avons conçu l'algorithme `k-miner`, qui extrait par niveaux des motifs k-libres fréquents. Pour cela, il exploite l'antimonotonie de la propriété de k-liberté [Calders et Goethals, 2003]. Rappelons qu'il s'agit, à partir d'un ensemble de motifs de même longueur, de produire les candidats à l'aide des classiques critères d'élagage 1 et 2 page 17. Pour décider de la k-liberté d'un candidat, des bornes sur sa fréquence sont calculées, en utilisant les fréquences de ses parties. Lorsque les bornes inférieures et supérieures sont égales ou coïncident avec la fréquence réelle, le candidat est refusé. Lorsque les bornes sont différentes et ne coïncident pas avec la mesure effectuée dans la base, le motif est k-libre.

Chaque motif est pourvu d'une fréquence et de deux bornes sur cette fréquence. Nous notons $fkfree(X)$ la contrainte qui caractérise un motif k-libre fréquent. La méthode complète est détaillée avec les algorithmes 2 et 3.

Pour calculer la fermeture généralisée d'un motif k-libre, nous utilisons l'algorithme de calcul des traverses minimales décrit dans [Kavvadias et Stavropoulos, 1999b], car cette proposition construit les traverses minimales attribut par attribut lors d'un parcours en profondeur. L'algorithme est simple à adapter pour obtenir les traverses minimales de longueur bornée. L'annexe 1 page 177 détaille cet aspect.

Données : une base r, une fréquence minimum γ et k une profondeur de règles
Résultat : l'ensemble $\mathcal{S}$ des motifs vérifiant $fkfree$
/*`initialiser` $\mathcal{C}and$ `avec la liste des singletons` */
$\mathcal{S} = \emptyset$;
répéter
  /*`conserver les candidats qui vérifient` $fkfree$ */
  $\mathcal{S} = \mathcal{S} \cup \mathcal{C}and \backslash \{X \in \mathcal{C}and \mid \neg fkfree(X)\}$;
  générer les candidats dans $\mathcal{C}and$;
jusqu'à $\mathcal{C}and = \emptyset$;
retourner $\mathcal{S}$;

Algorithme 2 – `k-miner`- extraction de motifs k-libres.

3.5.2 Expérimentation et discussion

Nous avons réalisé quelques expériences d'extraction des motifs k-libres sur la base de données `Mushroom` [Blake et Merz, 1998] pour présenter des tendances générales sur les variations de k. Cette base est bien connue comme test pour les algorithmes de fouille de données. La table 3.1

Données : un ensemble de motifs k-libres de longueur l
Résultat : l'ensemble $\mathcal{C}and$ des motifs candidats à la vérification de $fkfree$
pour *chaque candidat Z, généré par fusion de deux k-libres ayant un préfixe commun de longueur $l-1$* **faire**
  vérifier que ses sous-ensembles de longueur $l-1$ sont k-libres;
  /*`calculer les bornes de sa fréquence` */
  début
    construire l'arbre contenant les motifs X et leurs fréquences tels que $|Z \backslash X| \leq k$;
    pour chaque X de l'arbre, calculer la somme alternée des fréquences de tous ses sur-ensembles, qui constitue selon les cas une borne inférieure ou supérieure de $\mathcal{F}(Z)$;
    conserver les meilleurs bornes pour $\mathcal{F}(Z)$;
  fin
  en cas d'égalité des bornes, refuser le candidat;
fin

Algorithme 3 – Génération des candidats de longueur $l+1$.

indique, pour k allant de 1 à 5, le nombre de motifs k-libres présents (de fréquence minimale 1) ainsi que le temps de calcul nécessaire. Pour $k = 0$, nous avons représenté le nombre de motifs fréquents. On constate une nette diminution du nombre de motifs découverts, quand on passe de $k = 0$ à $k = 1$, puis $k = 2$. Le temps d'extraction est également fortement réduit.

En revanche, le nombre de motifs découverts commence à stagner quand $k \geq 3$ et ne diminue plus à partir de $k = 5$. De son côté, le temps de calcul augmente. Sur cet exemple, choisir $k = 2$ permet d'obtenir les meilleurs avantages des motifs k-libres. Lors de nos expériences sur d'autres bases de données, nous avons constaté des phénomènes similaires, également rapportés dans [Calders et Goethals, 2002].

k	nombre de k-libres	temps de calcul (s)
0	$2,7 \cdot 10^9$	2483
1	426134	99
2	224154	52
3	214947	61
4	214538	70
5	214530	75
6	214530	80

Tab. 3.1 – Extraction des motifs k-libres dans Mushroom.

3.6 Conclusion

Calders et Goethals indiquent dans leur travail que le premier niveau de développement de la k-liberté (rappelons que pour $k = 1$, cela correspond aux motifs libres classiques) fournit déjà une excellente représentation, tant par la concision du résultat que par sa lisibilité. Dans [Bykowski et Rigotti, 2001], un deuxième niveau introduit les motifs exempts de disjonction (*disjunction-free*) et améliore encore suffisamment l'opération pour rendre inutiles les niveaux suivants si l'on poursuit un but d'efficacité lors de l'extraction ou de condensation.

La technologie des représentations condensées fondées sur les motifs k-libres arrive aujourd'hui à maturité [Calders et Goethals, 2003, Kryszkiewicz, 2004] : il est possible de raffiner le procédé d'extraction jusqu'à différencier les parties de bordure de théorie utiles au calcul. Il semble difficile d'améliorer les performances d'extraction, sous peine de perte de lisibilité des résultats et de difficulté de regénération d'une solution exploitable. C'est pourquoi les communautés se mobilisent désormais pour l'élaboration de solutions qui exploitent ces motifs.

Les techniques à base de motifs k-libres sont très récentes. Leur mise au point est initialement motivée par la réalisation d'algorithmes d'extraction efficaces. Leurs applications restent cependant rares. Aux chapitres 6 et 7, nous étudions l'adaptation de ces principes aux données contenant des valeurs manquantes. Le chapitre 10 explique comment certaines règles généralisées informatives peuvent constituer des règles de classification.

Chapitre 4

Analyse en moyenne du nombre de motifs fréquents ou fermés

Ce travail a été réalisé avec Loïck Lhote *et Arnaud* Soulet *du GREYC [Lhote et al., 2005a, Lhote et al., 2005b]. Loïck* Lhote *effectue des recherches sur la modélisation et l'approximation de sources complexes, Arnaud* Soulet *travaille sur l'extraction de motifs sous contraintes.*

Nous terminons cette présentation sur l'extraction de motifs en réalisant une analyse en moyenne du nombre de motifs fréquents ou fermés. Nous pensons qu'il s'agit d'un point de vue original sur la complexité de l'extraction des motifs, car il s'attache à déterminer une taille moyenne pour la solution du problème, plutôt que d'étudier l'espace de recherche à parcourir dans le pire des cas. Cette étude permet d'apporter un regard nouveau sur la difficulté des tâches d'extraction et motive la mise en œuvre de notre méthode de transposition, exposée au chapitre suivant.

La section 4.1 rappelle les principes de l'analyse en moyenne et nous définissons quelques hypothèses à la section 4.2. Les modélisations de bases de données utilisées dans ce chapitre sont détaillées à la section 4.3. Le nombre moyen de motifs fréquents est étudié à la section 4.4, dans des modèles de Bernoulli et de Markov. La section 4.5 améliore le modèle de Bernoulli en proposant un modèle générique attribut/valeur. Enfin, la section 4.6 analyse le nombre de motifs fermés. Les preuves des théorèmes énoncés sont regroupées à l'annexe 2.

4.1 Introduction à l'analyse en moyenne

L'analyse en moyenne consiste à considérer toutes les instances d'un problème et à réaliser des calculs en moyenne de la complexité des tâches à évaluer. Les instances sont générées à l'aide de modèles probabilistes ou de séries génératrices. Pour l'étude de la complexité, l'analyse en moyenne fournit un point de vue intéressant pour deux raisons :

– chaque base de données est associée à un modèle de probabilité, ce qui permet de prendre en compte la diversité de chaque cas ;
– si le modèle est proche de la réalité, l'analyse décrit un comportement réaliste moyen des paramètres étudiés qui est parfois éloigné de celui fournit par l'analyse d'un cas pathologique ou d'une situation extrême.

La réalité est souvent différente des modèles probabilistes simples. En outre, plus complexe est le modèle, plus complexe est son analyse. Les techniques d'analyse en moyenne sont récentes et plus couramment utilisées pour l'étude de la complexité des algorithmes classiques, comme celui du calcul de `pgcd`. Pour `A-priori`, nous n'avons trouvé qu'une seule étude de ce type [Purdom *et al.*, 2004], qui estime le nombre de candidats. Ce travail confirme des résultats plus combinatoires fournis par une borne inférieure [Geerts *et al.*, 2001]. D'autre part, les auteurs d'`A-priori` ont utilisé un modèle probabiliste de Bernoulli (voir section 4.3.1) et montré qu'il y a peu de motifs longs dans une base de données aléatoire [Agrawal *et al.*, 1996].

4.1.1 Résultats classiques

Rappelons que Gunopulos et al. ont montré que décider s'il existe un motif fréquent avec t attributs est NP-complet [Gunopulos *et al.*, 1997b, Purdom *et al.*, 2004]. Le problème de comptage associé est #P-difficile. L'espace de recherche pour ce problème de motifs est l'ensemble $2^{\mathcal{A}}$, de taille exponentielle en m le nombre d'attributs de $\mathcal{A}$. Dans certains cas pathologiques, comme celui d'une matrice remplie de 1 (voir figure 4.1-a), la solution au problème de recherche des motifs 1-fréquents coïncide avec l'intégralité de l'espace de recherche, car tous les motifs sont 1-fréquents dans une telle base : il y en a donc 2^m. Dans une matrice où seule la diagonale ne contient pas de 1 (figure 4.1-b), il y a $2^m - 1$ motifs fermés. Finalement, dans la matrice de la figure 4.1-c [Boros *et al.*, 2002], il y a k motifs fréquent maximaux (k est tel que $n = k\gamma$), $2^k - 2$ motifs fermés et plus de $2^{k(l-1)}$ motifs fréquents (l est tel que $m = kl$). C'est une situation où le nombre de motifs fréquents est exponentiellement plus grand que celui de motifs fermés, lui même exponentiellement plus grand que le nombre de motifs maximaux.

En marge de ces travaux, [Dexters et Calders, 2004] a étudié le lien entre la longueur maximale d'un motif k-libre et le nombre de motifs k-libres. Les auteurs montrent que si l est cette longueur, il y a nécessairement 2^l motifs k-libres. Cela donne également une indication sur le nombre minimum d'objets nécessaire dans une base de données afin qu'elle contienne un motif k-libre de longueur donnée.

4.1.2 Point de vue

La complexité de l'extraction des motifs fréquents est largement étudiée dans la littérature. Nous proposons ici une analyse en moyenne du nombre de motifs fréquents ou fermés. Pour cela,

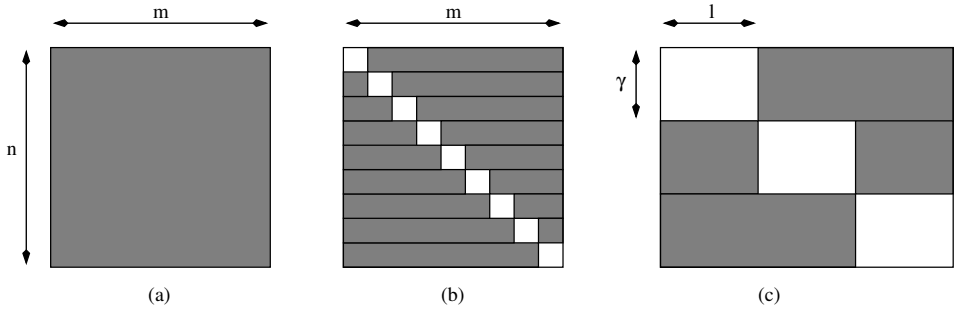

FIG. 4.1 – Trois cas extrêmes pour l'extraction de motifs (les zones remplies symbolisent la présence de 1 dans la matrice).

nous modélisons des bases de données aléatoires à partir de deux modèles :

- un modèle de BERNOULLI, où chaque objet est en relation avec chaque attribut selon une probabilité uniforme. C'est le modèle le plus simple mais aucune corrélation n'est prise en compte, ce qui l'éloigne fortement de la réalité. Cela ne signifie pas que les instances générées par ce modèle sont dépourvues de corrélations, mais ce n'est pas le but du modèle. Malgré tout, il fournit déjà des résultats inattendus qui permettent d'appréhender la complexité de l'extraction des fréquents sous un angle original ;
- un modèle de MARKOV, qui reflète mieux les corrélations présentes dans les données réelles, car chaque objet est produit par une source de MARKOV. Les corrélations locales entre des attributs voisins sont ainsi prises en compte, comme on peut en trouver en bioinformatique entre des gènes voisins, ou dans les séries temporelles. Toutefois, les objets restent indépendants et ces corrélations ne sont pas conservées transversalement, ce qui éloigne également ce modèle de la réalité.

Selon ces modélisations, nous dénombrons le nombre moyen de motifs fréquents ou fermés et calculons une asymptotique pour cette expression, qui permet de l'étudier plus intuitivement. Ces résultats permettent d'appréhender la complexité des algorithmes d'extraction de motifs non pas sous l'angle de taille l'espace de recherche, mais selon la taille de la solution.

4.2 Hypothèses

Rappelons qu'une base de données est un contexte $r = (\mathcal{A}, \mathcal{O}, R)$, et $m = |\mathcal{A}|$, $n = |\mathcal{O}|$. Cette présentation considère plus précisément la version matricielle de r, notée $(\chi_{i,j})_{i=1..n,j=1..m}$. Le terme *motif fermé* sous-entend qu'il est aussi fréquent.

Lors de nos calculs, nous cherchons des asymptotiques lorsque m et n grandissent, pour appréhender la complexité des tâches d'extraction selon un point de vue général. Nous faisons

ici l'hypothèse que m et n restent d'un même ordre de grandeur polynomial :

Hypothèse 1 (Taille des données) *Il existe un constante* $c_0 > 0$ *telle que*

$$\log m \sim c_0 \log n, \quad c_0 > 0.$$

Selon le modèle de calcul employé, nous verrons que nos résultats proviennent d'équivalences asymptotiques réalisables sous certaines hypothèses, comme celle énoncée ci-dessus sur la taille des données. Nous supposons ici que le seuil de fréquence minimum γ sera fixé et petit devant le nombre d'objets, ou qu'il sera proportionnel au nombre d'objets et non négligeable. Cette différence est utile car deux cas particuliers soulignent l'immense contraste entre les deux comportements : les expériences montrent qu'un seuil fixe $\gamma = 1$ fournit un nombre exponentiel de motifs fréquents, tandis qu'un seuil élevé $\gamma = n$ en donne un nombre constant. Nos deux hypothèses permettent de refléter ce contraste.

Hypothèse 2 (Seuil γ fixe) *Le seuil de fréquence minimum* γ *est fixe et ne dépend ni de* n*, ni de* m*. Dans la pratique, on peut considérer un seuil faible , entre* 1 *et* 10 *par exemple.*

Hypothèse 3 (Seuil γ proportionnel) *Il existe* $\alpha \in]0, 1[$ *et le seuil de fréquence minimale est tel que* $\gamma = \alpha \cdot n$*. Dans ce cas, c'est une proportion non négligeable du nombre d'objets.*

Lorsque n est grand, il n'est pas possible de modéliser un seuil de fréquence petit à l'aide d'un seuil proportionnel. Cet argument justifie encore les deux hypothèses. D'un point de vue technique, elles permettent des simplifications différentes dans les calculs et sont exploitées dans les preuves.

La figure 4.2 montre la différence entre les deux hypothèses, lorsque l'accroissement des paramètres de taille augmente la taille du treillis de l'espace de recherche. La partie supérieure blanche représente la solution du problème.

4.3 Modèles de bases de données

Pour notre analyse, nous utilisons deux modèles distincts pour générer les bases de données aléatoire. Il s'agit d'un modèle de Bernoulli, et du modèle de Markov.

4.3.1 Modèle de Bernoulli

Le modèle de base de données de Bernoulli attribue une probabilité uniforme de présence des attributs dans les objets.

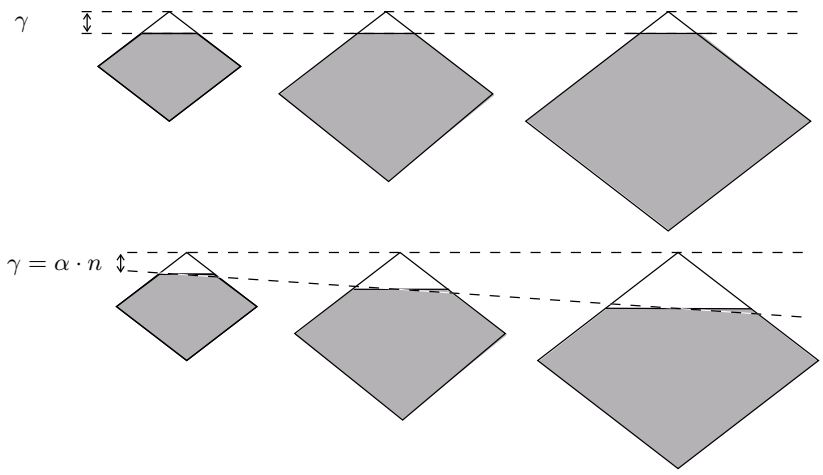

FIG. 4.2 – Différences entre un seuil γ fixe et proportionnel.

Modèle 1 (Modèle simple de BERNOULLI) *La base de données* $(\chi_{i,j})_{i=1..n,j=1..m}$ *forme une famille indépendante de variables aléatoires, qui suivent une loi de* BERNOULLI *de paramètre* p *dans* $]0,1[$.

Ce modèle est éloigné de la réalité, car il ne tient compte d'aucune corrélation. Il est cohérent avec des bases de données qui répertorient des transactions de produits, où chaque unique attribut désigne un produit.

Lorsque les bases étudiées sont au format attribut/valeur, ce sont plusieurs attributs qui s'excluent les uns des autres pour coder une variable. Nous définissons alors un modèle grossier pour ce type de corrélation, où les attributs sont regroupés par bloc de taille t. La figure 4.3 montre un exemple où $t = 3$: il ne peut y avoir qu'un seul 1 par triplet de colonnes. Nous supposerons qu'il existe m_1 tel que le nombre d'attributs satisfait $m = m_1 t$.

Modèle 2 (Modèle groupé de BERNOULLI) *La base de données* $(\chi_{i,j} = (\chi_{i,tj+1}, \chi_{i,tj+2}, \ldots, \chi_{i,tj+t}))_{i=1..n,j=0..m_1-1}$ *forme une famille indépendante de variables aléatoires de même loi uniforme sur l'ensemble des séquences de taille* t *composées d'un seul* 1 *et de* $t-1$ *zéros (la densité est* $\frac{1}{t}$*).*

Ce modèle est moins éloigné de la réalité et il permet pour $t = 2$ de traiter le cas des bases de données où chaque attribut est binaire. Il modélise cependant ces simples corrélations, assimilables à des connaissances du domaine. Nous verrons à la section 4.5 qu'il indique plus précisément le rôle des corrélations pour rapport à un modèle entièrement décorrélé comme la version simple de BERNOULLI.

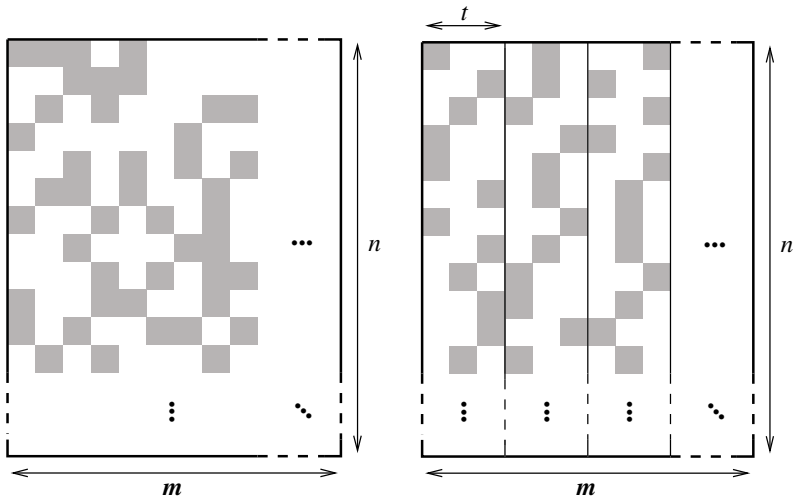

FIG. 4.3 – Modélisation simple et groupée d'une base de données (un carré gris indique un 1 dans la matrice).

4.3.2 Modèle de MARKOV

Contrairement aux modèles précédents, le modèle de MARKOV considère des corrélations locales entre attributs proches : chaque objet est une séquence de m variables aléatoires dans $\{0,1\}$ qui suivent un processus de MARKOV.

Définition 20 (Processus de MARKOV) *Une famille $(\chi_i)_{i=1..m}$ est un processus de MARKOV d'ordre K, si pour tout i, $1 \leq i \leq (m-K)$,*

$$\begin{aligned}&\mathbb{P}\left[\chi_{i+K} = x_{i+K} \mid \chi_{i+K-1} = x_{i+K-1}, \ldots, \chi_1 = x_1\right]\\ &\quad = \mathbb{P}\left[\chi_{i+K} = x_{i+K} \mid \chi_{i+K-1} = x_{i+K-1}, \ldots, \chi_i = x_i\right]\\ &\quad = \mathbb{P}\left[\chi_{K+1} = x_{i+K} \mid \chi_K = x_{i+K-1}, \ldots, \chi_1 = x_i\right]\end{aligned}$$

La valeur d'une variable aléatoire ne dépend que des K variables précédentes et cette dépendance est invariante pour les objets. Un processus de MARKOV d'ordre K est donc complètement décrit par les K premières variables et les probabilités de transition. Nous notons $f_{init} = (f_w)_{w\in\{0,1\}^K}$ pour la distribution initiale, i.e.

$$\mathbb{P}\left[(\chi_1, \ldots, \chi_K) = w\right] = f_w.$$

Le modèle de MARKOV pour les bases de données est donc le suivant :

Modèle 3 (Modèle de MARKOV) *Soit $K \geq 1$, $f_{init} = (f_w)_{w\in\{0,1\}^K}$ une distribution initiale sur $\{0,1\}^K$ et $(p_{w\to x})_{w\in\{0,1\}^K, x\in\{0,1\}}$ les probabilités de transition. Chaque objet est calculé indépendamment des autres de la façon suivante : pour un objet $o = (\chi_1, \ldots, \chi_m)$, $(\chi_1, \ldots, \chi_K)$ est*

calculé selon la distribution initiale f_{init}. Puis, les objets $\chi_{K+1}, \ldots, \chi_m$ sont séquentiellement obtenus à l'aide des K précédents en utilisant les probabilités de transition.

$$p_{w \to x} = \mathbb{P}\left[\chi_{K+1} = x \mid (\chi_1, \ldots, \chi_K) = w\right], \quad x \in \{0,1\}, w \in \{0,1\}^K.$$

Un processus de Markov $((p_{w\to x})_{w\in\{0,1\}^K, x\in\{0,1\}}, f_{init})$ d'ordre K sur $\{0,1\}$ est équivalent à un processus $((p_{w\to v})_{w,v\in\{0,1\}^K}, f_{init})$ d'ordre 1 défini sur $\{0,1\}^K$ dont les probabilités de transition sont

$$p_{(w_1,\ldots,w_K)\to(w_{K+1},\ldots,w_{2K})} = \prod_{i=K+1}^{2K} p_{(w_{i-K},\ldots,w_{i-1})\to w_i}$$

La matrice $P = (p_{w\to v})_{w,v\in\{0,1\}^K}$ est appelée la matrice de transition du processus. Dans les preuves, nous utilisons ce point de vue et considérerons qu'un objet est une séquence de K attributs plutôt qu'une séquence d'attributs. Nous supposerons également que les coefficients de la matrice de transition sont tous positifs, ce qui entraîne que le processus est irréductible et apériodique. En outre, nous supposerons que le nombre d'attributs est un multiple de K, l'ordre du processus.

4.4 Nombre de motifs fréquents

Les résultats énoncés dans cette section montrent des comportements identiques pour les modèles de Bernoulli et Markov pour le nombre moyen de motifs fréquents. La différence concerne leur expression, puisque nous donnons des asymptotiques qui utilisent dans le premier cas des constantes explicites et dans l'autre cas elle contiennent des constantes théoriques reliées à la matrice de transition.

Nous commençons par donner la valeur exacte du nombre moyen de motifs fréquents selon le modèle de Bernoulli.

Théorème 1 *Soit un seuil fixe γ (hypothèse 2) de fréquence minimale. Selon l'hypothèse 1 de taille des bases de données, le nombre moyen de motifs fréquents $F_{m,n,\gamma}$ dans une base de données de modèle simple* Bernoulli *de paramètre p vaut*

$$F_{m,n,\gamma} = \sum_{j=1}^{m} \binom{m}{j} \sum_{i=\gamma}^{n} \binom{n}{i} p^{ij} (1-p^j)^{n-i}.$$

Preuve : Voir annexe 2. □

Le théorème suivant exprime une approximation pour cette quantité :

Théorème 2 *Soit un seuil fixe γ (hypothèse 2) de fréquence minimale. Selon l'hypothèse 1 de taille des bases de données, le nombre moyen de motifs fréquents $F_{m,n,\gamma}$ dans une base de données*

de Bernoulli *et de paramètre p ou de* Markov *de matrice de transition P est polynomial en nombre d'objets et exponentiel en nombre d'attributs,*

$$F_{m,n,\gamma} \sim c_1 \frac{n^\gamma}{\gamma!}\theta^m, \qquad \theta > 1.$$

Dans le modèle de Bernoulli, *$\theta = 1+p^\gamma$ et $c_1 = 1$. Dans le modèle de* Markov, *θ est la valeur propre dominante de P et c_1 est relié à l'objet spectral dominant.*

Preuve : Voir annexe 2. □

Le résultat en moyenne est conforme avec l'intuition fournie par le pire des cas : le nombre de motifs fréquents est exponentiel selon le nombre d'attributs et la réelle difficulté de cette tâche réside ici. Ce phénomène est bien connu et la dépendance polynomiale selon le nombre d'objets est cohérente avec la complexité des algorithmes par niveaux [Mannila et Toivonen, 1997].

Le résultat avec un seuil proportionnel est plus surprenant :

Théorème 3 *Soit $\alpha \in]0,1[$ définissant un seuil proportionnel $\gamma = \lfloor \alpha n \rfloor$ (hypothèse 3) de fréquence minimale. Selon l'hypothèse 1 de taille des bases de données, le nombre moyen de motifs fréquents $F_{m,n,\gamma}$ dans une base de données de* Bernoulli *et de paramètre p, ou de* Markov *de matrice de transition P, est polynomial en nombre d'attributs et ne dépend pas du nombre d'objets,*

$$F_{m,n,\gamma} \leq c_2 m^s.$$

Dans le modèle Bernoulli, *c'est une équivalence et $c_2 = 1/s!$, $s = \lfloor \log \alpha / \log p \rfloor$.*

Preuve : Voir annexe 2. □

Le théorème 3 est assez surprenant, car les expériences mettent en évidence un nombre important de motifs fréquents, même avec des seuils proportionnels. C'est cependant insuffisant pour conclure que la complexité de ce problème est systématiquement exponentielle en nombre d'attributs. Nos résultats montrent que suivant les deux hypothèses de seuil minimal, les comportements sont différents. L'intuition suggère qu'un faible seuil peut être à la fois considéré comme fixe et proportionnel. Malgré cela, ces deux hypothèses occasionnent des comportements théoriques dissemblables. Entre un comportement constant quand le seuil est très élevé et un comportement exponentiel quand il est faible, notre étude montre qu'il y a un intermédiaire polynomial.

Dans le cas d'un seuil proportionnel, l'estimation du nombre moyen de motifs fréquents ne dépend pas du nombre d'objets. Cela justifie des techniques d'échantillonnage pour obtenir une approximation du nombre de motifs fréquents [Lhote *et al.*, 2005b]. L'extraction des motifs dans une portion de la base permet de déterminer les constantes de la formule.

4.5 Modèle groupé

À l'aide de la modélisation de Bernoulli sous forme de groupes d'attributs de longueur t, nous obtenons des résultats similaires aux résultats précédents.

Théorème 4 *Pour un seuil fixe γ de fréquence minimal, le nombre $F_{m,n,\gamma}$ de motifs γ-fréquent vaut*

$$F_{m,n,\gamma} = \binom{n}{\gamma}\left(1+\left(\frac{1}{t}\right)^{\gamma-1}\right)^{m_1}\left[1+O\left(n\left(\frac{1+(1/t)^{\gamma}}{1+(1/t)^{\gamma-1}}\right)^{m_1}\right)\right]$$

Preuve : Voir annexe 2. □

Théorème 5 *Pour un seuil $\gamma = \lfloor \alpha n \rfloor$ proportionnel de fréquence minimale, le nombre moyen de motifs fréquents satisfait*

$$F_{m,n,\gamma} \sim \binom{m_1}{j_0} t^{\,j_0}, \quad \textit{avec} \quad j_0 = \left\lfloor \frac{-\log r}{\log t} \right\rfloor.$$

Preuve : Voir annexe 2. □

Les théorèmes 4 et 5 montrent que le comportement asymptotique pour un modèle groupé est très proche de celui obtenu avec une modélisation simple. Néanmoins, le nombre de motifs fréquents avec le modèle groupé est exponentiellement plus petit que celui fourni par le modèle simple. Le facteur entre les deux modélisations est donné par

$$\left(\frac{(1+(1/t)^{\gamma-1})^{1/t}}{1+(1/t)^{\gamma}}\right)^{m} = \delta^m \qquad \text{avec } \delta < 1$$

La présence de corrélations induit donc une **décroissance exponentielle** du nombre de motifs fréquents, même si le facteur δ est ici proche de 1. Ce nouveau modèle affine donc les résultats précédents et confirme le comportement polynomial.

4.6 Nombre de motifs fermés

Sous l'hypothèse du modèle simple de Bernoulli, nous montrons maintenant que le nombre de motifs fermés est équivalent à celui des motifs fréquents lorsque l'on adopte un seuil suffisant.

Théorème 6 *Soit un seuil fixe γ (hypothèse 2) de fréquence minimale. Selon l'hypothèse 1 de taille des bases de données, le nombre moyen de motifs fermés $C_{m,n,\gamma}$ dans une base de données de modèle simple* Bernoulli *de paramètre p vaut*

$$C_{m,n,\gamma} = \sum_{j=1}^{m}\binom{m}{j}\sum_{i=\gamma}^{n}\binom{n}{i}p^{ij}(1-p^i)^{m-j}(1-p^j)^{n-i}.$$

Preuve : Voir annexe 2. □

Théorème 7 (Nombre moyen de motifs fermés) *Pour* $\gamma > (1+\epsilon)\dfrac{\log m}{|\log p|}$, *les nombres de motifs fermés* $C_{m,n,\gamma}$ *et de motifs fréquents sont équivalents,*

$$C_{m,n,\gamma} \sim F_{m,n,\gamma}.$$

Preuve : Voir annexe 2. □

Ce résultat est également valable dans le modèle groupé de Bernoulli :

Théorème 8 *Si le seuil de fréquence minimale* γ *satisfait* $\gamma > \lfloor (1+\epsilon)\log m_1/\log t \rfloor$ *pour un* ϵ *strictement positif, alors dans un modèle groupé de* Bernoulli *le nombre moyen de motifs* γ*-fréquent et celui des motifs fermés est équivalent,*

$$C_{m,n,\gamma} \sim F_{m,n,\gamma}.$$

Preuve : Voir annexe 2. □

Lorsque les données ne sont pas corrélées, le nombre moyen de motifs fermés et celui des motifs fréquents sont équivalents. Dans les bases de données réelles, les motifs fermés sont bien connus pour être nettement moins nombreux que les motifs fréquents. Le chapitre 2 a montré que ce constat motive l'élaboration d'algorithmes très efficaces qui synthétisent les corrélations. Extraire des motifs fermés est plus compliqué que d'extraire des motifs fréquents et ce n'est pas justifié dans des bases non corrélées.

Corollaire 1 *Avec un seuil proportionnel* $\gamma = \lfloor \alpha n \rfloor$ *de fréquence tel que* $\gamma > (1+\epsilon)\dfrac{\log m}{|\log p|}$, *le nombre moyen de motifs fermés* $C_{m,n,\gamma}$ *est polynomial en nombre d'attributs,*

$$C_{m,n,\gamma} \sim \frac{m^s}{s!} \text{ avec } s = \lfloor \log \alpha / \log p \rfloor.$$

Preuve : Il s'agit d'une application directe des théorèmes 3 et 7. $(1+\epsilon)\dfrac{\log m}{|\log p|}$ est équivalent à $(1+\epsilon)\dfrac{c\log n}{|\log p|}$ et $(1+\epsilon)\dfrac{c\log n}{|\log p|}$ est négligeable devant $\alpha \cdot n$. □

La figure 4.4 représente le nombre de motifs fréquents et de motifs fermés selon un seuil variable, pour deux classiques bases de données de test : `T40I10D100K`, générée artificiellement par l'outil de Srikant [Agrawal et Srikant, 1994] et `pumsb`, qui concerne des erreurs de diagnostic dans des appareils électromécaniques. On constate que `T40I10D100K` correspond bien à notre modèle. Le théorème 7 s'applique quand $\gamma > 2$. Par construction, cette base est synthétique et ne contient pas de corrélation. Des expériences menées avec une autre base, `T10I4D100K`, mènent à la même conclusion [Lhote *et al.*, 2005a]. Remarquons que le comportement d'une base réelle comme `pumsb` n'est pas modélisé par ce théorème.

À ce jour, nous n'avons pas de résultat sur le nombre de motifs fermés lorsque le seuil est plus petit que $\dfrac{\log m}{|\log p|}$, ou dans l'hypothèse d'un seuil fixe. Il faut toutefois noter que le nombre

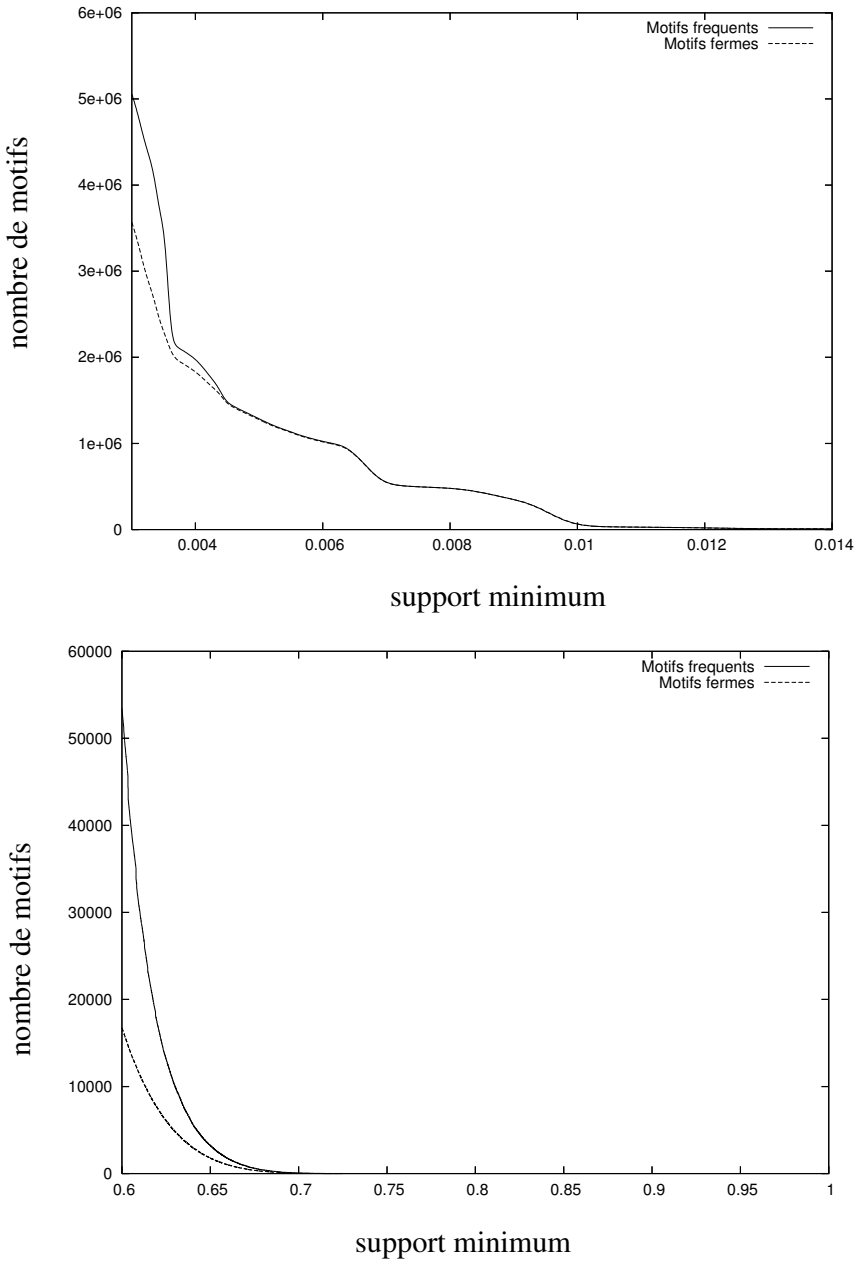

FIG. 4.4 – Équivalence entre le nombre de motifs fermés et fréquents (en haut `T40I10D100K` et en bas `pumsb`).

de motifs fermés de fréquence 1 est toujours inférieur au nombre d'objets, car un motif fermé est maximal dans sa classe d'équivalence de support. Un raisonnement similaire est valable pour le nombre de motifs fermés de fréquence 2, puis ainsi de suite jusqu'au seuil ci-dessus. Notre intuition est donc qu'il n'y a pas une quantité exponentielle de motifs fermés pour un seuil fixe.

4.7 Conclusion

Pour un seuil de fréquence minimal fixe, nous avons montré que le nombre moyen de motifs fréquents dans des bases respectant des modèles de Bernoulli ou Markov est exponentiel en nombre d'attributs et polynomial en nombre d'objets. Ce résultat sur la taille de la solution est analogue à celui fourni par des analyses de l'espace de recherche. Plus surprenant, le nombre moyen de motifs fréquents dans le cas d'un seuil proportionnel est polynomial en nombre d'attributs et ne dépend pas du nombre d'objets.

Dans les bases de données de Bernoulli, le nombre de motifs fermés est équivalent, au delà d'un certain seuil logarithmique en nombre d'attributs, à celui des motifs fréquents. En cas de seuil proportionnel, ce nombre est donc polynomial en nombre d'attributs et ne dépend pas du nombre de transactions.

Soulignons que la formule exacte qui donne le nombre moyen de motifs fermés dans une base de données (théorème 6) est symétrique entre le nombre d'attributs et le nombre de transactions, lorsque le seuil minimal vaut 1, c'est-à-dire lorsque l'on s'intéresse à **tous** les motifs fermés. Cette information est précieuse car elle indique qu'il y a en moyenne autant de motifs fermés présents dans une base que dans sa version transposée, où les valeurs de m et de n permutent. Nous exploitons cette propriété au chapitre 8 page 111 pour proposer une méthode d'extraction dans les contextes difficiles comportant un grand nombre d'attributs devant le nombre d'objets.

En deçà du seuil d'équivalence entre fermés et fréquents, nous ne maîtrisons que le comportement du nombre de motifs fréquents et pas celui des motifs fermés. Entre un comportement asymptotique exponentiel ou polynomial, notre intuition est qu'il y a pour le nombre de motifs fermés un intermédiaire entre ces deux extrêmes. L'asymptotique pour l'expression est certainement symétrique en m et n, grâce aux propriétés de transposition (cf. section 8.2 page 112).

Conclusion

Après avoir introduit l'intérêt des motifs fréquents pour la fouille de données, nous avons détaillé un formalisme général pour la majorité des algorithmes d'extraction de motifs. Sur le modèle de `Guess & Correct` qui génère des candidats potentiels à partir d'une solution partielle, augmentant progressivement la qualité de solution, nous avons expliqué deux méthodes de parcours de l'espace de recherche pour déterminer l'ensemble des motifs vérifiant une contrainte antimonotone. La première, par niveaux, utilise des générateurs d'égale longueur ce qui simplifie la méthode de production des candidats. La seconde, en profondeur, extrait directement les motifs maximaux en utilisant les traverses minimales d'hypergraphe, un problème classique de la théorie de la complexité.

Cette étude formelle a conclu que l'algorithme classique par niveaux répond à la majorité des besoins : il est simple à mettre en œuvre et le faible nombre d'accès aux données qu'il effectue est bien adapté aux bases ayant de nombreux objets, le cas le plus courant. L'élégant algorithme en profondeur ne montrera sa supériorité qu'au détriment d'une heuristique, mais la condensation des résultats qu'il fournit est remarquable. Les maximaux sont directement calculés et permettent de regénérer l'ensemble des motifs valides. Sa capacité à proposer une solution partielle rapidement est précieuse.

Par la suite, nous avons justifié le besoin de représentations condensées des motifs fréquents, tant pour obtenir des résumés des motifs que pour améliorer les performances des algorithmes classiques. Les motifs fermés, libres et δ-libres ont été introduits, qui permettent d'avoir une vision plus profonde des difficultés de l'extraction de motifs.

La présentation des motifs k-libres permet de généraliser ces notions. Ils sont une étape importante pour la construction de règles d'associations généralisées informatives. Ces règles sont utiles pour l'extraction de connaissances dans les bases de données incomplètes et la mise au point de règles de classification.

Enfin, nous avons réalisé une analyse en moyenne du nombre de motifs fréquents ou fermés dans des bases de données aléatoires générées par des modèles de Bernoulli ou de Markov. Cette étude a montré que l'extraction des motifs est particulièrement difficile dans le cas où le seuil de support minimum est très faible. Plus surprenant, lorsque ce seuil est proportionnel au nombre

d'objets, le nombre de motifs n'est plus exponentiel en nombre d'attributs mais polynomial. D'autre part, le nombre de motifs fermés est équivalent au nombre de motifs fréquents dans ces bases aléatoires. Cet argument justifie l'emploi de représentations condensées à base de motifs fermés ou libres uniquement dans le cas où la base est corrélée.

Nous pensons que les méthodes d'extraction sont désormais au point, particulièrement en ce qui concerne les représentations condensées, qui ont permis de contourner les difficultés de l'extraction des motifs fréquents. S'il reste beaucoup de questions et de nombreux choix dans les détails de l'implémentation des algorithmes, la résolution de ce problème est bien avancée.

Cependant, dans certaines configurations, l'extraction de motifs reste une tâche difficile, voire impossible. Nous avons choisi dans ce document de nous concentrer plus précisément sur deux verrous bien identifiés par la communauté fouille de données. Il s'agit des bases de données comportant des valeurs manquantes (partie II), comme c'est souvent le cas dans les applications mettant en œuvre des données issues de processus réels. L'autre point concerne des bases ayant un grand nombre d'attributs devant le nombre d'objets (partie III) à l'image des données issues du séquençage du génome. Dans ces parties, nous utiliserons les notions développées précédemment, car la k-liberté et la connexion de Galois sont centrales dans les propriétés que nous allons mettre en évidence.

Deuxième partie

Fouille de données dans les bases comportant des valeurs manquantes

Introduction

La présence de valeurs manquantes dans les bases de données est un problème ancien qui s'est toujours posé lors de l'exploitation de données réelles. Dans le domaine médical, le protocole d'expérimentation est à forte variabilité humaine, les données à exploiter sont rarement complètement renseignées : le temps manque pour collecter un résultat d'examen ou encore le patient n'est pas en état de le supporter. Un autre exemple concerne les sondages d'opinion, que les interviewés prennent rarement la peine de remplir complètement.

Les méthodes classiques de fouille de données (classification supervisée et non supervisée, associations) ne sont pas conçues pour prendre en compte les données incomplètes. Nous verrons au chapitre 5 que la plupart des traitements des valeurs manquantes introduisent de nombreux biais dans l'analyse. Le problème se pose également pour les représentations condensées.

La contribution de ce travail sur les valeurs manquantes est triple. Premièrement nous décrivons les dommages causés par les valeurs manquantes sur les représentations condensées de motifs. Deuxièmement nous choisissons, par des calculs menés sur la base incomplète, de caractériser des propriétés dans la base complète dont elle est issue. Nous proposons une méthode de traitement des motifs en présence de valeurs manquantes pour les représentations condensées fondées sur les motifs k-libres. Nous montrons la consistance de la représentation obtenue d'un point de vue formel, c'est-à-dire que les extractions de motifs k-libres dans la base incomplète menées selon nos définitions fournissent des motifs également k-libres dans la base complète. Des expérimentations confirment la portée de ce résultat. Nous pensons qu'il s'agit d'une amélioration importante pour les multiples usages de ces représentations condensées dans des bases incomplètes : d'une part les données sont exploitées dans leur intégralité, valeurs manquantes comprises ; d'autre part les connaissances extraites sont qualifiées relativement à la base complète. Nous proposons pour finir une méthode originale d'extraction de règles d'association généralisées informatives en présence de valeurs manquantes, fondée sur l'utilisation de la base opposée.

Cet exposé est divisé en trois chapitres :

– le chapitre 5 décrit l'état de l'art et pose le problème des valeurs manquantes dans les bases de données. Nous y énonçons et explicitons nos hypothèses de travail. Puis nous détaillons le codage que nous réalisons pour ces valeurs et décrivons, à l'aide d'exemples simples, les

effets induits sur les représentations condensées ;

- le chapitre 6 s'intéresse à la recherche d'information fiable en présence de valeurs manquantes. Pour cela, nous formalisons le calcul de support des motifs généralisés et de la k-liberté dans les bases incomplètes. Nous concevons un mode de calcul pour la k-liberté, dont nous montrons qu'il est consistant avec la propriété correspondante dans la base complète ;
- le chapitre 7 propose l'utilisation de la base opposée pour confirmer les associations découvertes et obtenir une fermeture généralisée qui permet de former des règles d'associations généralisées informatives.

Chapitre 5

Extraction de connaissances dans les bases de données comportant des valeurs manquantes

Ce chapitre introduit le problème des valeurs manquantes dans les bases de données. Ce phénomène se produit fréquemment dans le cas de données issues de problèmes réels, mais également lors de la fusion de données en provenance de plusieurs sources, pour des raisons d'incompatibilité entre les différents formats.

Nous commençons section 5.1 par établir un état de l'art bibliographique des principales méthodes d'appréhension des valeurs manquantes, en examinant les solutions mises en œuvre par les techniques statistiques, ou la théorie des bases de données. À partir de ces fondements, nous présentons notre positionnement section 5.2 pour la prise en compte des valeurs manquantes. La section 5.3 précise le codage que nous utilisons pour les données incomplètes. Enfin, la section 5.4 souligne, à partir d'un exemple, les effets négatifs induits par ces valeurs lors de l'extraction de connaissance et la section 5.5 montre leur impact sur des données de test.

5.1 État de l'art

Nous relatons ci-dessous les principales tendances des travaux sur valeurs manquantes, du point de vue des méthodes d'apprentissage automatique et de la fouille de données en particulier. Nous examinons les solutions envisagées dans les domaines de la statistique et des bases de données. Puis nous présentons brièvement des travaux relevant du domaine des ensembles flous ou d'approximation. Nous terminons par des méthodes plus proches de nos travaux, qui concernent les règles d'association.

Au cours de cette présentation, nous rapporterons des exemples de méthodes qui réalisent

une *complétion* ou *imputation* des valeurs manquantes, c'est-à-dire qui proposent une valeur de remplacement pour chaque valeur inconnue. Ces méthodes sont souvent évaluées par la précision d'un test de classification ou par introduction artificielle de valeurs manquantes et comparaison de la base imputée à la base originale.

5.1.1 Statistique

Les statistiques définissent trois modèles de probabilité de valeurs manquantes [Agarwal, 2001] :

1. *missing completely at random (MCAR)*, quand cette probabilité est complètement indépendante des données ;
2. *missing at random*, quand cette probabilité ne dépend pas des valeurs manquantes, mais des valeurs *observées* (par exemple, quand une pièce n'est pas défectueuse, on n'observe pas la taille du défaut) ;
3. *not missing at random*, quand la cause des valeurs manquantes peut-être expliquée par des relations entre les valeurs observées, mais également entre les valeurs manquantes (lorsque la taille du défaut d'une pièce n'est pas observée, d'autres paramètres connexes sont également manquants).

Notons que ce modèle ne prend pas en compte les configurations où les valeurs manquantes sont le résultat d'une fusion de données de formats différents, à moins d'introduire un attribut pour qualifier la provenance des données. Pour des illustrations de tous ces modèles, le chapitre 12 étudie un exemple de données médicales qui cumulent toutes ces causes.

Dans la suite de notre travail, nous ne ferons pas d'hypothèse statistique particulière concernant le modèle de probabilité des valeurs manquantes. Pour des besoins de modélisation, nous verrons que nous définirons à la section 6.1 un opérateur qui transforme une base complète en une base incomplète, sur laquelle nous effectuons les calculs.

La contribution majeure des statistiques pour le traitement des données manquantes concerne l'algorithme EM (espérance/maximisation) [Little et Rubin, 1985], qui simule les valeurs possibles cachées par les valeurs manquantes et réalise une analyse statistique combinée des résultats obtenus. Le but n'est pas de compléter les données mais de refléter correctement l'incertitude des valeurs manquantes et de préserver les aspects des distributions [Nakache et Gueguen, 2000].

5.1.2 Bases de données

Dans le domaine des bases de données, l'expression « valeur nulle » est plus couramment utilisée que celle de valeur manquante. Les problèmes posés sont : comment effectuer des requêtes sur des bases incomplètes ? Comment élaborer des schémas, réaliser des fusions de données ?

Dans [Dyreson, 1997], Curtis Dyreson répertorie 438 publications sur des thèmes très divers : valeurs nulles, logique, exécution de requêtes, design de schémas, analyses de complexité. À l'aide d'une étude de la quantité d'articles parus par année, l'auteur montre que ce thème a émergé au milieu des années 70 et a connu une apogée à la fin des années 80.

Plus récemment, des travaux ont étudié des problèmes comme celui des dépendances fonctionnelles. Celles-ci sont utilisées pour la rétro conception de bases de données et entre autres pour vérifier les cohérences. [Levene et Loizou, 1999] examine ce cadre en présence de valeurs nulles, sous l'angle de l'additivité. Les auteurs montrent que la transitivité énoncée par les axiomes fondateurs d'Armstrong [Armstrong *et al.*, 2002] n'est plus conservée. Ce résultat n'est pas surprenant, car les travaux de Codd [Codd, 1979] ont montré qu'on ne pouvait pas espérer étendre l'ensemble de l'algèbre relationnelle à ce type de base. Cependant, [Bosc *et al.*, 2002] donne des exemples en logique où, même avec des informations incomplètes, on peut répondre à des questions avec certitude. Par exemple, sans connaître le véritable emploi d'une personne, on pourra malgré tout affirmer qu'elle cotise à la Sécurité Sociale.

5.1.3 Ensembles flous et d'approximation

Il existe deux extensions à la théorie des ensembles : les *ensembles flous* (fuzzy sets) et les *ensembles d'approximation* (rough sets). Les ensembles flous permettent de représenter des classes d'éléments dont la frontière entre appartenance et exclusion n'est pas brutale mais graduelle. Ils dérivent de la logique floue et permettent d'obtenir des règles dont la conclusion est de nature probabiliste. [Bosc *et al.*, 2002] revient largement sur cette théorie, prolongé par [Bosc *et al.*, 2004], traitant également de la théorie des possibilités.

Pour la tâche de classification, les ensembles d'approximation proposent des classes d'équivalence qui rassemblent les instances. Ces ensembles symbolisent des bornes (pour l'inclusion). Pour les valeurs manquantes, [Grzymala-Busse et Hu, 2001] a proposé plusieurs algorithmes de classification et comparé plusieurs techniques d'imputation possibles :

- remplacer par la valeur la plus fréquente ;
- remplacer par la valeur la plus fréquente au sein de la classe ;
- appliquer la méthode de C4.5 de traitement des valeurs manquantes, fondée sur l'entropie et qui partitionne les exemples incomplets [Quinlan, 1993] ;
- remplacer par toutes les possibilités ;
- remplacer par toutes les possibilités en se restreignant aux valeurs possibles dans la classe ;
- ignorer les exemples incomplets ;
- utiliser une technique de sélection d'exemples statistiquement indépendants et compléter de façon probabiliste ;
- désactiver temporairement les exemples incomplets suivant les attributs prédits ;

– créer une valeur particulière « manquante » pour chaque attribut.

L'auteur conclut que la meilleure méthode consiste à générer toutes les possibilités, mais est impraticable. Cette idée est exploitée par [Latkowski, 2003], qui propose de décomposer la base de données incomplète en plusieurs bases complètes de dimensions différentes, formant ainsi un genre de pavage. Il fusionne les règles obtenues dans les différentes portions. L'inconvénient récurrent de ces méthodes à base d'imputation multiple est que des conflits surgissent lors de la phase d'imputation.

5.1.4 Arbres de décision

Quinlan [Quinlan, 1987] propose plusieurs stratégies pour les différentes étapes de construction d'un arbre de décision que sont l'évaluation des exemples selon un attribut, la partition de l'ensemble d'apprentissage et classification d'un exemple. Différentes stratégies d'action face aux valeurs manquantes sont évaluées sur un jeu de données particulier. Ses conclusions, pour les trois étapes, sont les suivantes :

– lors de l'évaluation des tests, les approches qui ignorent les cas incomplets ne se comportent pas correctement ;
– lors du partitionnement, cette ignorance est fatale. Il vaut bien mieux assigner chaque exemple, dont la valeur de l'attribut est inconnue, à la fraction qui a la meilleure probabilité selon les valeurs connues ;
– pendant la classification, la combinaison de toutes les réponses possibles donne de bonnes performances.

5.1.5 Règles d'association

À l'origine, le problème des valeurs manquantes n'était pas implicitement défini en fouille de données. En effet, lors de l'analyse des transactions de clients de supermarché, un produit est présent ou absent, mais pas manquant. Les valeurs manquantes apparaissent lorsque les transactions proviennent de données au format attribut/valeur, comme nous le verrons à la section 5.3.

Notre tâche poursuit un travail de thèse initié Arnaud Ragel (GREYC) [Ragel, 1999], motivé par l'imputation des valeurs manquantes pour les arbres de décision. À la recherche de règles permettant de proposer une valeur de remplacement, l'auteur a étudié les règles d'association à des fins de prédiction [Ragel et Crémilleux, 1999] et leur calcul dans les bases incomplètes [Ragel et Crémilleux, 1998], en redéfinissant la fréquence et la confiance dans ces conditions.

Nous présentons rapidement deux autres contributions. La première [Nayak et Cook, 2001] concerne une méthode simple d'imputation, fondée sur la probabilité des différentes valeurs d'un attribut. De cette façon, la condition de présence d'un motif dans un objet n'est plus

booléenne, mais probabiliste. La deuxième contribution, initiée dans [Jami *et al.*, 1998] et revisitée dans [Jami *et al.*, 2004], calcule des règles de prédiction pour les valeurs manquantes. Ces règles déterminent des intervalles pour les attributs continus. Les mesures utilisées (fréquence, confiance, gain de précision) possèdent des caractéristiques antimonotones, mises à profit par un algorithme d'extraction par niveaux. Cette méthode donne de bons résultats lors de l'application des règles de prédiction pour imputer une base incomplète générée artificiellement, en comparaison avec la base originale.

5.1.6 Combinaison de classifieurs

[Zheng et Toh Low, 1999] étudie l'intérêt des méthodes de combinaison de classifieurs (committee learning), comme le bagging et le boosting, face à des bases incomplètes. Cet article propose de pondérer les résultats des membres du comité (chaque classifieur individuel) selon leur réussite en présence de valeurs manquantes.

[Kerdprasop *et al.*, 2003] reproduit les expériences décrites à la section 5.1.3, avec trois méthodes de classification : le Bayes naïf, les arbres d'induction et les plus proches voisins. Il ajoute une méthode de complétion qui utilise un principe de régression. Il suggère de créer une valeur spéciale « manquante » et indique que l'utilisation de la moyenne améliore les performances des réseaux bayésiens.

5.1.7 Autres

Pour finir cette rapide revue des travaux du domaine, nous citerons un travail original [Liu *et al.*, 1997, Liu *et al.*, 1998b], dans lequel les auteurs cherchent des trous dans les bases de données. En considérant chaque objet comme un point de l'espace, ils y cherchent des points absents. Ils pensent que ces trous expriment une impossibilité d'imputation des valeurs manquantes. Malheureusement, les auteurs n'ont pas poursuivi leur travail au delà du difficile problème du calcul de ces trous.

5.2 Positionnement de notre travail

À la lumière des travaux relatés ci-dessus, nous orientons notre travail sur les valeurs manquantes selon deux grands principes :

– nous ne voulons pas imputer les valeurs manquantes. Les conclusions des diverses publications dans ce domaine sont mitigées, voire contradictoires et on sent bien qu'il faudrait mieux prendre en compte les différents modèles d'apparition des valeurs manquantes. Si l'extraction de règles en présence de valeurs manquantes est une réussite (les simulations montrent que les règles qui seraient obtenues s'il n'y avait pas de valeurs manquantes

sont très correctement retrouvées), ces règles sont délicates à utiliser pour compléter de façon fiable les données incomplètes [Ragel et Crémilleux, 1999]. En outre, les méthodes d'imputation doivent inévitablement faire face à des conflits ;

– il n'est pas raisonnable d'ignorer définitivement des objets au prétexte qu'ils ne sont pas complets. Nous souhaitons extraire des connaissances à partir de la base incomplète, sans la réduire à une quelconque portion complète qui de fait serait particulièrement congrue. *A contrario*, nous adopterons une démarche d'oubli temporaire pendant le calcul des fréquences, sous le terme de *désactivation*.

Nous réserverons donc un traitement particulier à toute valeur manquante en étendant le domaine de définition des valeurs d'un attribut. Comme nous l'avons exprimé précédemment (cf. section 5.1.1), nous ne ferons pas d'hypothèse statistique particulière concernant le modèle de probabilité des valeurs manquantes. Nous définissons pour cela à la section suivante un opérateur de modélisation d'une base avec des valeurs manquantes, à partir d'une base de données sans valeur manquante.

Sous ces hypothèses, le chapitre suivant montre qu'il est possible de découvrir des connaissances valides dans la base complète, comme cela avait été signalé dans [Bosc *et al.*, 2002]. Ce principe n'est pas si surprenant : si l'on considère que les valeurs manquantes occultent la véritable valeur d'une donnée, les fréquences d'apparition de certains motifs vont diminuer, car la décision de présence n'est plus possible dans certains objets. Un motif fréquent dans une base incomplète est donc *a fortiori* fréquent dans la base complète. C'est cette propriété que nous développerons pour les motifs k-libres.

Nous pensons que ce positionnement s'inscrit plus globalement dans la démarche des représentations condensées. En effet, elles correspondent à une étape intermédiaire entre le jeu de données et la méthode d'exploration utilisée. Leur utilisation en présence de valeurs manquantes évite d'imputer ces valeurs, lors d'une phase préliminaire de traitement des données. Puis, en aval du processus, l'utilisation de ces représentations permet d'aborder toutes les méthodes d'exploration. Ceci permet une prise en compte des valeurs manquantes qui est indépendante du but final.

5.3 Codage des données en présence de valeurs manquantes

Supposons que certains attributs de l'exemple de la table 1.1 page 12 n'aient pas pu être mesurés, ignorant par exemple, pour certains objets, si $X_1 = a_1$ ou $X_1 = a_2$. Une valeur manquante apparaît et nous utilisons le caractère `'?'` à la place de la valeur dans le cas attribut/valeur et dans le format booléen pour chaque attribut booléen codant cet attribut multivalué. La table 5.1 illustre ce codage où trois valeurs manquantes ont été introduites dans r.

La nouvelle base obtenue est appelée $mv(r)$. Nous utilisons cette notation car nous disons

objet	r (booléen)						
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_1	×		×		×		
o_2		×	×		×		
o_3	×		×		×		
o_4	×			×		×	
o_5		×	×			×	
o_6		×	×			×	
o_7	×			×			×
o_8		×		×			×

$mv(r)$ (attr/val)		
X_1	X_2	X_3
+	→	0, 2
−	→	0
+	→	?
+	←	0, 4
−	→	0, 6
?	→	0, 5
+	←	1
−	?	0, 8

$mv(r)$ (booléen)						
a_1	a_2	a_3	a_4	a_5	a_6	a_7
×		×		×		
	×	×		×		
×		×		?	?	?
×			×		×	
	×	×			×	
?	?	×			×	
×			×			×
	×	?	?			×

TAB. 5.1 – Présence de valeurs manquantes dans r. *La valeur manquante est soulignée lorsqu'elle coïncide avec la valeur dans la base complète. Lors de l'analyse de bases de données réelles, cette information n'est pas disponible.*

qu'il est toujours possible de considérer une base complète r, une opération $mv()$ qui introduit des valeurs manquantes et le résultat est une base incomplète $mv(r)$. L'opérateur $mv()$ est par exemple relié au processus réel qui engendre les données, auquel cas r est indisponible ; ou bien $mv()$ désigne une opération automatique, comme masquer aléatoirement la valeur de certaines variables. Cet opérateur est défini formellement au début du chapitre suivant, à la section 6.1.

5.4 Effet des valeurs manquantes sur les représentations condensées de motifs k-libres

À partir de l'exemple pédagogique que nous développons au cours de ce mémoire, cette section se propose de montrer intuitivement les dégâts causés par les valeurs manquantes sur les représentation condensées.

La table 5.2 fournit les motifs 1-libres de fréquence minimale 2 pour la base complète de la table 1.2 page 13. Pour chaque motif, nous indiquons sa fermeture. De cette représentation condensée, nous pouvons par exemple déduire la règle informative $a_1a_3 \rightarrow a_5$, présente deux fois dans les données (l'attribut a_5 est toujours présent avec le motif a_1a_3).

Comment agir avec les valeurs manquantes ? Prenons la méthode élémentaire consistant à retirer ces valeurs (un attribut manquant est déclaré absent et nous dirons qu'il a été *ignoré*). Quand un attribut est manquant dans un objet, aucun motif contenant cet attribut ne peut y être présent. Il y aura donc diminution de la fréquence de ces motifs et perte d'association, ce qui désagrège la représentation condensée.

1-libre	fermeture	1-libre	fermeture
a_1		a_1a_3	a_5
a_2		a_1a_4	
a_3		a_1a_5	a_3
a_4		a_2a_3	
a_5	a_3	a_2a_6	a_3
a_6		a_3a_6	a_2
a_7	a_4		

TAB. 5.2 – Représentation condensée de r.

La table 5.3 liste les motifs 1-libres extraits de $mv(r)$. On constate par exemple que le motif a_1a_4 n'est plus 1-libre. En outre, cette table contient des motifs, par exemple a_4a_7, qui ne sont pas présents dans la représentation issue de la base originale r. Nous qualifierons plus tard ces motifs de *pollution* (voir section 5.5). Notons également que de nombreux attributs ont disparu des fermetures, par exemple a_5 n'est plus dans la fermeture de a_1a_3 et consécutivement la règle $a_1a_3 \rightarrow a_5$ n'est plus retrouvée.

1-libre	fermeture	1-libre	fermeture
a_1		a_1a_3	
a_2		a_1a_5	
a_3		a_2a_3	
a_4	a_1	a_2a_5	a_3
a_5	a_3	a_2a_6	a_3
a_6		a_3a_6	
a_7		a_4a_7	

TAB. 5.3 – Représentation condensée de $mv(r)$ en ignorant les valeurs manquantes.

Les valeurs manquantes produisent des effets à la fois sur les motifs libres et sur les attributs des fermetures. Concernant un motif X 1-libre, supposons qu'un attribut a appartienne dans la base complète à la fermeture de X : cela signifie que a est toujours présent avec X. Si des valeurs manquantes se produisent sur a, il existe donc des objets pour lesquels cette association est rompue : a sort alors de la fermeture de X (effet sur la fermeture) et Xa peut devenir libre (effet sur le libre). Sur notre exemple, a_4 est dans la fermeture de a_7 dans r, tandis que a_4 sort de cette fermeture dans $mv(r)$ à cause de la valeur manquante de l'objet o_8. En outre, a_4a_7 est devenu libre et participe à la pollution. Le chapitre suivant indique comment aménager le

calcul de la liberté dans un contexte incomplet, pour d'éviter la pollution inhérente aux valeurs manquantes.

Clairement, les valeurs manquantes sont responsables de la disparition d'associations et de la pollution des motifs libres avec des motifs invalides.

5.5 Mise en évidence des effets des valeurs manquantes sur des données de l'UCI

Le but de cette section est de mesurer précisément la pollution sur les motifs k-libres induite par la méthode usuelle de traitement des valeurs manquantes consistant à les ignorer. Nous commençons par calculer les motifs 3-libres 1 %-fréquents dans des bases r de test de l'UCI [Blake et Merz, 1998]. Ensuite, nous introduisons des valeurs manquantes dans ces bases selon une loi de Bernoulli. Ces valeurs sont ignorées en les remplaçant des valeurs absentes et nous notons $mv(r)^\circ$ la base correspondante.

Les figures 5.1 à 5.3 comparent les motifs 3-libres 1 %-fréquents extraits de r et de $mv(r)^\circ$. Elles indiquent le taux de pollution à mesure que le nombre de valeurs manquantes augmente, c'est-à-dire le nombre de motifs aberrants de $mv(r)^\circ$ proportionnellement au nombre de motifs corrects de r à découvrir. Les résultats pour les différentes bases de test sont répartis en trois figures, suivant l'importance de la pollution. Le nombre de motifs k-libres dans la base originale est indiqué après le nom de chaque base.

La figure 5.1 rassemble les mesures pour les bases qui enregistrent une faible pollution, inférieure à 8 %. La figure 5.2 montre une pollution plus conséquente, inférieure à 100 %. Enfin, la pollution est très critique à la figure 5.3 où elle dépasse les 100 % et atteint des sommets.

Ces expériences utilisent des bases de données sans valeurs manquantes et permettent donc de mesurer la pollution. Dans des conditions réelles, il est bien sûr impossible de discerner les bon motifs des mauvais. Peut-on raisonnablement utiliser les motifs calculés pour la fouille de données ? Comment savoir si la base utilisée pour l'extraction de connaissance subira une pollution faible ou importante ?

Ces expériences montrent la pollution sur les motifs k-libres qui est induite par le fait d'ignorer les valeurs manquantes : il est impossible d'utiliser ces motifs de façon fiable et il est clair que le calcul de la représentation condensée doit être aménagé en présence de valeurs manquantes. La suite de cette partie propose une solution à ce problème ouvert. Nous verrons en particulier qu'avec de nouvelles définitions, les calculs menés dans $mv(r)$ déterminent des motifs k-libres dans r, ce qui évite toute pollution.

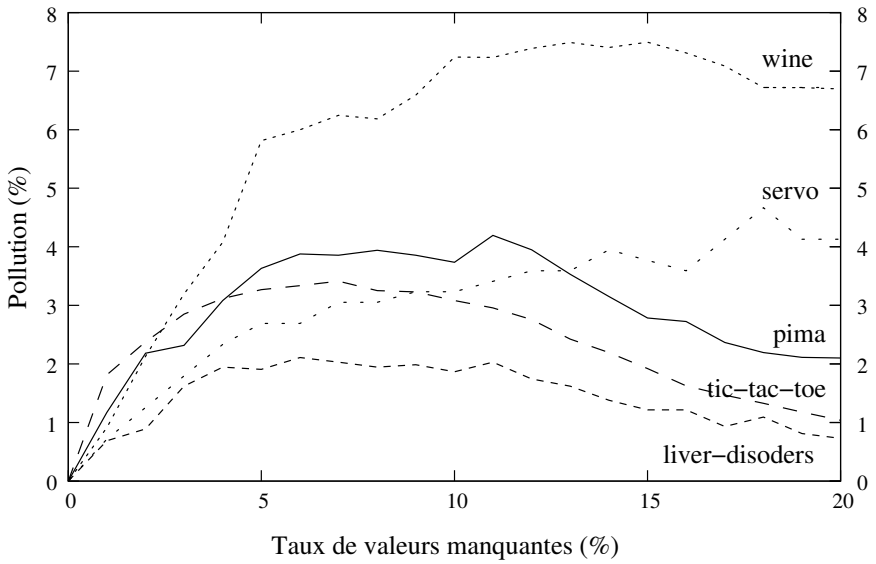

FIG. 5.1 – Pollution lors du calcul de 3-libres dans les bases `pima` (9 799 motifs), `wine` (33 775), `liver-disorders` (2 464), `servo` (557) et `tic-tac-toe` (16 905).

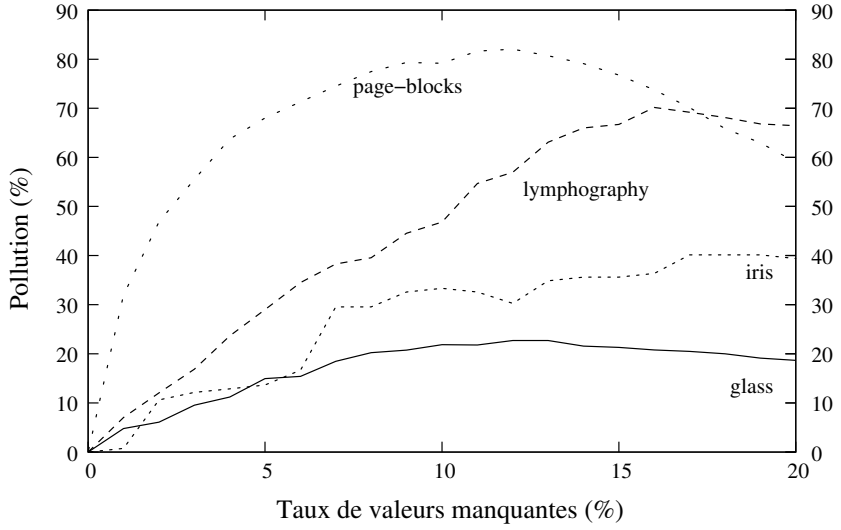

FIG. 5.2 – Pollution lors du calcul de 3-libres dans les bases `iris` (132 motifs), `glass` (4 999), `lymphography` (18 602) et `page-blocks` (11 160).

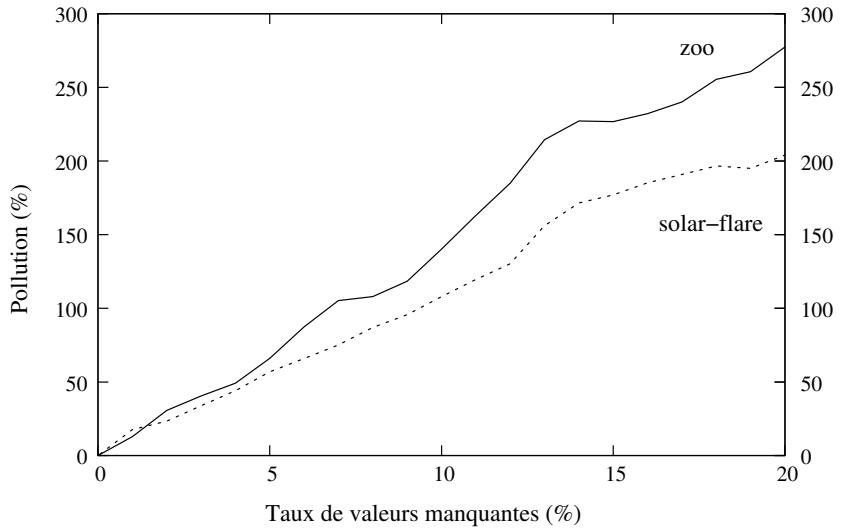

FIG. 5.3 – Pollution lors du calcul de 3-libres dans les bases `zoo` (1 468 motifs) et `solar-flare` (2 102).

Chapitre 6

Consistance du calcul de motifs k-libres en présence de valeurs manquantes

Dans ce chapitre, nous montrons comment effectuer dans une base incomplète l'extraction de motifs qui sont k-libres dans la base complète correspondante. À l'aide de la désactivation temporaire des objets incomplets, nous mettons en évidence des relations entre la fréquence des motifs généralisés dans les bases complète et incomplète. Ces relations permettent, en présence de valeurs manquantes, de déterminer dans la base incomplète des connaissances sur les motifs k-libres, qui sont consistantes avec les connaissances exactes dans la base complète dont elle est issue.

La section 6.1 formalise l'opérateur de modélisation des valeurs manquantes introduit au chapitre précédent lors du positionnement de notre travail. La désactivation d'objets incomplets est définie à la section 6.2. La section 6.3 explique notre méthode de calcul de la k-liberté et montre sa consistance. Pour finir, la section 6.4 détaille l'algorithme complet d'extraction des motifs k-libres en présence de valeurs manquantes.

6.1 Opérateur de modélisation des valeurs manquantes

Comme nous l'avons explicité à la section 5.2, notre positionnement sur le problème des valeurs manquantes requiert l'utilisation d'un opérateur de modélisation des valeurs manquantes. Celui-ci définit la relation entre une base incomplète et toute base complète correspondante.

Définition 21 (Opérateur de modélisation de valeurs manquantes) *Soit $r = (\mathcal{A}, \mathcal{O}, R)$ un contexte booléen. Un opérateur $mv()$ est appelé opérateur de modélisation de valeurs manquantes s'il transforme r en $mv(r) = (\mathcal{A}, \mathcal{O}, mv(R))$. La nouvelle relation binaire $mv(R)$ prend ses valeurs dans $\{present$,*

absent, manquant} *et satisfait les propriétés suivantes, pour tout attribut* a *de* $\mathcal{A}$ *et tout objet* o *de* $\mathcal{O}$*, et* $valeur \in \{present, absent\}$ *:*

1. $mv(R)(a,o) = valeur \Rightarrow R(a,o) = valeur$;
2. $R(a,o) = valeur \Rightarrow mv(R)(a,o) \in \{valeur, absent\}$;

L'opérateur $mv()$ modélise un effacement des données. Dans le cas où une valeur est manquante dans $mv(r)$, il est donc impossible de connaître la valeur originale dans r et c'est une propriété de compatibilité forte. Quand la valeur est connue dans $mv(r)$, elle l'est également dans la base complète et c'est la même valeur. En revanche, la deuxième propriété de $mv()$ assure qu'une valeur présente ou absente dans r conservera cette qualité à l'identique dans $mv(r)$, ou sera manquante.

6.2 Désactivation temporaire d'objets

La notion de désactivation permet de qualifier l'écart de fréquence d'un motif entre les bases complètes et incomplètes. En effet, en présence de valeurs manquantes, les fréquences décroissent. Sur notre exemple (table 1.2), $supp(a_3a_5, r) = 3$ mais $supp(a_3a_5, mv(r)) = 2$ (table 5.1). Pour pouvoir calculer correctement la fréquence d'un motif X dans $mv(r)$, il est nécessaire de distinguer les objets de $mv(r)$ qui ont une valeur manquante parmi les attributs de X. Ces objets vont être temporairement désactivés pour calculer une estimation de $supp(X, r)$ à l'aide de $supp(X, mv(r))$, car il est impossible de décider si oui ou non ils contiennent X.

6.2.1 Désactivation pour un motif classique

Nous commençons par définir formellement les objets désactivés relativement à un motif classique :

Définition 22 (Objet désactivé) *Pour un motif classique* $X \subseteq \mathcal{A}$*, un objet* $o \in \mathcal{O}$ *est désactivé si* $\forall a \in X, mv(R)(a,o) \neq absent$ *et* $\exists a \in X$ *t.q.* $mv(R)(a,o) = manquant$*. Nous notons* $Des(X, mv(r))$ *pour les objets de* $mv(r)$ *désactivés pour* X*.*

La proposition suivante nous sera utile au cours des développements à venir. Si une base r' contient plus d'objets qu'une autre base r, les objets désactivés pour X de r forment un sous ensemble de ceux qui sont désactivés dans r' :

Proposition 3 *Soit* $X \subseteq \mathcal{A}$ *un motif,* $r = (\mathcal{A}, \mathcal{O}, R)$ *et* $r' = (\mathcal{A}, R', \mathcal{O}')$ *deux bases de données telles que* $\mathcal{O} \subseteq \mathcal{O}'$ *et* $R \subseteq R'$*. Alors* $Des(X, mv(r)) \subseteq Des(X, mv(r'))$*.*

La figure 6.1 illustre la notion de désactivation, en représentant simultanément la base complète r (sur la gauche) et la base incomplète $mv(r)$ (sur la droite). On suppose que chaque objet

de la moitié supérieure de la base contient X et cette moitié de la base est repérée r_X. La moitié inférieure est repérée $r_{\overline{X}}$.

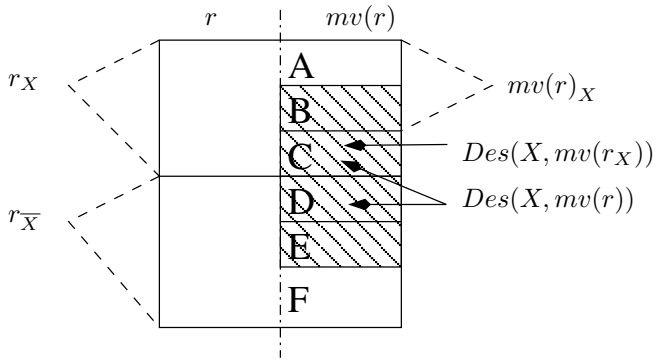

FIG. 6.1 – Base $mv(r)$ et objets désactivés pour X.

Sur la droite, la zone hachurée désigne les objets de $mv(r)$ qui contiennent des valeurs manquantes. Elle est constituée de six ensembles d'objets qui nous décrivons ci-dessous, en indiquant leur composition exacte pour notre exemple de la table 5.1, avec $X = a_2a_3$:

Région A : (o_2, o_5) les objets sans valeur manquante, contenant X ;

Région B : (aucun objet dans notre exemple) les objets contenant initialement X, dont les valeurs manquantes n'occultent pas la présence de X. Ces objets appartiennent à $mv(r)_X$;

Région C : (o_6) les objets contenant initialement X, dont les valeurs manquantes cachent la présence de X et constituent $Des(X, mv(r_X))$;

Région D : (o_8) les objets ne contenant pas X dans la base complète mais qui pourraient le contenir avec un remplacement judicieux des valeurs manquantes. Certains attributs s'excluant les uns les autres quand les données binaires proviennent de formats attribut-valeur, un attribut manquant cache autant de valeurs manquantes que l'attribut a de valeurs possibles. Dans notre exemple, l'objet o_8 ne contient pas le motif a_2a_3 dans la base complète. Plus rien n'indique cette absence quand il y a une valeur manquante sur le deuxième attribut X_2, qui induit des valeurs manquantes sur a_3 et a_4. Dans la pratique, ces objets seront désactivés à tort ;

Région E : (o_3) les objets incomplets ne contenant X ni dans la base originale, ni après une quelconque substitution des valeurs manquantes ;

Région F : (o_1, o_4, o_7) les objets complets qui ne contiennent pas X.

À l'examen de la base de données incomplète $mv(r)$, les objets seront donc répartis en trois groupes pour décider du support de X :

Régions A et B : les objets qui supportent X, malgré les valeurs manquantes de la région B. Ce groupe est noté $mv(r)_X$;

Régions C et D : les objets désactivés pour X. Ce groupe est noté $Des(X, mv(r))$ et le support de X est indécidable dans ce groupe ;

Régions E et F : les objets qui ne supportent pas X.

La notion de désactivation permet donc de capturer trois comportements des objets pour le support d'un motif. La région C ($Des(X, mv(r_X))$) correspond précisément à la différence de support pour X entre la base incomplète et la base complète. La suite de cette présentation étudie cette différence :

Proposition 4 *Soit X un motif classique, r une base de données et mv un opérateur de modélisation de valeurs manquantes, alors*

$$Des(X, mv(r_X)) = r_X \setminus mv(r)_X.$$

Preuve :

- $\subseteq$: Soit $o \in Des(X, mv(r_X))$. Par définition de la désactivation, $o \in r_X$. En raisonnant par l'absurde, supposons que $o \in mv(r)_X$. Cela signifie que X est présent dans o, ou $\forall a \in X,\ mv(R)(a, o) = present$, ce qui contredit l'appartenance de o à $Des(X, mv(r_X))$.
- $\supseteq$: Soit $o \in r_X \setminus mv(r)_X$. En raisonnant par l'absurde, supposons maintenant que $\neg\exists a \in X$ *t.q.* $mv(R)(a, o) = manquant$. Donc, $\forall a \in X,\ mv(R)(a, o) = present$ ou bien $mv(R)(a, o) = absent$. Or les hypothèses stipulent que o n'est pas dans $mv(r)_X$, donc $\exists a \in X,\ mv(R)(a, o) \neq present$. Cet attribut n'est finalement pas présent : il est absent dans $mv(r)$, et a fortiori dans r. Cela contredit l'hypothèse selon laquelle o est dans r_X et $\exists a \in X$ *t.q.* $mv(R)(a, o) = manquant$. De plus, $o \in r_X$, ou $\forall a \in X,\ R(a, o) = present$ et d'après la définition 21, $\forall a \in X,\ mv(R)(a, o) \neq absent$: nous retrouvons la définition 22 d'un objet désactivé pour X dans r. Plus précisément, comme nous savons que $o \in r_X$, o est finalement un objet de r_X désactivé pour X, soit $o \in Des(X, mv(r_X))$.

□

La proposition suivante, relative aux fréquences, est immédiate :

Proposition 5 *Soit X un motif classique, r une base de données et mv un opérateur de modélisation de valeurs manquantes.*

$$|Des(X, mv(r_X))| = \mathcal{F}(X, r) - \mathcal{F}(X, mv(r)).$$

Détaillons ce principe sur notre exemple pour le motif a_2a_3 : $r_{a_2a_3} = \{o_2, o_5, o_6\}$ et sa fréquence vaut 3. Dans la base incomplète, sa fréquence n'est plus que 2 et $Des(a_2a_3, mv(r_{a_2a_3})) = \{o_6\}$: nous avons bien l'égalité de la proposition 5. Quand on ne connaît pas la base complète r,

on ne connaît pas non plus r_X, encore moins $|Des(X, mv(r_X))|$. Mais grâce à la proposition 3 nous pouvons borner cette quantité en considérant les objets désactivés dans $mv(r)$, car cette base contient plus d'objets que $mv(r_X)$. Sur notre exemple $Des(a_2a_3, mv(r)) = \{o_6, o_8\}$, à cause de la confusion induite dans l'objet o_8 par la valeur manquante qui touche a_3 et a_4. $\mathcal{F}(a_2a_3, r)$ est donc compris entre $\mathcal{F}(a_2a_3, mv(r))$ et $\mathcal{F}(a_2a_3, mv(r)) + |Des(a_2a_3, mv(r))|$, soit entre 2 et 4.

Si l'on ne dispose que des données incomplètes, le nombre précis d'objets contenant initialement X et désactivés pour ce motif (*i.e.* $Des(X, mv(r_X))$) est impossible à déterminer. En revanche, une borne supérieure $Des(X, mv(r))$ est disponible, calculable dans la base incomplète. Notre travail repose sur ce principe, que nous étendons maintenant au cas des motifs généralisés.

6.2.2 Désactivation d'un motif généralisé

Nous souhaitons étudier le comportement de la notion de k-liberté en présence de valeurs manquantes. Pour cela, nous avons besoin d'étendre la proposition 5 relative à l'écart de fréquence d'un motif classique, pour caractériser l'écart de fréquence d'un motif généralisé. À la section suivante, nous fournirons des bornes pour cet écart, qui seront calculées dans la base incomplète. La désactivation d'un motif généralisé $X\overline{Y}$ décrit cette notion, suivant le principe d'inclusion-exclusion utilisé à l'équation 3.1 page 42.

Définition 23 (Désactivation généralisée) *Soit $X\overline{Y}$ un motif généralisé, r une base de données et $mv()$ un opérateur de modélisation de valeurs manquantes. La désactivation de $X\overline{Y}$ est la quantité*

$$des(X\overline{Y}, mv(r_{X\overline{Y}})) = \sum_{\emptyset \subseteq J \subseteq Y} (-1)^{|J|} |Des(XJ, mv(r_{XJ}))|.$$

Notons qu'à la différence de la désactivation pour un motif classique, l'ensemble $Des(X\overline{Y}, mv(r_{X\overline{Y}}))$ n'est pas défini et c'est pourquoi nous notons la désactivation généralisée avec un « d » minuscule. En effet, la suite montrera que la somme alternée $des(X\overline{Y}, mv(r_{X\overline{Y}}))$ peut être négative. Cette quantité ne peut donc pas représenter le cardinal d'un ensemble. Toutefois, cette définition permet de qualifier le rapport de fréquence d'un motif généralisé entre la base complète et la base incomplète.

Proposition 6 *Soit $X\overline{Y}$ un motif généralisé, r une base de données et mv un opérateur de modélisation de valeurs manquantes.*

$$des(X\overline{Y}, mv(r_{X\overline{Y}})) = \mathcal{F}(X\overline{Y}, r) - \mathcal{F}(X\overline{Y}, mv(r)).$$

Preuve : La proposition 5 indique que $|Des(XJ, mv(r_{XJ}))| = \mathcal{F}(XJ, r) - \mathcal{F}(XJ, mv(r))$. Alors, en remplaçant dans la définition 23, on obtient $des(X\overline{Y}, mv(r_{X\overline{Y}})) = \sum_{\emptyset \subseteq J \subseteq Y} (-1)^{|J|} (\mathcal{F}(XJ, r) - \mathcal{F}(XJ, mv(r))) = \sum_{\emptyset \subseteq J \subseteq Y} (-1)^{|J|} \mathcal{F}(XJ, r) - \sum_{\emptyset \subseteq J \subseteq Y} (-1)^{|J|} \mathcal{F}(XJ, mv(r))$. En appliquant le principe d'inclusion-exclusion sur ces deux sommes, on obtient la différence entre les fréquences dans les bases complète et incomplète. □

Comme nous l'avons signalé, cet écart de fréquence peut être négatif. Lorsque l'association entre X et Y existe dans la base complète ($\mathcal{F}(X\overline{Y}, r)$ est alors nulle), la présence d'une valeur manquante peut la faire disparaître dans la base incomplète (alors $\mathcal{F}(X\overline{Y}, mv(r)) > 0$). Dans ce cas, la différence est négative. Pour notre exemple, l'association $a_7 \rightarrow a_4$ existe dans r. À cause de la valeur manquante dans l'objet o_8, elle n'est plus exacte dans $mv(r)$: $des(a_7\overline{a_4}) = 0 - 1 = -1$.

Nous examinons maintenant ce qu'il se produit lorsque l'on considère les objets désactivés pour une association $X \rightarrow \vee Y$:

Définition 24 (Désactivation d'un objet pour une association) *Soient X et Y deux motifs classiques, r une base de données et $mv()$ un opérateur de modélisation de valeurs manquantes. Le nombre d'objets désactivés pour une association $X \rightarrow \vee Y$ est*

$$|Des(X \rightarrow \vee Y, mv(r_{X \rightarrow \vee Y}))| = |Des(X, mv(r_X))| - des(X\overline{Y}, mv(r_{X\overline{Y}})).$$

Nous retrouvons alors pour une association le comportement de la désactivation, analogue à celui déjà mis en évidence par les propositions 5 et 6 :

Proposition 7 *Soient X et Y deux motifs classiques, r une base de données et mv un opérateur de modélisation de valeurs manquantes.*

$$|Des(X \rightarrow \vee Y, mv(r_{X \rightarrow \vee Y}))| = \mathcal{F}(X \rightarrow \vee Y, r) - \mathcal{F}(X \rightarrow \vee Y, mv(r)).$$

Preuve : Selon la proposition 5, $Des(X, mv(r_X)) = \mathcal{F}(X, r) - \mathcal{F}(X, mv(r))$. La proposition 6 décompose $des(X\overline{Y}, mv(r_{X\overline{Y}}))$ en $\mathcal{F}(X\overline{Y}, r)$ et $\mathcal{F}(X\overline{Y}, mv(r))$. Nous avons donc $|Des(X \rightarrow \vee Y, mv(r_{X \rightarrow \vee Y}))| = \mathcal{F}(X, r) - \mathcal{F}(X, mv(r)) - (\mathcal{F}(X\overline{Y}, r) - \mathcal{F}(X\overline{Y}, mv(r)))$. Nous utilisons alors la proposition 1 page 44 sous sa forme $\mathcal{F}(X \rightarrow \vee Y) = \mathcal{F}(X) - \mathcal{F}(X\overline{Y})$ pour regrouper et faire apparaître la propriété. □

Nous introduisons enfin la notion de désactivation pour un motif *dans son intégralité*. Cette définition est utile car elle permet de simplifier l'écriture de la désactivation pour une association :

Définition 25 (Désactivation intégrale) *Soient X et Y deux motifs classiques. Les objets contenant X et dont chaque attribut de Y est manquant sont les objets contenant la prémisse mais désactivés pour l'association correspondante. Pour cela, chaque attribut de Y doit être manquant et nous utilisons le signe $\wedge$ devant le nom du motif.*

$$Des(\wedge Y, mv(r)_X) = Des(X \rightarrow \vee Y, mv(r)).$$

Lorsque tous les attributs de Y sont manquants, il est impossible de décider, pour les objets contenant la prémisse, s'ils contiennent ou pas un attribut de la conclusion.

À l'aide de cet ensemble de définitions et de propriétés, nous pouvons désormais appréhender la liberté généralisée d'un motif dans une base, qualifiée par la nullité de $\mathcal{F}(X\overline{Y})$. Pour cela, nous bornons l'écart de cette fréquence entre les deux bases, en utilisant notre notion de désactivation.

6.3 Consistance de la k-liberté dans les bases incomplètes

Cette section montre comment effectuer le calcul des motifs k-libres dans une base incomplète, pour obtenir des motif k-libres de la base complète. Nous commençons par montrer qu'il est possible d'obtenir des bornes pour la fréquence d'un motif généralisé. Ces bornes établissent de nouvelles conditions de k-liberté en présence de valeurs manquantes, qui sont consistantes avec la k-liberté dans la base complète.

6.3.1 Nouvelles bornes pour la fréquence de $X\overline{Y}$

La désactivation et la fréquence de $X\overline{Y}$ nous intéressent particulièrement car elles déterminent la k-liberté. Rappelons que cette désactivation peut être négative. Nous en donnons donc une borne inférieure négative, faisant appel à la notion de désactivation intégrale, et une borne supérieure positive :

Proposition 8 *Soit $X\overline{Y}$ un motif généralisé, r une base de données et $mv()$ un opérateur de modélisation de valeurs manquantes.*

$$-|Des(\wedge Y, (mv(r))_X)| \leq des(X\overline{Y}, mv(r_{X\overline{Y}})) \leq |Des(X, mv(r))|.$$

Preuve : D'après la définition 24, $des(X\overline{Y}, mv(r_{X\overline{Y}})) = |Des(X, mv(r_X))| - |Des(X \rightarrow \vee Y, mv(r_{X\rightarrow\vee Y}))|$. Il s'agit de la différence entre deux quantités positives. La première peut être bornée en généralisant la base de référence : $|Des(X, mv(r_X))| \leq |Des(X, mv(r))|$. Pour la seconde, nous utilisons la définition 25 pour écrire $|Des(X \rightarrow \vee Y, mv(r_{X\rightarrow\vee Y}))| = Des(\wedge Y, mv(r_{X\rightarrow\vee Y})_X)$. Nous bornons alors cette quantité en généralisant la base de référence et obtenons la deuxième borne : $Des(\wedge Y, mv(r)_X)$. □

Cette propriété est particulièrement intéressante, car elle permet de borner une quantité inconnue, $des(X\overline{Y}, mv(r_{X\overline{Y}}))$, relative à la base complète, par deux valeurs calculables dans la base incomplète. En effet, il est possible de connaître $-|Des(\wedge Y, (mv(r))_X)|$ en examinant $mv(r)_X$, les objets de $mv(r)$ contenant X et dont tous les attributs de Y sont manquants et $|Des(X, mv(r))|$ en examinant les objets de $mv(r)$ désactivés pour X.

La fréquence réelle de $X\overline{Y}$ dans r peut finalement être bornée par des quantités calculées dans $mv(r)$:

Théorème 1 *Soit $X\overline{Y}$ un motif généralisé, r une base de données et mv un opérateur de modélisation de valeurs manquantes.*

$$\mathcal{F}(X\overline{Y}, mv(r)) - |Des(\wedge Y, (mv(r))_X)| \leq \mathcal{F}(X\overline{Y}, r) \leq \mathcal{F}(X\overline{Y}, mv(r)) + |Des(X, mv(r))|.$$

Preuve : Il s'agit d'une combinaison des propositions 6 et 8. □

6.3.2 k-liberté dans les bases incomplètes

La k-liberté de $Z = XY$ est liée à la valeur de $X\overline{Y}$. La désactivation permet de *borner* cette notion dans un contexte incomplet et nous définissons pour cela la *mv-k-liberté* dans la base incomplète. Cette définition permet le résultat de consistance énoncé au théorème 2 :

Définition 26 (mv-k-liberté)

- *Un motif Z est mv-k-libre dans $mv(r)$ si et seulement si*

$$\forall XY = Z,\ |Y| \leq k,\ \mathcal{F}(X\overline{Y}, mv(r)) - |Des(\wedge Y, (mv(r))_X)| > 0. \tag{6.1}$$

- *Un motif Z est* nonmv-k-libre *dans $mv(r)$ si et seulement si*

$$\exists XY = Z,\ |Y| \leq k,\ \mathcal{F}(X\overline{Y}, mv(r)) + |Des(X, mv(r))| = 0. \tag{6.2}$$

Notons tout d'abord que dans le cas d'une base complète, la notion de mv-k-liberté est **compatible** avec la k-liberté. En effet, les ensembles d'objets désactivés sont vides lorsqu'il n'y a pas de valeurs manquantes. Utilisées dans une base de données complètes, nos définitions sont équivalentes à celle de la k-liberté. C'est un point important pour développer des algorithmes qui travaillent indifféremment sur les contextes complets ou incomplets.

Les deux notions de mv-k-liberté et de nonmv-k-liberté sont introduites indépendamment l'une de l'autre. La suite justifiera cette distinction, car ces deux définitions ne sont pas contraires à cause des valeurs manquantes.

Ces définitions permettent de relier la mv-k-liberté dans la base incomplète à la k-liberté dans la base complète grâce à l'importante propriété suivante :

Théorème 2 (Consistance de la (non)mv-k-liberté) *Soit X un motif, r une base de données complète et mv un opérateur de modélisation de valeurs manquantes :*

- *X est mv-k-libre dans $mv(r) \Longrightarrow X$ est k-libre dans r*
- *X est nonmv-k-libre dans $mv(r) \Longrightarrow X$ n'est pas k-libre dans r*

Preuve : Le théorème 1 stipule que $\mathcal{F}(X\overline{Y}, r)$ est borné par $\mathcal{F}(X\overline{Y}, mv(r)) - |Des(\wedge Y, (mv(r))_X)|$ et $\mathcal{F}(X\overline{Y}, mv(r)) + |Des(X, mv(r))|$. Si la borne inférieure est strictement positive (équation 6.1 de la définition 26 de la mv-k-liberté dans $mv(r)$), $\mathcal{F}(X\overline{Y}, r)$ est alors strictement positive donc non nulle et le motif est libre dans r. De même, si la borne supérieure de $\mathcal{F}(X\overline{Y}, r)$ est nulle (équation 6.2 de la définition 26 de la nonmv-k-liberté dans $mv(r)$), alors $\mathcal{F}(X\overline{Y}, r)$ est nulle et le motif n'est pas libre dans r. □

Dans [Rioult et Crémilleux, 2003a, Rioult et Crémilleux, 2004], cette consistance est montrée dans le cas particulier où $k = 1$, à l'aide de considérations sur la constitution des fermetures au sens de Galois.

Ces définitions de la mv-k-liberté et de la nonmv-k-liberté permettent de déterminer des propriétés de la base complète à partir de sa transformation en base incomplète. En cela, nous parlons de **consistance** de nos modes de calcul. Dans cette situation, la complétude consisterait à fournir la réciproque à cette propriété, c'est-à-dire à fournir *tous* les motifs k-libres. Dans ce mémoire, cela n'est pas le cas et nous montrons seulement un résultat de consistance. Cependant, ce résultat est important : notre méthode retrouve une partie des motifs k-libres de la base complète, et tous les motifs découverts ont cette propriété dans la base complète. Il assure que les motifs obtenus selon ces définitions sont valides en présence de valeurs manquantes et que l'algorithme ne génère pas de pollution.

Avec les nouvelles conditions de mv-k-liberté (ou de nonmv-k-liberté), la consistance peut s'interpréter comme suit :

mv-k-liberté : même si l'on transforme les valeurs manquantes pour Y en valeurs présentes (on soustrait la désactivation intégrale de Y du nombre d'exceptions à l'association), cela ne suffit toujours pas à créer l'association ;

nonmv-k-liberté : pour certifier en Z la validité d'une association $X \rightarrow \vee Y$, nous requérons qu'il n'y ait aucune valeur manquante sur X et que nous constatons la validité de l'association dans la base incomplète.

6.3.3 Propriétés de la (non)mv-k-liberté

Les définitions de la mv-k-liberté et de la nonmv-k-liberté ne sont pas complémentaires et les deux notations sont justifiées : un motif non mv-k-libre n'est pas nécessairement nonmv-k-libre. Pour préciser les rapports entre ces deux notions, nous énonçons le corollaire suivant :

Corollaire 2 *Pour un motif X, les deux implications équivalentes suivantes sont vraies :*

- *X est mv-k-libre $\Longrightarrow$ X n'est pas nonmv-k-libre ;*
- *X est nonmv-k-libre $\Longrightarrow$ X n'est pas mv-k-libre.*

Preuve : Il s'agit d'un corollaire du théorème 2, car n'être pas k-libre dans une base complète est bien le contraire d'être k-libre, donc les deux implications peuvent être combinées pour donner le corollaire. □

Les notions de mv-k-liberté et de nonmv-k-liberté ne sont pas contraires l'une de l'autre et certains motifs ne seront ni mv-k-libres, ni nonmv-k-libres, lorsque qu'il est impossible de décider à cause des valeurs manquantes, s'ils sont présents ou pas dans un objet. D'une certaine façon, nous avons transféré la notion d'inconnue sur le support, à celle d'inconnue de la k-liberté. Par exemple, la table 6.1 détaille le calcul de la liberté pour le motif a_4a_7. Il n'est ni mv-k-libre, ni nonmv-k-libre.

X	Y	$\mathcal{F}(X\overline{Y}, mv(r))$	$\lvert Des(\wedge Y, mv(r)_X)\rvert$	$\lvert Des(X, mv(r))\rvert$	1-libre ?	non-1-libre ?
a_4	a_7	1	1	1	$1-1 \not> 0$: non	$1+1 \neq 0$: non
a_7	a_4	1	1	1	$1-1 \not> 0$: non	$1+1 \neq 0$: non

TAB. 6.1 – Décision sur la liberté de $Z = a_4a_7$.

Nous donnons maintenant une propriété dont l'étude est capitale pour le développement d'algorithmes d'extraction des motifs mv-k-libres. Elle concerne l'(anti)monotonie des nouvelles définitions pour la k-liberté. *A priori*, la mv-k-liberté n'a pas de bonne propriété. Ce n'est pas le cas pour la `non`mv-k-liberté :

Proposition 9 (Monotonie de la `non`mv-k-liberté) *La propriété de `non`mv-k-liberté est monotone, i.e. pour tout motif Z*

$$Z \subseteq Z' \Rightarrow (\texttt{non}\text{mv}-k-\text{libre}(Z) \Rightarrow \texttt{non}\text{mv}-k-\text{libre}(Z'))$$

Preuve : Soit Z un motif `non`mv-k-libre. Alors $\exists XY = Z$, $\mathcal{F}(X\overline{Y}, mv(r)) + \lvert Des(X, mv(r))\rvert = 0$ soit $\mathcal{F}(X\overline{Y}, mv(r)) = 0$ et $\lvert Des(X, mv(r))\rvert = 0$. $\mathcal{F}(X\overline{Y}, mv(r)) = 0$ signifie que pour tout objet $o \in \mathcal{O}$, $X \subseteq o \Rightarrow Y \cap o \neq \emptyset$. *A fortiori*, $X \subseteq o \Rightarrow aY \cap o \neq \emptyset$ pour tout $a \in \mathcal{A}$, donc $\mathcal{F}(X\overline{aY}, mv(r)) = 0$. Par induction sur tous les attributs de $Z' \backslash Z$, on déduit que Z' est également `non`mv-k-libre. □

Cette propriété justifie l'adaptation d'un classique algorithme d'extraction sous contrainte antimonotone, avec la négation de la contrainte de `non`mv-k-liberté. Baptisé `MV-k-miner`, cet algorithme extrait par niveaux des motifs qui ne sont pas `non`mv-k-libres mais ne fournit à l'utilisateur que les motifs mv-k-libres. Le corollaire 2 indique qu'un motif `non`mv-k-libre n'est pas mv-k-libre ; cette propriété est monotone, donc l'algorithme d'extraction doit élaguer ces motifs, car aucun motif plus spécifique ne peut être mv-k-libre. La négation de la `non`mv-k-liberté sert donc de critère d'élagage lors la production des candidats et la mv-k-liberté sert à écarter les candidats invalides d'après les informations disponibles dans la base. `MV-k-miner` est détaillé à la section 6.4.

6.3.4 Exemple

Le tableau 6.2 étudie les motifs 2-libres présents dans notre base de données d'exemple. Les trois colonnes concernent respectivement la base complète r, la base incomplète $mv(r)$ et la base où les valeurs manquantes ont été ignorées et considérées absentes, notée $mv(r)^\circ$. Bien évidemment, à cause du résultat de consistance, on constate que tous les k-libres calculés dans $mv(r)$ le sont également dans la base complète. En revanche, dans la base $mv(r)^\circ$ dont les valeurs manquantes ont été ignorées, l'extraction indique que le motif a_4a_7 est libre, ce qui est faux dans la base d'origine.

r	$mv(r)$	$mv(r)^\circ$
a_1a_3	a_1a_3	a_1a_3
a_1a_4		
a_1a_5		a_1a_5
a_1a_6	a_1a_6	a_1a_6
a_1a_7		a_1a_7
a_2a_3		a_2a_3
a_2a_4		
a_2a_5		a_2a_5
a_2a_6	a_2a_6	a_2a_6
a_2a_7		a_2a_7
a_3a_6		a_3a_6
a_4a_6		a_4a_6
		a_4a_7

TAB. 6.2 – Comparaison des motifs 2-libres obtenus en présence de valeurs manquantes (les singletons ne sont pas indiqués).

6.4 Algorithme de calcul des motifs k-libres dans les bases incomplètes

Nous détaillons ici le fonctionnement de l'algorithme `MV-k-miner` d'extraction des motifs mv-k-libres en présence de valeurs manquantes. Il consiste en une adaptation de l'algorithme `k-miner` (algorithmes 2 page 47 et 3 page 48), qui effectue cette tâche sur des bases complètes. Pour traiter les bases incomplètes, `MV-k-miner` nécessite quelques modifications par rapport à `k-miner`. Tout d'abord, il doit indiquer distinctement les motifs mv-k-libres et `non`mv-k-libres. Selon la propriété de monotonie de la `non`mv-k-liberté (proposition 9), les `non`mv-k-libres indiquent les motifs dont aucune spécialisation n'est mv-k-libre. L'algorithme doit donc générer des motifs par niveaux tant qu'ils ne sont pas `non`mv-k-libres et ne conserver que les motifs mv-k-libres.

Deux séries de bornes ($-|Des(\wedge Y, mv(r_X))|$ et $|Des(X, mv(r))|$) doivent être calculées préventivement pendant la génération, mais seule celle qui disqualifie les `non`mv-k-libres est conservée pour disqualifier définitivement les candidats après calcul de leur fréquence dans la base. Ce procédé est décrit par les algorithmes 4 et 5.

Données : une base incomplète $mv(r)$, une fréquence minimum γ et k une profondeur de règles
Résultat : l'ensemble $\mathcal{S}$ des motifs vérifiant mv-k-libre et l'ensemble $\mathcal{D}$ des motifs vérifiant `non`mv-k-libre ou non fréquent

$l = 1$;
initialiser $\mathcal{C}and_1$ avec la liste des singletons ;
répéter
  /*`calcul des disqualifieurs` */
  $\mathcal{D}_l = \{X \in \mathcal{C}and_l \ t.q. \ \texttt{non}mv{-}\text{k}{-}libre(X) \vee \neg frequent(X, mv(r))\}$;
  /*`ne pas conserver les disqualifieurs` */
  $\mathcal{S}_l = \mathcal{C}and_l \backslash \mathcal{D}_l$;
  générer les candidats dans $\mathcal{C}and_{l+1}$;
  l = l + 1 ;
jusqu'à $\mathcal{C}and_l = \emptyset$;
retourner $\bigcup_l \mathcal{S}_l$;

Algorithme 4 – `MV-k-miner-` extraction de motifs mv-k-libres.

Données : un ensemble $\mathcal{S}_l$ de motifs k-libres de longueur l
Résultat : l'ensemble $\mathcal{C}and_{l+1}$ des motifs candidats à la vérification de $\texttt{non}mv{-}\text{k}{-}libre$
pour *chaque candidat Z, généré par fusion de deux mv-k-libres ayant un préfixe commun de longueur $l-1$* **faire**
  début
    vérifier que ses sous-ensembles sont mv-k-libres ;
    /*`calculer les bornes de sa fréquence` */
    construire l'arbre des motifs X et leurs fréquences tels que $|Z \backslash X| \leq k$;
    pour chaque X de l'arbre, calculer la somme alternée des fréquences de tous ses sur-ensembles, qui constitue une version préliminaire $\sigma(X, Y)$ de la borne de $\mathcal{F}(Z)$;
    séparer $\sigma(X, Y)$ en $\sigma(X, Y) - |Des(\wedge Y, mv(r_X))|$ et $\sigma(X, Y) + |Des(X, mv(r))|$ pour tenir compte des valeurs manquantes ;
    mémoriser les bornes $\sigma(X, Y) + |Des(X, mv(r))|$ pour $\mathcal{F}(Z)$;
    en cas d'égalité des bornes, refuser le candidat ;
  fin
fin

Algorithme 5 – Génération des candidats de longueur $l + 1$.

Chapitre 7

Règles d'association généralisées informatives dans les bases incomplètes

Pour éviter certaines lourdeurs et lorsqu'il n'y a pas d'ambiguïté, le terme « règle informative » résume l'expression « règle d'association généralisée informative », et celui de « fermeture » est employé pour désigner la fermeture généralisée ou l'un de ses éléments.

Au chapitre 3, nous avons montré l'intérêt des règles d'association généralisées informatives. Rappelons qu'elles sont construites à partir des motifs k-libres et de leurs fermetures généralisées. Dans le chapitre précédent, nous avons proposé un calcul consistant de la k-liberté, qui permet dans une base incomplète d'exhiber des motifs qui ont cette propriété dans toute base complète correspondante. Ce chapitre s'intéresse au calcul de la fermeture généralisée des motifs mv-k-libres afin de pouvoir obtenir des règles informatives en présence de valeurs manquantes. Pour cela, nous proposons d'utiliser le concept d'opposition de base de données, une transformation qui laisse invariantes les valeurs manquantes. Les règles informatives issues de la base opposée ont un rapport direct avec celles extraites depuis la base originale et nous montrons que les règles communes entre les deux versions de la base s'avèrent particulièrement intéressantes pour être des règles informatives de la base complète.

Ce chapitre commence section 7.1 par introduire deux stratégies naturelles pour le calcul de fermetures généralisées dans les bases incomplètes. La notion de base opposée est expliquée à la section 7.2 et la section 7.2.2 montre que les règles exactes dans la base opposée correspondent à une inversion de celles de la base originale. Ce principe est utilisé à la section 7.3 pour définir les règles *correctes*, qui sont les règles informatives communes à la base incomplète et sa base opposée. Son intérêt expérimental est illustré par les expériences de la section 7.4.

7.1 Stratégies de calcul de fermeture généralisée

Nous examinons ici les modes possibles de calcul des fermetures généralisées. Pour un motif X k-libre, la proposition 2 page 46 indique qu'il faut déterminer les traverses minimales de longueur bornée par k qui intersectent avec chaque objet contenant X. Comment effectuer ce calcul en présence de valeurs manquantes ?

7.1.1 Définition des stratégies

Pour construire les traverses minimales, un attribut manquant peut être considéré selon ses deux valeurs théoriques possibles : une valeur absente, auquel cas il ne fait pas partie d'une traverse, ou comme une valeur présente et elle peut alors être intégrée à une traverse. Sur notre exemple table 5.1, examinons ce que pourrait être la fermeture généralisée du motif a_2 pour $k = 2$. Pour calculer cette fermeture, seul l'objet o_8 pose problème avec sa valeur manquante sur les attributs a_3 et a_4. Si l'on considère que a_3 est présent dans o_8, cet attribut peut être traversé et la fermeture généralisée contient le motif a_3. Si l'on considère qu'il est absent, il ne peut être traversé et la fermeture est a_3a_7, car seul a_7 permet de traverser o_8.

Il y a donc deux façons naturelles de procéder à la construction d'une fermeture généralisée suivant que l'on considère qu'une valeur manquante masque une valeur présente ou absente. Dans le premier cas, une stratégie nommée S_1 effectuerait un choix « optimiste », dans le second cas une stratégie nommée S_2 effectuerait un choix « pessimiste ». Rappelons que dans la théorie des bases de données, la stratégie S_2 est appelée *hypothèse du monde fermé* (closed world assumption) [Bosc *et al.*, 2002]. Les définitions formelles pour S_1 et S_2 sont précédées par les définitions des objets qu'elles considèrent :

Définition 27 (Objet S_1) *Pour o un objet de $r = (\mathcal{A}, \mathcal{O}, R)$, l'objet $S_1(o)$ utilisé par la stratégie S_1 est défini comme suit ($\forall a \in \mathcal{A}$) :*

$$R(a, S_1(o)) = \begin{cases} present & si\ R(a,o) = present \vee R(a,o) = manquant \\ absent & si\ R(a,o) = absent \end{cases}$$

Définition 28 (Objet S_2) *Pour o un objet de $r = (\mathcal{A}, \mathcal{O}, R)$, l'objet $S_2(o)$ utilisé par la stratégie S_2 est défini comme suit ($\forall a \in \mathcal{A}$) :*

$$R(a, S_2(o)) = \begin{cases} present & si\ R(a,o) = present \\ absent & si\ R(a,o) = absent \vee R(a,o) = manquant \end{cases}$$

Définition 29 (Stratégie S de calcul de fermeture généralisée) *À l'aide de la stratégie S ($S \in \{S_1, S_2\}$), la fermeture généralisée d'un motif X dans une base incomplète r est*

$$\mathcal{FG}_k(X, mv(r), S) = Tr_k(\{S(o \backslash X) \mid o \in \mathcal{O}\ et\ X \subseteq o\})$$

7.1.2 Propriétés des fermetures généralisées calculées dans les bases incomplètes

Le choix de l'une ou l'autre stratégie a un impact fort sur la validité des fermetures calculées en présence de valeurs manquantes, relativement aux fermetures qui seraient obtenues dans la base complète. Plus précisément, le théorème suivant montre que la stratégie S_1 détermine des fermetures dont les motifs sont inclus dans l'un de ceux de la fermeture généralisée calculée dans r.

Théorème 3 *Soient r une base complète et X un motif mv-k-libre. La stratégie S_1 calcule pour X dans $mv(r)$ une fermeture qui « minore » la fermeture réelle Y.*

$$\forall Y' \in \mathcal{FG}_k(X, mv(r), S_1)\ \exists Y \in \mathcal{FG}_k(X, r) \mid Y' \subseteq Y$$

Preuve : Lorsque les valeurs manquantes affectent les attributs présents avec un motif X, la stratégie S_1 considère qu'il y a plus d'attributs présents dans les objets de la base incomplète que dans ceux de la base complète. Les traverses calculées seront donc plus courtes qu'elles ne le seraient sans valeurs manquantes. Si un objet incomplet contient X, il n'est pas utilisé pour le calcul des traverses. Dans ce cas également, les traverses seront plus courtes. □

Revenons à notre exemple : la stratégie S_1 indique que la fermeture de a_2 est a_3. Dans la base complète correspondante, les fermetures possibles sont $a_3 \vee a_4$ ou $a_3 \vee a_7$. S_1 fournit bien un « minorant » de la fermeture réelle.

La propriété duale (*i.e.* « majoration » de la fermeture calculée dans r) est fausse pour la stratégie S_2. Pourtant dans la pratique nous avons constaté que la stratégie S_2 calcule des fermetures majorant la fermeture obtenue dans r, comme l'illustre notre exemple (cf. table 7.1). Mais des contre-exemples se produisent quand les valeurs manquantes apparaissent sur les attributs du motif libre. Ainsi, pour notre exemple, la règle $a_4 \rightarrow a_1 \vee a_2$ est exacte dans r (en gras à la table 7.1). La valeur manquante sur l'objet o_8 désactive la présence de a_4 et cet objet est ignoré car il ne contient pas la prémisse. Lorsque l'on calcule la fermeture de a_4, on constate que a_1 est systématiquement présent, valeur manquante ou pas. Le choix de la stratégie importe donc peu : la seule fermeture proposée sera a_1, qui minore la fermeture originale a_1a_2.

La table 7.1 montre les fermetures trouvées lors de l'extraction des 2-libres, dans la base complète r et la base incomplète $mv(r)$, en utilisant l'une ou l'autre de ces stratégies. Ce tableau inclut l'exemple précédent (mis en gras) montrant que S_2 ne majore pas systématiquement la fermeture dans r. Conformément au théorème 3, la stratégie S_1 fournit bien une borne inférieure de cette fermeture.

$mv(r)$ avec stratégie S_1	r	$mv(r)$ avec stratégie S_2
	$1 \rightarrow 3 \vee 4$	$1 \rightarrow 3 \vee 4$
$1 \rightarrow 4 \vee 5$	$1 \rightarrow 4 \vee 5$	
$2 \rightarrow 3$	$2 \rightarrow 3 \vee 4$	
$''$	$2 \rightarrow 3 \vee 7$	$2 \rightarrow 3 \vee 7$
$3 \rightarrow 2 \vee 5$	$3 \rightarrow 2 \vee 5$	
$3 \rightarrow 1 \vee 2$	$3 \rightarrow 1 \vee 2$	
$3 \rightarrow 5 \vee 6$	$3 \rightarrow 5 \vee 6$	
$4 \rightarrow 6 \vee 7$	$4 \rightarrow 6 \vee 7$	$4 \rightarrow 6 \vee 7$
$\mathbf{4 \rightarrow 1}$	$\mathbf{4 \rightarrow 1 \vee 2}$	$\mathbf{4 \rightarrow 1}$
$''$	$\mathbf{4 \rightarrow 1 \vee 7}$	$''$
$5 \rightarrow 1 \vee 2$	$5 \rightarrow 1 \vee 2$	$5 \rightarrow 1 \vee 2$
$5 \rightarrow 3$	$5 \rightarrow 3$	$5 \rightarrow 3$
$6 \rightarrow 1 \vee 2$	$6 \rightarrow 1 \vee 2$	
$6 \rightarrow 2 \vee 4$	$6 \rightarrow 2 \vee 4$	
$6 \rightarrow 1 \vee 3$	$6 \rightarrow 1 \vee 3$	$6 \rightarrow 1 \vee 3$
$6 \rightarrow 3 \vee 4$	$6 \rightarrow 3 \vee 4$	$6 \rightarrow 3 \vee 4$
$7 \rightarrow 1 \vee 2$	$7 \rightarrow 1 \vee 2$	$7 \rightarrow 1 \vee 2$
$7 \rightarrow 4$	$7 \rightarrow 4$	$7 \rightarrow 2 \vee 4$
$1\ 3 \rightarrow 5$	$1\ 3 \rightarrow 5$	
$2\ 6 \rightarrow 3$	$2\ 6 \rightarrow 3$	$2\ 6 \rightarrow 3$

TAB. 7.1 – Stratégies de calcul des fermetures pour $k = 2$ (pour une meilleure lisibilité, les attributs sont résumés par leur numéro).

7.1.3 Discussion

L'absence de dualité du théorème 3 implique que lorsque les calculs menés selon les deux stratégies concordent, cela ne suffit pas pour conclure sur la fermeture dans la base complète. Ce phénomène est conforme au manque de réciprocité du théorème 2 page 88, qui interdit toute complétude pour le calcul de la liberté. Ce manque de complétude indique que nous ne pouvons pas décider pour certains motifs s'ils sont k-libres ou pas. Si nous avions cette possibilité, nous serions capables de mieux calculer leurs fermetures.

Dans le meilleur des cas, nous retrouverons donc des motifs k-libres de la base complète mais nous ne pourrons jamais que proposer une borne inférieure pour leur fermeture. Comme l'exemple de la table 7.1 l'a montré, il est illusoire de penser que la combinaison des stratégies amène un surcroît d'information sur la fermeture, en particulier il est impossible de prédire cette fermeture avec exactitude. Les expériences décrites à la section 7.4 évaluent l'impact expérimental du choix de l'une ou l'autre stratégie.

	$mv(r)$							$^{-}mv(r)$						
objet	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_1	×		×		×				×		×		×	×
o_2		×	×		×			×			×		×	×
o_3	×		×		?	?	?		×		×	?	?	?
o_4	×			×		×			×	×		×		×
o_5		×	×			×		×			×	×		×
o_6	?	?	×			×		?	?		×	×		×
o_7	×			×			×		×	×		×	×	
o_8		×	?	?			×	×		?	?	×	×	

TAB. 7.2 – Base d'exemple et sa base opposée.

7.2 Règles informatives communes à une base et son opposée

Nous introduisons maintenant la notion de base opposée, utile pour la recherche de règles informatives, en permettant de lier les règles d'une base et de son opposée.

7.2.1 Base opposée

La définition formelle de la base opposée est la suivante :

Définition 30 (Base opposée) *Soit $r = (\mathcal{A}, \mathcal{O}, R)$ un contexte booléen. Nous notons ^{-}r la base opposée de r, qui admet pour relation ^{-}R, définie comme suit (pour chaque $a \in \mathcal{A}$ et chaque $o \in \mathcal{O}$) :*

$$^{-}R(a,o) = \begin{cases} absent & si\ R(a,o) = present \\ present & si\ R(a,o) = absent \\ manquant & si\ R(a,o) = manquant \end{cases}$$

Nous notons également ^{-}o pour désigner l'objet correspondant à o de r dans ^{-}r. Si les objets sont vus comme des ensembles d'attributs booléens, alors dans le cas d'une base complète, o et ^{-}o sont complémentaires.

La table 7.2 indique la base opposée pour notre exemple de base de données incomplète. On remarque immédiatement qu'outre l'opposition entre la présence et l'absence, le caractère *manquant* de la relation est laissé invariant lors de la transformation : les valeurs manquantes sont les points fixes de l'opposition de bases de données.

7.2.2 Inversion de règles d'association généralisées

La notion de base opposée permet de mettre en évidence une relation simple entre les règles d'associations généralisées extraites dans une base et sa base opposée. Le théorème suivant montre que la prémisse et la conclusion sont inversées suite à l'opposition du contexte.

Théorème 4 (Inversion de règle d'association généralisée) *Soit une règle d'association généralisée exacte $X \rightarrow \vee Y$ dans une base de données complète r. Si Y est présent dans ^{-}r, alors $Y \rightarrow \vee X$ est une règle d'association exacte dans ^{-}r, i.e.*

$$\models_r X \rightarrow \vee Y \iff \models_{^{-}r} Y \rightarrow \vee X.$$

Preuve : Nous montrons cette équivalence en prouvant celle de la nullité des fréquences $X\overline{Y}$ et $Y\overline{X}$. Soit $o \in \mathcal{O} \mid X \subseteq o \Rightarrow Y \cap o \neq \emptyset$. Nous avons l'équivalence entre les implications suivantes :

$$X \subseteq o \Rightarrow Y \cap o \neq \emptyset$$

$$X \cap {}^{-}o = \emptyset \Rightarrow Y \cap o \neq \emptyset \text{ car } o \text{ et } {}^{-}o \text{ sont complémentaires}$$

$$Y \cap o = \emptyset \Rightarrow X \cap {}^{-}o \neq \emptyset \text{ par contraposition}$$

$$Y \subseteq {}^{-}o \Rightarrow X \cap {}^{-}o \neq \emptyset$$

Ainsi, pour tout objet o contenant $X\overline{Y}$, l'objet ^{-}o contient $Y\overline{X}$, donc leurs fréquences coïncident. □

Ce théorème est bien connu en logique. Les règles d'association généralisées caractérisent les modèles de formules qui satisfont l'univers représenté par la base de données. Lorsque l'univers est inversé, les formules le sont également.

Regardons plus précisément les conditions d'application du théorème. Deux situations sont possibles pour obtenir dans ^{-}r la nullité de la fréquence de $Y\overline{X}$:

1. le cas où tout objet contenant Y contient un attribut de X. C'est le cas qui nous intéresse dans la pratique car il traduit une corrélation entre Y et X ;
2. celui où aucun objet de ^{-}r ne contient Y. En effet, l'inversion induit une perte locale des attributs qui sont communs à Y.

La condition $\mathcal{F}(Y, {}^{-}r) \neq 0$ est nécessaire. Sinon, l'association ne peut pas être « conservée » par l'opposition (*i.e.* sa forme inversée n'est pas exacte dans la base opposée, car la fréquence de la prémisse est nulle), comme le montre le tableau 7.3. Il présente un exemple simple de base de données, sous forme transactionnelle, ainsi que son opposée. Pour r, a_1 est 1-libre car il n'est pas présent dans chaque objet. Sa fermeture est $\{a_2\}$ et donne lieu à la règle $a_1 \rightarrow a_2$. Cependant, comme a_2 disparaît lors de l'opposition car il est présent dans chaque objet de r, il ne peut pas y avoir de règle d'association exacte dans ^{-}r contenant a_2 en prémisse.

r	^{-}r
a_1 a_2	a_3
a_2 a_3	a_1

TAB. 7.3 – Exemple pédagogique de base de données et de son opposée.

Le théorème d'inversion des règles ne permet donc pas de retrouver toutes les associations d'une base dans son opposée. Cependant, il propose dans le cas des bases incomplètes des règles d'association généralisées potentiellement intéressantes. En effet, une règle d'association $X \rightarrow \vee Y$ exacte dans $mv(r)$ a de bonnes chances de figurer inversée dans $^{-}mv(r)$. Nous préciserons à la section 7.4 les aspects quantitatifs de ce phénomène.

Un algorithme calculant les règles $Y \rightarrow \vee X$ dans ^{-}r doit donc agir avec des précautions particulières s'il souhaite comparer ces règles avec celles de r. Pour que ces règles soient exactes dans r, les motifs X de la fermeture généralisée de Y doivent être fréquents dans la base initiale. Cela correspond à une contrainte qui ne possède pas de bonne propriété pour utiliser les algorithmes classiques d'extraction sous contrainte antimonotone. Elle est complexe à gérer lors d'une extraction dans la base opposée, et elle n'est pas antimonotone. Pour utiliser malgré tout notre extracteur de motifs mv-k-libres, la contrainte de fréquence minimale est neutralisée (*i.e.* $\gamma = 1$). En contre-partie, la longueur des motifs k-libres est limitée à k, car des motifs plus longs n'auraient pas leur équivalent dans r sous la forme de fermeture généralisée, dont la longueur est limitée à k. La sélection des règles de ^{-}r pertinentes pour une comparaison avec les règles de r est effectuée en post-traitement, en calculant leur fréquence dans r.

7.3 Extraction de règles d'association généralisées informatives en présence de valeurs manquantes

Muni des résultats précédents, l'extraction de règles d'association généralisées informatives *a priori* intéressantes devient simple : il s'agit des règles communes aux règles informatives extraites dans $mv(r)$ et $^{-}mv(r)$. Précisons pourquoi ces règles sont *a priori* intéressantes.

Le théorème 2 page 88 de consistance garantit de calculer dans $mv(r)$ des prémisses k-libres dans r pour les règles d'association généralisées. Appliquée dans $^{-}mv(r)$, cette méthode fournit également des prémisses valides. Or, entre ces deux bases, les règles sont théoriquement inversées (théorème 4). Les règles communes sont donc particulièrement intéressantes, du fait de la fiabilité des motifs k-libres découverts qui les constituent.

Précisons que le théorème 4 ne porte pas sur la propriété « informative » des règles généralisées. En effet, si une règle $X \rightarrow \vee Y$ est informative dans r, $Y \rightarrow \vee X$ n'est pas nécessairement informative dans ^{-}r, ni inversement. Nous définissons donc précisément les règles d'association

généralisées *correctes* dans une base incomplète :

Définition 31 (Règle correcte dans une base incomplète) *Pour une stratégie S de calcul des fermetures généralisées, une règle d'association généralisée $X \rightarrow \vee Y$ est correcte si :*

1. *X est k-libre dans $mv(r)$ et $Y \in \mathcal{FG}_k(X, mv(r), S)$ ($X \rightarrow \vee Y$ est informative dans $mv(r)$ selon nos critères) ;*
2. *Y est k-libre dans $^{-}mv(r)$ et $X \in \mathcal{FG}_k(Y, ^{-}mv(r), S)$ ($Y \rightarrow \vee X$ est informative dans $^{-}mv(r)$ selon nos critères).*

Pour obtenir les règles correctes, `MV-k-miner` sélectionne les règles informatives obtenues dans $^{-}mv(r)$ qui correspondent à une règle informative de $mv(r)$.

La table 7.4 indique les règles correctes selon les quatre combinaisons des deux stratégies de calcul de fermeture. La section suivante effectue des expérimentations sur des bases de test. Remarquons que pour le nombre de règles correctes, la combinaison de deux stratégies identiques « encadre » les combinaisons hybrides. $S_2 - S_2$ fournit le moins de règles et pour les autres combinaisons, seules les règles de prémisses a_2 et a_4 s'avèrent incorrectes. Ce résultat était prévisible étant donné le théorème 3 sur la minoration de fermeture induite par la stratégie S_1.

		Base directe $mv(r)$	
		S_1	S_2
Base opposée $^{-}mv(r)$	S_1	$1 \rightarrow 4 \vee 5$	
		$2 \rightarrow 3$	
		$3 \rightarrow 2 \vee 5$	
		$3 \rightarrow 5 \vee 6$	
		$4 \rightarrow 1$	$4 \rightarrow 1$
		$4 \rightarrow 6 \vee 7$	$4 \rightarrow 6 \vee 7$
		$5 \rightarrow 3$	$5 \rightarrow 3$
		$6 \rightarrow 2 \vee 4$	
		$7 \rightarrow 4$	
		$1\ 3 \rightarrow 5$	
		$1\ 6 \rightarrow 4$	$1\ 6 \rightarrow 4$
	S_2	$2 \rightarrow 3$	
		$3 \rightarrow 2 \vee 5$	
		$4 \rightarrow 6 \vee 7$	$4 \rightarrow 6 \vee 7$
		$5 \rightarrow 3$	$5 \rightarrow 3$

TAB. 7.4 – Règles correctes dans $mv(r)$ (pour une meilleure lisibilité, les attributs sont résumés par leur numéro).

7.4 Expérimentations

Cette section présente des expériences menées sur des bases de données de test fournies par l'UCI [Blake et Merz, 1998]. Elles mesurent quantitativement l'intérêt des règles correctes, communes aux règles informatives d'une base incomplète et de sa version opposée.

7.4.1 Protocole

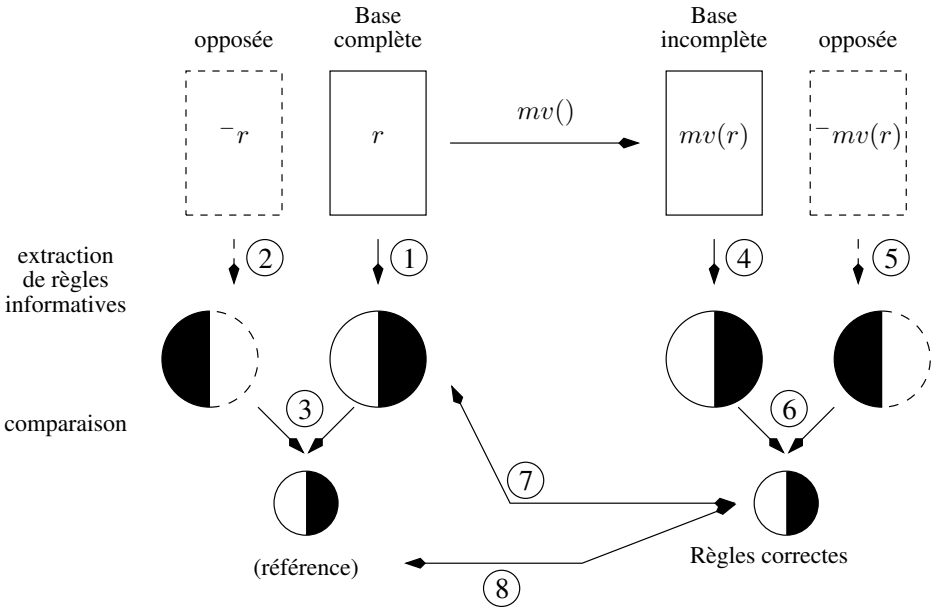

FIG. 7.1 – Protocole d'expériences pour les valeurs manquantes.

La figure 7.1 schématise le protocole défini : partant d'une base complète r, nous introduisons des valeurs manquantes suivant une loi uniforme pour obtenir une base incomplète $mv(r)$. Les extractions de règles informatives dans ces bases sont représentées par un rond bicolore qui schématise les inversions entre la prémisse et la conclusion. Les opérations effectuées sont :

- calcul des règles dans r (étape 1) ;
- calcul des règles dans ^{-}r (étape 2). La comparaison de ces deux calculs fournit une référence pour la quantité de règles communes entre r et ^{-}r (étape 3) ;
- calcul des règles dans $mv(r)$ (étape 4) ;
- calcul des règles dans $^{-}mv(r)$ (étape 5). La comparaison avec les règles issues de $mv(r)$ donne les règles correctes (étape 6).

Les étapes d'évaluation de la qualité des extractions menées dans les bases incomplètes sont réalisées à l'étape 7 et 8. Elles consistent à mesurer le nombre de règles correctes qui sont informatives dans r. L'étape 7 détermine cette **proportion** relativement au nombre de règles

correctes. L'étape 8 compare cette **quantité** par rapport au nombre de règles communes entre r et ^{-}r.

7.4.2 Résultats

Seuls les résultats pour la base `Glass` sont présentés ici. L'annexe 3 expose le bilan des expériences sur d'autres bases. Dans la base `Glass` [Blake et Merz, 1998] (2140 relations attribut/objet), l'extraction de référence trouve 30 108 règles informatives de profondeur 3 et de support minimum 1 % communes entre r et ^{-}r. Pour les deux combinaisons de stratégies $S_1 - S_1$ et $S_2 - S_2$ de calcul de fermeture, la figure 7.2 indique la proportion de règles correctes de mv(`Glass`) qui sont informatives dans r (étape 7).

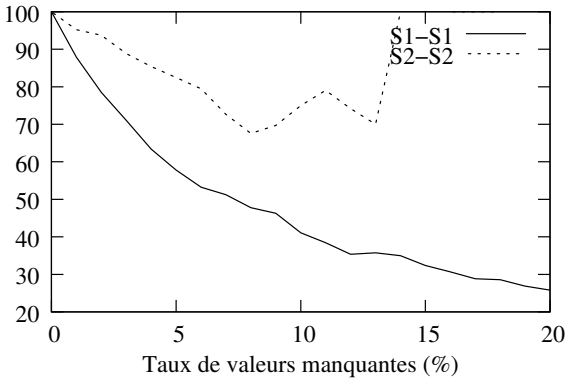

FIG. 7.2 – **Proportion** de règles correctes dans mv(`Glass`) qui sont informatives dans `Glass`.

La quantité de règles correctes dans mv(`Glass`) qui sont informatives dans `Glass` est représentée à la figure 7.3 (étape 8) par rapport au nombre de règles communes à r et ^{-}r. Nous utilisons cette référence car le nombre de règles correctes de la base incomplète est généralement très petit devant le nombre de règles informatives de la base complète. Au cours de nos expériences, le nombre de règles correctes n'a jamais dépassé cette référence.

Précisons que d'autres paramètres d'extraction produisent des effets similaires. Les comportements des combinaisons hybrides de stratégie ne sont pas reportés. Ils se situent à l'intermédiaire entre $S_1 - S_1$ et $S_2 - S_2$.

7.4.3 Discussion

Les différences entre les stratégies employées sont notables. La combinaison $S_1 - S_1$ retrouve une proportion moins importante de règles que la combinaison $S_2 - S_2$. Cependant, par rapport au nombre de règles communes entre r et ^{-}r, elles sont plus nombreuses qu'avec la combinaison $S_2 - S_2$. Les deux combinaisons suggèrent des compromis différents entre la quantité et la qualité des règles.

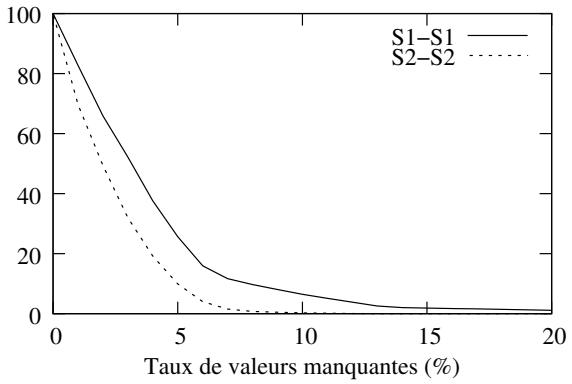

FIG. 7.3 – **Quantité** de règles correctes dans mv(`Glass`) qui sont informatives dans `Glass`.

De plus, pour la combinaison $S_2 - S_2$, la proportion importante de règles informatives dans r parmi les règles correctes de $mv(r)$ cache le faible nombre de règles correctes calculées. Certes, une proportion importante d'entre elles est informative dans la base complète, mais cette stratégie retrouve une faible quantité de règles.

Quelle stratégie choisir ? Si la base à traiter contient peu de valeurs manquantes (par exemple moins de 5 %), nous recommandons d'utiliser la combinaison $S_2 - S_2$. Le nombre d'associations informatives retrouvées est important parmi toutes les règles correctes et leur qualité est bonne. Sur une base avec un nombre plus conséquent de valeurs manquantes, la combinaison $S_1 - S_1$ permettra de découvrir plus de règles informatives, même si certaines sont entachées de doute.

7.5 Conclusion

Pour conclure, notre méthode permet de retrouver dans les bases incomplètes des règles d'associations généralisées, pour lesquelles la prémisse est un motif k-libre dans la base complète. Les conclusions de ces règles ont également cette propriété dans la base incomplète opposée. De plus, chaque conclusion calculée en présence de valeurs manquantes avec la stratégie S_1 est un sous ensemble de la conclusion qui serait calculée dans la base complète. Les expérimentations montrent l'intérêt pratique de notre méthode.

Conclusion

Dans cette partie, nous avons présenté les dommages causés par les valeurs manquantes sur les motifs k-libres, qui sont à la base des représentations condensées de motifs fréquents. Ce résultat, aussi illustré par des expérimentations, montre qu'il est impossible d'utiliser de façon fiable ces représentations dans les données incomplètes si les valeurs manquantes ne sont pas prises en compte.

Nous avons proposé l'utilisation d'une notion de mv-k-liberté pour les bases complètes, qui est consistante et compatible avec la k-liberté dans la base complète. Un résultat important est que notre approche assure la consistance des motifs k-libres découverts dans les contextes incomplets. Même sans connaître la base originale complète, nos modes de calculs assurent de ne retrouver que des motifs consistants avec les motifs de la base complète et n'introduisent pas de pollution dans les motifs découverts. Des expérimentations sur des bases de test confirment la pertinence de notre approche.

Ce résultat peut s'interpréter comme la recherche de motifs valides dans toutes les bases complètes possibles qui génèrent une base incomplète donnée. Comme le nombre de ces bases croît exponentiellement en fonction du nombre de valeurs manquantes, la quantité de motifs valides pour toutes ces bases est faible et peu de motifs sont retrouvés.

Cependant, au delà du résultat formel de cette méthode et de l'apport des motifs retrouvés, nous avons aussi proposé une solution originale, fondée sur l'utilisation de la base opposée, pour produire des règles d'association généralisées correctes en présence de valeurs manquantes. Leurs conclusions, issues des fermetures généralisées de motifs k-libres dans la base complète, possèdent la propriété de minorer les fermetures calculées dans la base complète.

Nous pensons que ce travail constitue une étape importante vers l'identification de propriétés intéressantes dans les bases de données incomplètes. Grâce à une meilleure maîtrise des effets des valeurs manquantes et des mécanismes qui assurent de retrouver des motifs valides, notre approche ouvre de nouvelles perspectives pour les méthodes d'exploration de données fondées sur les associations généralisées issues de motifs k-libres. Le chapitre 10 revient sur ce point et le chapitre 12 décrit une application.

Troisième partie

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

Introduction

Si l'extraction de motifs fréquents est aujourd'hui une tâche bien maîtrisée dans les contextes de dimension classique, il n'en va pas de même dans certaines situations où les données sont décrites par un grand nombre d'attributs. Typiquement, il s'agit de contextes provenant de la biologie, bien que ce problème de format affecte également les études médicales centrées sur un petit nombre de patients [Rioult *et al.*, 2005]. En bio-informatique, les objets représentent les expériences réalisées et les attributs les expressions de gènes. Le nombre de gènes utilisés par les biologistes est de l'ordre de plusieurs milliers, tandis que le nombre d'objets étudiés est faible (une centaine tout au plus), en raison des coûts de séquençage importants.

En moyenne comme dans le pire des cas, la complexité des algorithmes d'extraction de motifs est polynomiale en nombre d'objets (voir chapitre 4) mais exponentielle en nombre d'attributs. La configuration particulière des matrices d'expression de gènes semble donc rédhibitoire pour les méthodes usuelles à cause du grand nombre d'attributs. De plus, ces matrices sont généralement assez corrélées, ce qui n'est pas nécessairement le cas des matrices de transactions commerciales pour lesquelles les algorithmes classiques ont été mis au point.

Dans cette situation particulière, il est alors tentant d'appliquer les techniques de fouille de données à la matrice transposée des données. La nouvelle base comporte ainsi des dimensions compatibles avec une extraction de motifs sur de nombreuses lignes (les expressions de gènes) et peu de colonnes (les expériences). Hélas, les résultats obtenus sont relatifs à des motifs d'expériences, ce qui passionne peu les biologistes si ces motifs ne sont pas liés à des expressions de gènes. La difficulté du calcul est réduite, mais de nouveaux problèmes surgissent : interprétation du résultat, conversion des paramètres d'extraction, etc.

Quand le format est propice (peu d'objets, beaucoup d'attributs), la transposition de matrice permet de travailler sur des données dont l'exploration sera facilitée par un renversement des tendances lignes/colonnes. Cette partie de notre mémoire propose une nouvelle méthode d'extraction de motifs qui tire pleinement parti des caractéristiques géométriques de la base. Elle repose sur l'utilisation conjointe des propriétés de la connexion de Galois et de la transposition. Même si elle intègre des outils classiques de la communauté apprentissage (extraction de motifs, connexion de Galois, treillis de concepts), nous pensons que cette méthode est nouvelle

car elle propose une combinaison astucieuse de méthodes conventionnelles pour les contextes de dimensions particulières.

L'introduction de la notion de contrainte transposée et l'utilisation de la relaxation de contrainte étend cette approche à des contraintes variées. La définition d'un cadre formel pour la transposition de contraintes permet des extractions sous contrainte dans des contextes comportant un grand nombre d'attributs.

Cette partie relate des travaux qui s'appuient sur des collaborations à Lyon avec le Laboratoire d'Informatique des Images et des Systèmes d'information (LIRIS CNRS UMR 5205), le Centre de Génétique Moléculaire et Cellulaire (CGMC, CNRS UMR 5534) et l'INRA/INSERM U449 pour le chapitre 8 qui présente notre nouvelle méthode d'extraction de motifs et avec l'Équipe Universitaire de Recherche en Informatique de Saint-Étienne (EURISE, CNRS EA 3721) pour le chapitre 9 qui détaille la méthode générique d'extraction de motifs contraints dans les bases de données très larges.

Chapitre 8

Extraction de motifs fréquents par transposition de base de données

Ce chapitre présente une nouvelle méthode d'extraction de motifs fréquents dans les bases de données qui comportent un grand nombre d'attributs devant le nombre d'objets. Par exemple, lors de nos expériences décrites au chapitre 13, les matrices décrivent une très petite quantité de situations biologiques (90) à l'aide de 27 679 expressions de gènes. Notre motivation concerne la réutilisation des algorithmes classiques d'extraction de motifs de la communauté fouille de données pour ce type de matrices. Le but n'est pas de développer des solutions dédiées et adaptées aux formats de données inhabituels, mais plutôt d'aménager les calculs pour exploiter les solutions connues.

Nous montrons dans un premier temps, grâce aux propriétés de la connexion de Galois, que l'ensemble des concepts d'une base coïncide, moyennant une simple inversion, avec ceux de la base transposée. Des arguments combinatoires, développés au chapitre 4, avaient déjà suggéré cette propriété. Cette correspondance est l'élément formel central de notre méthode. Elle est simple à mettre en œuvre et ne nécessite qu'un classique extracteur de concepts. Elle permet d'effectuer des extractions dans des bases de données réputées difficiles du fait d'un espace de recherche prohibitif.

La transposition de base de données est introduite à la section 8.1. L'extraction de motifs fréquents par transposition de concepts est expliquée section 8.2. La section 8.3 détaille notre méthode sur un exemple jouet, et des expériences sur des données biologiques sont réalisées à la section 8.4. Enfin, la section 8.5 discute des impacts de cette méthode.

8.1 Transposition de base de données

La transposition d'un contexte permute les ensembles d'attributs et d'objets et transpose la relation qui les unit. Pour cette transformation, il est également nécessaire de redéfinir les opérateurs de connexion de GALOIS correspondants (la section 2.2.1 introduit ces opérateurs).

Définition 32 (Base transposée) *Soit $r = (\mathcal{A}, \mathcal{O}, R)$ une base de données munie des opérateurs f et g de* GALOIS. *La base transposée s'écrit ${}^t r = (\mathcal{O}, \mathcal{A}, {}^t R)$ où $(o, a) \in {}^t R \iff (a, o) \in R$. Les opérateurs de* GALOIS *correspondants sont ${}^t f = g$ et ${}^t g = f$.*

8.2 Extraction de motifs fréquents par transposition de concepts

Nous définissons ci-dessous le transposé d'un concept de la base originale :

Définition 33 (Concept transposé) *Pour un concept $c = (A, O)$ de r, le concept transposé dans ${}^t r$ est ${}^t c = (O, A)$.*

La transposition vérifie une propriété fondamentale : les concepts transposés sont des concepts pour la base transposée.

Proposition 10 ([Wille, 1982]) *(A, O) est un concept de $r \iff (O, A)$ est un concept de ${}^t r$.*

Preuve : Soit (A, O) un concept de r. Alors $O = g(A)$ (et $A = f(O)$) et par définition de la base transposée $g(A) = {}^t f(A)$ (et $f(O) = {}^t g(O)$). Donc le concept $({}^t f(A), A)$ (ou $(O, {}^t g(O))$) associe deux motifs fermés en connexion par ${}^t g$ ou ${}^t f$: c'est un concept de ${}^t r$. □

Ainsi, l'ensemble des concepts d'une base peut être obtenu en pratiquant une extraction dans la base transposée, puis en transposant les concepts obtenus pour en connaître les concepts originaux. La transposition constitue un changement de représentation [Courtine, 2002, Zucker, 2001], dont les dimensions sont plus propices pour l'exécution des algorithmes. Cette méthode permet de fournir une rapide solution aux problèmes posés par le vaste espace de recherche d'une base comportant un grand nombre d'attributs devant celui d'objets.

Rappelons que les motifs fermés qui constituent les concepts sont les motifs maximaux des classes d'équivalence des supports et forment une représentation condensée de l'ensemble des motifs fréquents. L'extraction des concepts dans la transposée permet donc de regénérer l'intégralité des motifs fréquents et leurs supports.

8.3 Exemple

Illustrons notre méthode avec un exemple jouet de base très « large » (table 8.1). Ce contexte décrit trois objets $o_1, \ldots, o_3$ à l'aide de quatre attributs $a_1, \ldots, a_4$: il comporte plus d'attributs

	a_1	a_2	a_3	a_4
o_1	×	×	×	
o_2	×	×	×	
o_3		×	×	×

TAB. 8.1 – Exemple de matrice d'expression de gènes.

que d'objets. La figure 8.1 montre l'espace de recherche parcouru par un algorithme par niveaux qui extrait les concepts. Chaque nœud du treillis symbolise l'association entre un motif d'attributs et son support (un motif d'objets). La partie haute de la figure expose l'espace de recherche parcouru dans le contexte original en spécialisant les attributs et la partie basse fournit celui obtenu en spécialisant les objets. Les classes d'équivalence regroupent les nœuds qui ont le même support et les concepts sont encadrés.

Bien sûr, le nombre de concepts (ou de classes d'équivalence) est identique selon les deux formats de la base (originale ou transposée). Le nombre de classes d'équivalence n'étant pas modifié par la transposition de contexte, la différence entre les espaces de recherche selon les attributs (base originale) ou les objets (base opposée) réside dans la taille de ces classes. Dans notre exemple, relativement aux attributs, elles contiennent 4, 3 et 4 éléments et relativement aux objets elles contiennent 3, 3 et 1 éléments. Finalement, l'exploration des motifs selon la spécialisation des attributs fait apparaître 11 nœuds, tandis que celle des objets ne comporte que 7 nœuds : elle est plus efficace.

8.4 Expériences

La méthode fondée sur la transposition s'est avérée particulièrement pertinente dans le cadre de notre collaboration avec des biologistes de l'INRA/INSERM U449 de Lyon sur l'expression de gènes humains dans des cellules musculaires en réponse à l'insuline [Rome *et al.*, 2003]. La matrice d'expression `sain8` contient 6 objets (les situations biologiques) pour 1065 attributs (les expressions de gènes). L'extraction des 41 concepts sur cette matrice nécessite quelques minutes et ils proviennent de 667 831 motifs 1-libres [Rioult et Crémilleux, 2003b]. Les classes d'équivalence contiennent donc 16 300 motifs en moyenne. Dans la matrice transposée, l'extraction ne prend que quelques centièmes de secondes et se contente de 42 motifs 1-libres, pour bien sûr les mêmes 41 motifs fermés. Les classes d'équivalence contiennent quasiment un seul élément.

Le tableau 8.2 détaille les expériences pour `sain8` et sa transposée. Nous voulons ici évaluer la quantité de critères d'élagages relatifs à l'antimonotonie de la propriété de 1-liberté. Sous la dénomination « succès », la table présente le nombre de motifs qui ont satisfait ces deux critères (ce sont les candidats). La colonne « échecs » indique combien de motifs ne satisfont pas le critère 2 (*i.e.* l'un de ses sous-ensembles n'est pas 1-libre). On constate que l'algorithme appliqué

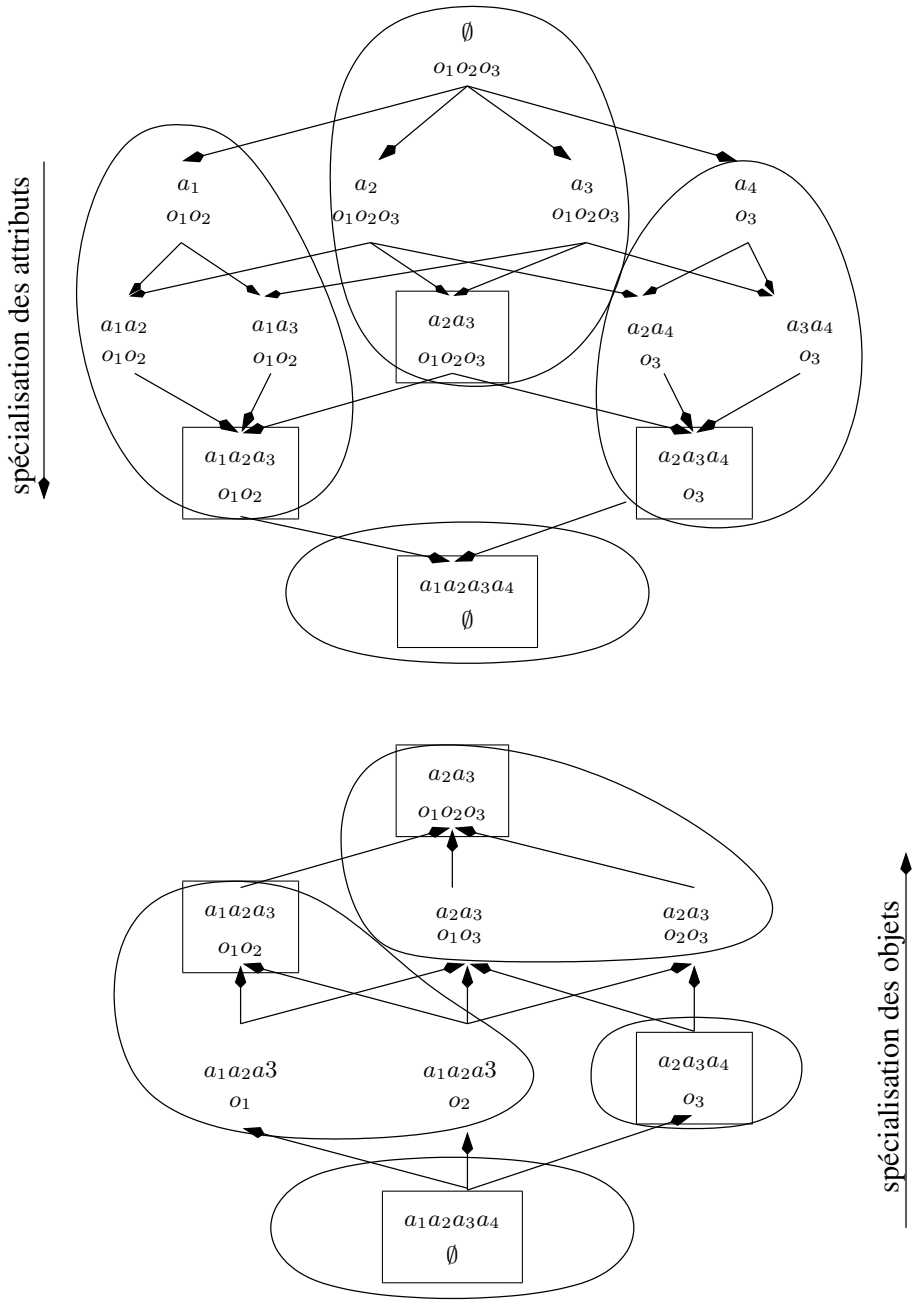

FIG. 8.1 – Extraction dans la base d'exemple, selon les attributs ou les objets.

à la base originale teste beaucoup de motifs qui s'avèrent finalement inintéressants (un motif valide pour trois invalides). Dans la transposée, l'algorithme est nettement plus efficace et cette tendance est inversée.

Cet exemple est tout à fait symptomatique de l'efficacité de l'extraction dans la transposée. Pour les contextes biologiques de dimensions « pathologiques », cette amélioration est spectaculaire. Dans le chapitre 13 où nous relatons des expériences sur une très grosse matrice d'expression

	sain8		^{t}sain8	
long.	succès	échecs	succès	échecs
1	777	0	6	0
2	172 548	128 928	15	0
3	2 315 383	4 713 114	16	4
4	2 965 726	9 371 325	6	9
5	0	1 544 485	0	2
Total	5 454 434	15 757 852	43	15
libres	667 831		42	
fermés	41			

TAB. 8.2 – Échecs/succès du critère d'élagage sur la base sain8.

SAGE, les extractions ne sont pas faisables sans cette méthode. Remarquons que même si le domaine de la fouille de textes est généralement mieux doté en objets que la biologie, ceux-ci peuvent comporter un grand nombre d'attributs. L'extraction des concepts dans la transposée offre alors une opportunité de résoudre ce classique problème de dimensions.

8.5 Discussion

Il est clair que la transposition de contexte n'est intéressante que si l'algorithme utilisé pour extraire les concepts est sensible à la taille de l'espace de recherche. C'est le cas pour celui que nous utilisons, qui calcule les motifs 1-libres et leurs fermetures selon une approche par niveaux (voir section 11.4). En revanche, pour d'autres algorithmes, comme celui de BORDAT, l'intérêt est moindre. Dans [Fu, 2005], H. FU compare les algorithmes classiques d'extraction de la totalité des concepts (GANTER, BORDAT, CHEIN, NORRIS) selon leur efficacité sur des bases de test de l'UCI et sur le contexte « pire cas » constitué par une matrice pleine sauf sur la diagonale (voir figure 4.1-a page 53). Il montre que l'algorithme de GANTER est le plus efficace et qu'il est très sensible à la transposition.

La transposition du contexte implique le calcul l'intégralité des concepts. On peut s'interroger sur l'intérêt d'une telle opération et surtout sur sa faisabilité. Dans [Rioult *et al.*, 2003b, Besson *et al.*, 2004], nous avons montré que cette solution est praticable pour un jeu de données beaucoup plus conséquent que sain8, puisqu'il décrit 90 situations biologiques à l'aide de 27 679 identifiants de gènes SAGE (cette base de données est utilisée au chapitre 13). Selon les taux de discrétisation employés pour coder les facteurs d'expression, l'extraction complète dans la base transposée n'est pas toujours réalisable. Mais lorsqu'elle l'est, elle est toujours plus efficace que l'extraction dans le sens usuel. Dans de nombreux cas, l'extraction directe est impossible à réaliser alors que

l'extraction dans la transposée est praticable. Le contexte transposé fournit donc une réponse immédiate au difficile problème des matrices très larges, et utilise des techniques simples et éprouvées.

Chapitre 9

Extraction de motifs contraints par transposition de contrainte

Ce travail [Jeudy et Rioult, 2005b, Jeudy et Rioult, 2005b] a été réalisé en collaboration avec Baptiste Jeudy *(EURISE), spécialiste de l'extraction de motifs sous contraintes.*

Dans ce chapitre, nous souhaitons étendre notre résultat d'extraction de motifs à l'aide de la base transposée vers la recherche de motifs sous contrainte. Le chapitre précédent a montré que les extractions de concepts pratiquées dans la base et sa version transposée coïncident. L'extraction sous contrainte peut donc, dans un premier temps, se concevoir comme une sélection des concepts pertinents à partir de l'ensemble des concepts.

Cependant, l'extraction de l'intégralité de la collection des concepts doit être évitée, car c'est une tâche souvent trop coûteuse. Ce chapitre présente un résultat général concernant l'utilisation de la *contrainte transposée* : les concepts contraints coïncident avec les transposés des concepts satisfaisant la contrainte transposée. Nous fournissons également une méthode complète de transposition des contraintes, sous la forme d'un dictionnaire des contraintes usuelles. La relaxation de la contrainte permet de généraliser la solution aux motifs qui ne sont pas nécessairement fermés.

La section 9.1 définit la contrainte de recherche transposée et la section 9.2 montre que l'extraction dans la base transposée réalisée avec cette contrainte coïncide, sur les concepts, avec l'extraction dans la base originale. La section 9.3 explique comment calculer la contrainte transposée. Enfin, la méthode complète d'extraction sous contrainte est détaillée à la section 9.4 et la section 9.5 discute de ses impacts.

9.1 Contrainte transposée

Rappelons que notre stratégie face aux bases de données comportant un grand nombre d'attributs consiste à exécuter des algorithmes classiques sur la base transposée. Ce faisant, ce n'est

pas l'espace de recherche selon les attributs qui est parcouru car il est trop vaste, mais l'espace de recherche concernant les objets. Les algorithmes de recherche sous contrainte utilisés doivent donc prendre en compte des contraintes portant sur les motifs d'objets. Pour une contrainte $\mathcal{C}$ définie sur un motif d'attributs, nous définissons alors la contrainte transposée de $\mathcal{C}$ pour un motif d'objets (f et g sont les opérateurs de Galois, h est l'opérateur de fermeture, cf. section 2.2.1 page 35) :

Définition 34 (Contrainte transposée) *Soit $\mathcal{C}$ une contrainte, fonction booléenne sur $\mathcal{L}_{\mathcal{A}} = 2^{\mathcal{A}}$. La contrainte transposée ${}^t\mathcal{C}$ sur un motif fermé $O \subseteq \mathcal{O}$ est telle que*

$$ {}^t\mathcal{C}(O) = \mathcal{C}(f(O)). $$

Par exemple, la transposée de la contrainte de fréquence supérieure à un seuil γ correspond à la contrainte de « longueur supérieure à γ ». En effet, $\mathcal{C}_{freq}(X) = |g(X)| \geq \gamma$, donc pour O fermé ${}^t\mathcal{C}(O) = \mathcal{C}(f(O)) = |g(f(O))| \geq \gamma = |O| \geq \gamma$, car O est fermé donc $g(f(O)) = O$.

La section 9.3 traite la transposition des contraintes classiques. Dans [Jeudy et Rioult, 2005a], la transposition est définie comme la *projection* d'une contrainte.

9.2 Extraction de motifs fermés sous contrainte

La contrainte transposée possède une propriété intéressante pour l'extraction des motifs fermés : elle permet d'exhiber les motifs contraints de la base originale, en menant des calculs dans la base transposée.

Nous introduisons la contrainte $\mathcal{C}_{\mathsf{close}}$ qui indique si un motif est fermé et nous définissons l'extraction de motifs fermés contraints de la façon suivante : étant donnée une base de données r et une contrainte $\mathcal{C}$, nous cherchons tous les motifs fermés satisfaisant $\mathcal{C}$ dans r. Plus formellement, nous calculons la collection

$$ \{A \subseteq \mathcal{A} \mid \mathcal{C}(A, r) \wedge \mathcal{C}_{\mathsf{close}}(A, r)\}. $$

Le théorème suivant montre comment calculer cette collection dans la base transposée, à l'aide de la contrainte transposée :

Théorème 5

$$ \{A \subseteq \mathcal{A} \mid \mathcal{C}(A) \wedge \mathcal{C}_{\mathsf{close}}(A)\} = \left\{f(O) \mid O \subseteq \mathcal{O} \wedge {}^t\mathcal{C}(O) \wedge \mathcal{C}_{\mathsf{close}}(O)\right\}. $$

Preuve : Grâce à la définition 34, $\{f(O) \mid {}^t\mathcal{C}(O) \wedge \mathcal{C}_{\mathsf{close}}(O)\} = \{f(O) \mid \mathcal{C}(f(O)) \wedge \mathcal{C}_{\mathsf{close}}(O)\} = \{A \mid \exists O \mid \mathcal{C}(A) \wedge A = f(O)\} = \{A \mid \mathcal{C}(A) \wedge \mathcal{C}_{\mathsf{close}}(A)\}$. □

Ce théorème signifie que si nous extrayons la collection des motifs fermés d'objets satisfaisant ${}^t\mathcal{C}$ dans la base transposée, alors nous obtenons celle des motifs fermés d'attributs satisfaisant $\mathcal{C}$ en calculant $f(O)$ pour chaque motif fermé d'objets O découvert dans la base transposée.

Le fait que nous ayons uniquement besoin des motifs fermés et non de tous les motifs est très intéressant, car ces motifs sont moins nombreux et peuvent être extraits plus efficacement. Pour extraire les motifs fermés sous contrainte dans une base très large, la stratégie que nous proposons est la suivante :

1. calculer la contrainte ${}^t\mathcal{C}$ (voir section suivante) ;
2. utiliser l'un des algorithmes connus pour extraire les motifs fermés O d'objets satisfaisant ${}^t\mathcal{C}$ dans la base transposée. Ces algorithmes extraient classiquement les motifs fermés fréquents, mais il est possible de les adapter pour intégrer des critères d'élagage en relation avec la contrainte ;
3. calculer $f(O)$ pour chaque motif O découvert pour retourner dans l'espace des attributs, ce qui fournit la solution au problème de recherche des motifs fermés d'attributs contraints. Dans la pratique, tous les algorithmes connus calculent le support pour déterminer la fréquence. Dans la base transposée, le support de O est $f(O)$. $f(O)$ est donc facilement disponible.

Notons que $f(O)$ est calculé à la dernière étape de notre stratégie. En particulier, $f(O)$ n'est pas utilisé pour le test de la contrainte. Cet argument est impératif si l'on souhaite utiliser les algorithmes connus, qui implémentent des contraintes sur O et pas sur $f(O)$. Les contraintes transposées que nous décrivons ci-dessous n'utilisent pas $f(O)$ dans leur expression.

9.3 Etude de transposée de contraintes complexes

Il est intéressant d'étudier l'effet de la transposition relativement à la monotonie ou l'antimonotonie d'une contrainte, car de nombreux algorithmes utilisent cette propriété.

Proposition 11 *Si une contrainte $\mathcal{C}$ est monotone (resp. antimonotone), la contrainte transposée ${}^t\mathcal{C}$ est antimonotone (resp. monotone).*

Preuve : f et g sont décroissantes (propriété 2 page 36), ce qui inverse monotonie et antimonotonie. □

Nous souhaitons également traiter des contraintes complexes, assemblées à l'aide d'opérateurs booléens. Cette construction est utile pour l'algébrisation [De Raedt *et al.*, 2002] du problème de recherche sous contrainte, où chaque contrainte complexe est décomposée en disjonction ou conjonction de contraintes élémentaires.

Proposition 12 *Si $\mathcal{C}_1$ et $\mathcal{C}_2$ sont deux contraintes, alors on a :*

$$ {}^t(\mathcal{C}_1 \wedge \mathcal{C}_2) = {}^t\mathcal{C}_1 \wedge {}^t\mathcal{C}_2 $$

$$^t(\mathcal{C}_1 \vee \mathcal{C}_2) = {}^t\mathcal{C}_1 \vee {}^t\mathcal{C}_2$$

$$^t(\neg\mathcal{C}_1) = \neg{}^t\mathcal{C}_1$$

Preuve : Pour la conjonction : $^t(\mathcal{C}_1 \wedge \mathcal{C}_2)(O) = (\mathcal{C}_1 \wedge \mathcal{C}_2)(f(O)) = \mathcal{C}_1(f(O)) \wedge \mathcal{C}_2(f(O)) = ({}^t\mathcal{C}_1 \wedge {}^t\mathcal{C}_2)(O)$. La preuve est similaire pour la disjonction et la négation. □

Les deux propositions suivantes fournissent la contrainte transposée de deux contraintes classiques : les contraintes de sous-ensemble et de sur-ensemble, par rapport à un motif paramètre constant E.

Proposition 13 *Soit $\mathcal{C}_{\subseteq E}$ la contrainte définie par $\mathcal{C}_{\subseteq E}(A) = (A \subseteq E)$ où E est un motif paramètre constant. Si E est fermé, alors pour O un motif d'objets,*

$$^t\mathcal{C}_{\subseteq E}(O) \Leftrightarrow g(E) \subseteq h(O),$$

et si E n'est pas fermé,

$$^t\mathcal{C}_{\subseteq E}(O) \Rightarrow g(E) \subseteq h(O).$$

Preuve : $^t\mathcal{C}_{\subseteq E}(O) \Leftrightarrow \mathcal{C}_{\subseteq E}(f(O)) \Leftrightarrow (f(O) \subseteq E) \Rightarrow (g(E) \subseteq g(f(O))) \Leftrightarrow (g(E) \subseteq h(O))$. Réciproquement (si E est fermé) : $(g(E) \subseteq g(f(O))) \Rightarrow (f(O) \subseteq h(E)) \Rightarrow (f(O) \subseteq E)$. □

Proposition 14 *Soit $\mathcal{C}_{\supseteq E}$ la contrainte définie par : $\mathcal{C}_{\supseteq E}(A) = (A \supseteq E)$ où E est un motif constant. Alors*

$$^t\mathcal{C}_{\supseteq E}(O) \Leftrightarrow g(E) \supseteq h(O).$$

Preuve : $^t\mathcal{C}(O) \Leftrightarrow (E \subseteq f(O)) \Rightarrow (g(f(O)) \subseteq g(E)) \Leftrightarrow (h(O) \subseteq g(E))$. Réciproquement, $(g(f(O)) \subseteq g(E) \Rightarrow (fg(E) \subseteq fgf(O)) \Rightarrow fg(E) \subseteq f(O) \Rightarrow h(E) \subseteq f(O) \Rightarrow E \subseteq f(O)$. □

Ces deux contraintes syntaxiques sont fondamentales car elles peuvent être utilisées pour construire de nombreuses autres contraintes. En fait, toute contrainte syntaxique peut être construite à l'aide des deux contraintes précédentes, en utilisant des conjonctions, des disjonctions et des négations. En outre, ces contraintes ont été identifiées dans [Goethals et Van den Bussche, 2000, Bonchi *et al.*, 2003b] pour formaliser des techniques de réduction des bases de données.

La table 9.1 donne la contrainte transposée de plusieurs contraintes classiques. Ces résultats s'obtiennent facilement en utilisant les trois propositions précédentes. Par exemple, si $\mathcal{C}(A) = (A \cap E \neq \emptyset)$, cela peut se récrire $A \not\subseteq \overline{E}$ ($\overline{E}$ est le complémentaire de E, *i.e.* $\mathcal{A} \setminus E$) et donc $\neg(A \subseteq \overline{E})$. En utilisant les propositions 12 et 13, la contrainte transposée est $\neg(g(\overline{E}) \subseteq O)$ (si $\overline{E}$ est fermé) et finalement $g(\overline{E}) \not\subseteq O$. Si $\overline{E}$ n'est pas fermé et $E = \{e_1, ..., e_n\}$, nous pouvons redéfinir la contrainte par $\mathcal{C}(A) = (e_1 \in A \vee e_2 \in A \vee ... \vee e_n \in A)$, puis en utilisant les propositions 12 et 14, nous obtenons la contrainte transposée $^t\mathcal{C}(O) = (O \subseteq g(e_1) \vee ... \vee O \subseteq g(e_n))$. Ces expressions

Contrainte $\mathcal{C}(A)$	Contrainte transposée ${}^t\mathcal{C}(O)$
$\mathcal{F}(A)\,\theta\,\alpha$	$\lvert O\rvert\,\theta\,\alpha$
$A \subseteq E$	si E est fermé : $g(E) \subseteq O$
	sinon : $O \not\subseteq g(f_1) \wedge ... \wedge O \not\subseteq g(f_m)$
$E \subseteq A$	$O \subseteq g(E)$
$A \not\subseteq E$	si E est fermé : $g(E) \not\subseteq O$
	sinon : $O \subseteq g(f_1) \vee ... \vee O \subseteq g(f_m)$
$E \not\subseteq A$	$O \not\subseteq g(E)$
$A \cap E = \emptyset$	si $\overline{E}$ est fermé : $g(\overline{E}) \subseteq O$
	sinon : $O \not\subseteq g(e_1) \wedge ... \wedge O \not\subseteq g(e_n)$
$A \cap E \neq \emptyset$	si $\overline{E}$ est fermé : $g(\overline{E}) \not\subseteq O$
	sinon : $O \subseteq g(e_1) \vee ... \vee O \subseteq g(e_n)$
$\mathrm{SUM}(A)\,\theta\,\alpha$	$\mathcal{F}_p(O)\,\theta\,\alpha$ (voir texte)
$\mathrm{MIN}(A)\,\theta\,\alpha$	(voir texte)
$\mathrm{MAX}(A)\,\theta\,\alpha$	(voir texte)
$\theta \in \{<, >, \leq, \geq\}$	

TAB. 9.1 – Contraintes transposées de contraintes classiques. A est un motif fermé d'attributs, $E = \{e_1, e_2, ..., e_n\}$ est un motif constant, O est un motif fermé d'objets et $\overline{E} = \mathcal{A} \setminus E = \{f_1, f_2, ..., f_m\}$.

sont utiles car elles ne nécessitent pas le calcul de $f(O)$. À la place, il nous faut $g(\overline{E})$ ou $g(e_i)$. Comme E est constant, ces valeurs sont calculées une seule fois au cours d'une phase préliminaire.

Donnons quelques exemples d'utilisation de la table 9.1 :

Exemple 1 *Nous montrons pour la base de la table 8.1 page 113 comment calculer la contrainte transposée de* $\mathcal{C}(A) = (A \cap a_1a_4 \neq \emptyset)$. *Le motif* $\overline{a_1a_4} = a_2a_3$ *est fermé dans la base. D'après le tableau 9.1,* ${}^t\mathcal{C}(O) = (g(a_2a_3) \not\subseteq O)$. *Comme* $g(a_2a_3) = o_1o_2o_3$, ${}^t\mathcal{C}(O) = (o_1o_2o_3 \not\subseteq O)$. *Les motifs fermés d'objets satisfaisant cette contrainte appartiennent à* $\{\emptyset, o_1o_2, o_3\}$. *Si nous appliquons* f *pour revenir à l'espace des attributs, nous obtenons finalement l'ensemble des solutions* $\{a_1a_2a_3a_4, a_1a_2a_3, a_2a_3a_4\}$ *qui contient, comme prévu et prouvé par le théorème 5, les motifs fermés d'attributs satisfaisant* $\mathcal{C}$.

Exemple 2 *Considérons maintenant la contrainte* $\mathcal{C}(A) = (A \cap a_1a_2 \neq \emptyset)$. *Dans ce cas* $E = \overline{a_1a_2} = a_3a_4$ *n'est pas fermé. Nous utilisons la seconde expression de la table 9.1 pour calculer la contrainte transposée* ${}^t\mathcal{C}(O) = (O \subseteq g(a_1) \vee O \subseteq g(a_2))$. *Comme* $g(a_1) = o_1o_2$ *et* $g(a_2) = o_1o_2o_3$, ${}^t\mathcal{C}(O) = (O \subseteq o_1o_2 \vee O \subseteq o_1o_2o_3)$ *qui peut être simplifié en* ${}^t\mathcal{C}(O) = (O \subseteq o_1o_2o_3)$. *Tous les motifs fermés d'objets satisfont cette contrainte, ce qui n'est pas surprenant puisque tous les*

motifs fermés d'attributs satisfont $\mathcal{C}$.

Exemple 3 *Notre dernier exemple est la contrainte* $\mathcal{C}(A) = (|A \cap a_1a_2a_4| \geq 2)$. *Elle peut être récrite* $\mathcal{C}(A) = ((a_1a_2 \subseteq A) \vee (a_1a_4 \subseteq A) \vee (a_2a_4 \subseteq A))$. *En utilisant la proposition 12 et la table 9.1, nous obtenons* ${}^t\mathcal{C}(O) = ((O \subseteq g(a_1a_2)) \vee (O \subseteq g(a_1a_4)) \vee (O \subseteq g(a_2a_4)))$ *qui est* ${}^t\mathcal{C}(O) = ((O \subseteq o_1o_2) \vee (O \subseteq \emptyset) \vee (O \subseteq o_3))$. *Les motifs fermés d'objets satisfaisant* ${}^t\mathcal{C}$ *sont donc dans* $\{\emptyset, o_1o_2, o_3\}$ *et les motifs d'attributs correspondant sont dans* $\{a_1a_2a_3a_4, a_1a_2a_3, a_2a_3a_4\}$.

D'autres contraintes intéressantes concernent l'agrégation [Ng *et al.*, 1998]. Si une valeur numérique $a.v$ est associée à chaque attribut $a \in \mathcal{A}$, nous pouvons définir les contraintes de la forme $\text{SUM}(A)\,\theta\,\alpha$ pour plusieurs opérateurs comme SUM, MIN, MAX ou AVG, où $\theta \in \{<, >, \leq, \geq\}$ et α est une valeur numérique. Dans ce cas, $\text{SUM}(A)$ dénote la somme de tous les $a.v$ pour tous les attributs a dans A.

Les contraintes $\text{MIN}(A)\,\theta\,\alpha$ et $\text{MAX}(A)\,\theta\,\alpha$ sont des cas particuliers des contraintes de la table 9.1. Par exemple, si nous définissons $sup_\alpha = \{a \in \mathcal{A} \mid a.v > \alpha\}$ alors $\text{MIN}(A) > \alpha$ est exactement $A \subseteq sup_\alpha$ et $\text{MIN}(A) \leq \alpha$ est $A \not\subseteq sup_\alpha$. La même type de relation est valable pour l'opérateur MAX : $\text{MAX}(A) > \alpha$ est équivalent à $A \cap sup_\alpha \neq \emptyset$ et $\text{MAX}(A) \leq \alpha$ est équivalent à $A \cap sup_\alpha = \emptyset$. Dans ce cas, comme α est constant, l'ensemble sup_α peut être précalculé.

La contrainte $\text{SUM}(A)\,\theta\,\alpha$ est plus difficile. Son expression est ${}^t\mathcal{C}(O) = (\text{SUM}(f(O))\,\theta\,\alpha)$. Dans la base de données, $f(O)$ est un ensemble d'attributs, donc dans la base transposée c'est un ensemble d'objets et O est un ensemble d'attributs. Les valeurs $a.v$ sont attachées aux objets de la base transposée et $\text{SUM}(f(O))$ est la somme de ces valeurs pour les objets contenant O. C'est donc la fréquence pondérée de O dans la base transposée et chaque objet contenant O contribue pour $a.v$ au total (nous notons cette fréquence pondérée $\mathcal{F}_p(O)$). Il est facile d'adapter les algorithmes classiques pour compter cette fréquence pondérée. Ce calcul est identique à celui de la fréquence classique sauf que chaque motif compté contribue à la fréquence par une valeur différente de 1. La forme définitive de la contrainte transposée pour la somme est $\mathcal{F}_p(O)\,\theta\,\alpha$.

9.4 Extraction de motifs sous contrainte

Le cadre formel défini précédemment permet d'extraire les motifs fermés contraints, en s'appuyant sur la connexion de Galois et les concepts. Nous montrons maintenant comment généraliser ce résultat pour réaliser l'extraction de tous les motifs vérifiant la contrainte et pas seulement des motifs fermés. Étant donnée une contrainte $\mathcal{C}$ dans une base r, nous cherchons à calculer l'ensemble

$$\{A \subseteq \mathcal{A} \mid \mathcal{C}(A, r)\}\,.$$

Pour réaliser cette extraction, il est nécessaire de s'appuyer sur les motifs fermés contraints. Puis une relaxation de la contrainte permet d'atteindre tous les motifs d'une classe d'équivalence

définie par un motif fermé satisfaisant la contrainte.

Prenons l'exemple de la contrainte de fréquence : les éléments des classes d'équivalence de support ont tous la même fréquence, celle du motif fermé. Connaissant la fréquence des motifs fermés, il est alors possible de connaître la fréquence d'un motif quelconque en examinant celle de sa fermeture. Par conséquent, si un motif de la classe satisfait la contrainte de fréquence, tous les motifs de la classe la satisfont également.

Ce cas n'est pas général. Certains motifs non fermés peuvent satisfaire la contrainte recherchée, sans que le motif fermé correspondant ait cette propriété. La figure 9.1 présente une telle situation, où les points représentent les motifs, les croix sont les motifs fermés et les lignes délimitent les classes d'équivalence. Les motifs à l'intérieur de la ligne épaisse satisfont la contrainte $\mathcal{C}$ et pas les autres. Les motifs fermés satisfaisant $\mathcal{C}$ sont ceux des classes 3, 4 et 5. Ils permettent de regénérer les motifs valides des trois classes. Pourtant, pour obtenir les deux motifs de la classe 2, nous avons besoin du motif fermé de cette classe, qui ne satisfait pas $\mathcal{C}$. On constate que connaître les motifs fermés satisfaisant $\mathcal{C}$ est insuffisant pour déterminer tous les motifs satisfaisant $\mathcal{C}$.

De plus, remarquons qu'un des motifs de la classe 5 ne satisfait pas $\mathcal{C}$. Nous reviendrons sur ce point à la section 9.4.2. Pour résoudre ce problème, nous introduisons la notion de *relaxation* de contrainte.

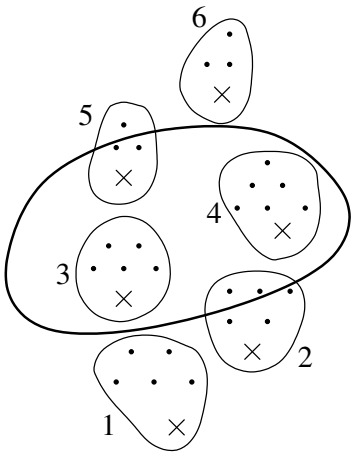

FIG. 9.1 – Exemple de problème contraint où les motifs fermés valides sont insuffisants pour connaître les autres motifs.

9.4.1 Relaxation de contrainte

Pour une contrainte $\mathcal{C}$, le rôle de la relaxation $\mathcal{C}'$ est de déterminer les motifs fermés qui définissent les classes d'équivalence contenant tous les motifs contraints par $\mathcal{C}$. À partir de ces motifs, ceux qui satisfont $\mathcal{C}$ et qui ne sont pas nécessairement fermés sont regénérés. Nous introduisons

d'abord une notion *bonne* relaxation et puis une relaxation *optimale*.

Définition 35 (Bonne relaxation de contrainte) *Soit $\mathcal{C}$ une contrainte. Une bonne relaxation $\mathcal{C}'$ pour $\mathcal{C}$ est telle que*

$$\forall A, \mathcal{C}(A) \Rightarrow \mathcal{C}'(h(A))$$

Par exemple, la contrainte constante « vrai », satisfaite par tout motif, est une bonne relaxation triviale pour toute contrainte. Cependant, elle ne fournit aucune opportunité d'élagage pour les algorithmes d'extraction.

L'ensemble des motifs fermés découverts à l'aide d'une bonne relaxation $\mathcal{C}'$ inclut les fermetures de tout motif quelconque satisfaisant C. Ce faisant, le motif fermé de la classe 2 est pris en compte, ce qui ne serait pas le cas en employant seulement C (cf. figure 9.1). Une bonne relaxation n'est cependant pas parfaite, car elle retrouve plus de motifs que nécessaire. Nous définissons donc une relaxation optimale, qui retrouve exactement les motifs recherchés :

Définition 36 (Relaxation optimale de contrainte) *Soit $\mathcal{C}$ une contrainte. Une relaxation optimale $\mathcal{C}'$ pour $\mathcal{C}$ est telle que*

$$\forall A, \mathcal{C}(A) \iff \mathcal{C}'(h(A))$$

La figure 9.1 suggère le rôle d'une relaxation optimale pour la contrainte de la figure 9.1.

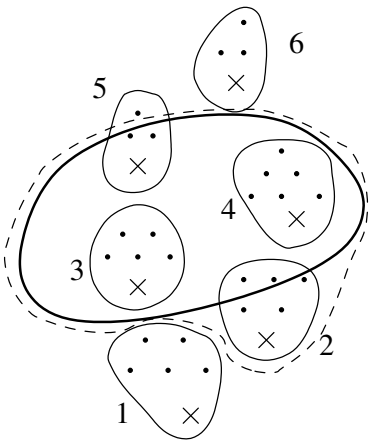

FIG. 9.2 – Relaxation optimale de $\mathcal{C}$. La contrainte $\mathcal{C}$ est représentée par la ligne pleine, la relaxation optimale par la ligne brisée.

Nous indiquons ci-dessous la relaxation optimale de quelques contraintes classiques, en commençant par deux cas triviaux :

Proposition 15

- *la relaxation optimale d'une contrainte monotone est la contrainte elle-même ;*
- *la relaxation optimale de la contrainte de fréquence est la contrainte de fréquence elle-même.*

Preuve : La fermeture étant extensive (proposition 2), alors $A \subseteq h(A)$. Une contrainte monotone, préservée par spécialisation, est donc également vérifiée par la fermeture de tout motif qui la satisfait. Dans le cas de la contrainte de fréquence minimale, A et $h(A)$ ont le même support, donc le même comportement vis-à-vis d'une contrainte de fréquence. □

La proposition suivante permet de calculer une contrainte relaxée optimale pour la disjonction de deux contraintes :

Proposition 16 *Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux contraintes, $\mathcal{C}'_1$ et $\mathcal{C}'_2$ leur relaxation optimale, alors $\mathcal{C}'_1 \vee \mathcal{C}'_2$ est une relaxation optimale de $\mathcal{C}_1 \vee \mathcal{C}_2$.*

Enfin, la proposition à suivre calcule une bonne contrainte relaxée pour la conjonction de deux contraintes :

Proposition 17 *Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux contraintes, $\mathcal{C}'_1$ et $\mathcal{C}'_2$ leur relaxation optimale, alors $\mathcal{C}'_1 \wedge \mathcal{C}'_2$ est une bonne relaxation de $\mathcal{C}_1 \wedge \mathcal{C}_2$.*

Preuve : Pour montrer que $\mathcal{C}'_1 \vee \mathcal{C}'_2$ est une relaxation optimale, nous devons prouver que si A est fermé et satisfait $\mathcal{C}'$, alors il existe un motif B satisfaisant $\mathcal{C}$ tel que $h(B) = A$ (cf. définition 36). Nous utiliserons ces deux faits dans notre preuve.
Soit A un motif satisfaisant $\mathcal{C}_1 \wedge \mathcal{C}_2$. Cela signifie que A satisfait $\mathcal{C}_1$ et satisfait également $\mathcal{C}_2$. Donc $h(A)$ satisfait $\mathcal{C}'_1$ et également $\mathcal{C}'_2$, car $\mathcal{C}'_1$ et $\mathcal{C}'_2$ sont les relaxations optimales de $\mathcal{C}_1$ et $\mathcal{C}_2$: $h(A)$ satisfait $\mathcal{C}'_1 \wedge \mathcal{C}'_2$. $\mathcal{C}'_1 \wedge \mathcal{C}'_2$ est donc une bonne relaxation de $\mathcal{C}_1 \wedge \mathcal{C}_2$.
De la même manière, $\mathcal{C}'_1 \vee \mathcal{C}'_2$ est une bonne relaxation de $\mathcal{C}_1 \vee \mathcal{C}_2$. Prouvons maintenant qu'elle est optimale : soit A un motif fermé satisfaisant $\mathcal{C}'_1 \vee \mathcal{C}'_2$. Supposons que A satisfait $\mathcal{C}'_1$. Comme $\mathcal{C}'_1$ est optimale pour $\mathcal{C}_1$, il existe B satisfaisant $\mathcal{C}_1$ tel que $h(B) = A$. Il existe donc un B satisfaisant $\mathcal{C}_1 \vee \mathcal{C}_2$ et $h(B) = A$: la relaxation est optimale. □

Il n'y a pas de relaxation simple pour la négation d'une contrainte. Cependant, si la contrainte est usuelle (*i.e.* dans la table 9.1), sa négation est également dans cette table et nous pouvons pousser la négation dans l'écriture de la contrainte, comme le montre l'exemple suivant.

Exemple 4 *Soit $\mathcal{C}(A) = (\neg(((\mathcal{F}(A) > 3) \wedge (A \not\subseteq E)) \vee (A \cap F = \emptyset)))$ où E et F sont deux motifs constants. En poussant la négation dans l'écriture de la contrainte, on obtient $\mathcal{C}(A) = ((\neg(\mathcal{F}(A) > 3) \vee \neg(A \not\subseteq E)) \wedge \neg(A \cap F = \emptyset))$ et finalement,*

$$\mathcal{C}(A) = (((\mathcal{F}(A) \leq 3) \vee (A \subseteq E)) \wedge (A \cap F \neq \emptyset)).$$

Avec les propositions 15, 16, 17, et la table 9.2, nous pouvons calculer une bonne relaxation $\mathcal{C}'$ pour $\mathcal{C}$:

$$\mathcal{C}'(A) = (((\mathcal{F}(A) \leq 3) \vee (A \subseteq h(E))) \wedge (A \cap F \neq \emptyset)).$$

La table 9.2 indique de bonnes relaxations pour les autres contraintes de la table 9.1 qui ne sont pas couvertes par les propositions précédentes (*i.e.* qui ne sont pas monotones), excepté pour les contraintes antimonotones utilisant SUM pour lesquelles nous n'avons pas de relaxation non triviale.

Contrainte $\mathcal{C}(A)$	bonne relaxation $\mathcal{C}'(A)$
$A \subseteq E$	$A \subseteq h(E)$
$E \not\subseteq A$	$A \subseteq h(\overline{e_1}) \vee A \subseteq h(\overline{e_2}) \vee ... \vee A \subseteq h(\overline{e_n})$
$A \cap E = \emptyset$	$A \subseteq h(\overline{E})$
$\text{MIN}(A) > \alpha$	$A \subseteq h(sup_\alpha)$
$\text{MAX}(A) < \alpha$	$A \subseteq h(\overline{supeq_\alpha})$

TAB. 9.2 – Bonnes relaxations de contraintes classiques. A est un motif fermé d'attributs, $E = \{e_1, e_2, ..., e_n\}$ un motif constant.

Preuve :

$\mathcal{C}(A) = (A \subseteq E)$, $\mathcal{C}'(A) = (A \subseteq h(E))$: si $A \subseteq E$ alors $h(A) \subseteq h(E)$. Cela signifie que $\mathcal{C}(A) \Rightarrow \mathcal{C}'(h(A))$ donc $\mathcal{C}'$ est une bonne relaxation de $\mathcal{C}$;

$\mathcal{C}(A) = (E \not\subseteq A)$: si $E = \{e_1, e_2, ..., e_n\}$, cette contrainte peut être récrite $\{e_1\} \not\subseteq A \vee \{e_2\} \not\subseteq A \vee \ldots \vee \{e_n\} \not\subseteq A$ ce qui donne $A \subseteq \overline{\{e_1\}} \vee \ldots \vee A \subseteq \overline{\{e_n\}}$. La proposition 16 s'applique et donne le résultat ;

$\mathcal{C}(A) = (A \cap E = \emptyset)$: $\mathcal{C}$ peut être récrite $\mathcal{C}(A) = (A \subseteq \overline{E})$ et le cas précédent s'applique avec $\overline{E}$ plutôt que E ;

$\mathcal{C}(A) = (\text{MIN}(A) > \alpha)$: $\mathcal{C}(A) = (\text{MIN}(A) > \alpha)$ peut être récrite $A \subseteq sup_\alpha$ avec $sup_\alpha = \{a \in \mathcal{A} \mid a.v > \alpha\}$ et la proposition 16 s'applique.

$\mathcal{C}(A) = (\text{MAX}(A) < \alpha)$: cette contrainte peut être récrite $A \cap supeq_\alpha = \emptyset$ avec $supeq_\alpha = \{a \in \mathcal{A} \mid a.v \geq \alpha\}$ et la proposition 17 s'applique.

□

9.4.2 Regénération

Étant donnée une base r et une contrainte $\mathcal{C}$, nous supposons dans cette section que la collection $\{A \subseteq \mathcal{A} \mid \mathcal{C}'(A) \wedge \mathcal{C}_{\mathsf{close}}(A)\}$ des motifs fermés satisfaisant la contrainte relaxée $\mathcal{C}'$ de $\mathcal{C}$ est disponible. Le but est maintenant de calculer l'ensemble $\{A \subseteq \mathcal{A} \mid \mathcal{C}(A)\}$ de tous les motifs satisfaisant $\mathcal{C}$ et pas seulement les motifs fermés, donc de regénérer à partir des motifs fermés vérifiant $\mathcal{C}'$ les motifs non fermés vérifiant $\mathcal{C}$.

Si $\mathcal{C}$ n'est pas la contrainte de fréquence, la méthode de regénération produit tous les sous-ensembles des motifs fermés contraints, mais certains de ces sous-ensembles ne satisfont pas $\mathcal{C}$. Par exemple, à la figure 9.1, tous les motifs des classes 2, 3, 4 et 5 sont regénérés, alors que

seulement ceux des classes 3 et 4 et certains des classes 2 et 5 satisfont $\mathcal{C}$. Une dernière phase de filtrage est donc nécessaire pour retirer tous les motifs générés qui ne satisfont pas $\mathcal{C}$. Cette phase peut être intégrée à l'algorithme de regénération.

9.5 Conclusion

Nous avons proposé dans ce chapitre une méthode formelle capable d'extraire des motifs contraints dans une base de données très larges. Pour cela, nous avons montré l'intérêt des notions de contrainte transposée et de relaxation de contrainte : la contrainte initiale est relaxée, puis transposée pour obtenir les motifs fermés qui définissent les classes d'équivalence contenant les motifs vérifiant la contrainte initiale. Le résultat de la recherche est fourni par une étape de regénération de motifs, à partir des ensembles maximaux des classes.

Cette méthode formelle ne nécessite pas la mise au point d'un extracteur dédié aux contextes très larges et se contente de l'adaptation d'un extracteur classique de motifs contraints en utilisant la contrainte transposée.

Conclusion

Dans cette partie, nous nous sommes intéressé à l'extraction de motifs dans les larges bases de données et aux difficultés posées par un grand nombre d'attributs devant le nombre d'objets. Nous avons présenté une nouvelle méthode d'extraction de motifs fréquents, reposant sur l'utilisation conjointe des propriétés de la connexion de Galois et de la transposition. Cette méthode tire pleinement avantage des caractéristiques géométriques de la base.

Grâce à l'introduction de la notion de contrainte transposée et de l'utilisation de la relaxation de contrainte, nous avons étendu cette approche à des contraintes autres que celle de fréquence. La définition d'un cadre formel pour la transposition de contraintes permet des extractions sous contrainte dans des contextes comportant un grand nombre d'attributs.

Outre l'intérêt pour les expériences menées en biologie et relatées à la section 8.4, le chapitre 13 dans la prochaine partie montre l'apport de cette approche pour la recherche de motifs émergent forts et leur utilisation dans des données biologiques SAGE.

Quatrième partie

Applications

Introduction

Cette partie montre l'apport de notre travail sur des problèmes réels et présente des applications menées en collaboration avec des experts des données. Les résultats obtenus aux parties II et III sont développés ici en terme de méthodes d'exploration des données et appliqués sur des problèmes réels.

Le chapitre 10 présente des usages des motifs k-libres pour la mise au point des méthodes : recherche de motifs émergents dans les larges jeux de données, règles de caractérisation et classification supervisée. Les réalisations logicielles sont développées pendant le chapitre 11. Le chapitre 12 relate une collaboration médicale sur la maladie de HODGKIN qui affecte les ganglions du système lymphatique. Nous nous intéressons à la recherche d'associations généralisées informatives pour la mise en évidence de facteurs pronostiques de cette maladie. Enfin, le chapitre 13 décrit nos expériences sur des données génomiques où nous extrayons des motifs émergents forts dans un but de caractérisation des cellules cancéreuses.

Chapitre 10

Usage des motifs k-libres

Ce chapitre détaille les usages possibles des motifs k-libres pour l'extraction de connaissances. Nous souhaitons communiquer au lecteur l'intuition qu'ils permettent aisément la réalisation de méthodes efficaces d'exploration de données.

Lorsque les données sont étiquetées par une valeur de classe, le terme de méthodes *supervisées* est employé. La classification supervisée a pour but de déterminer la classe d'appartenance d'objets non étiquetés. La caractérisation de classe [Michalski, 1983] est une tâche descriptive qui consiste à résumer de manière compacte les données à partir de leurs propriétés intrinsèques, en fonction de la valeur de classe. Par exemple, il peut s'agir de classes relatives au degré d'atteinte d'une maladie, ou de savoir si un objet « cellule » est cancéreux ou pas.

Les usages des motifs k-libres concernent d'une part la découverte de motifs émergents et de motifs émergent forts (section 10.1), dans un but de caractérisation de classe. Les motifs 1-libres et leur fermeture, constituant les motifs fermés, sont utiles pour leur calcul. La section 10.2 présente une méthode qui combine les résultats en transposition de données et l'extraction des motifs émergents forts, pour extraire ces motifs dans les larges jeux de données. D'autre part, les motifs k-libres et les règles d'association généralisées informatives peuvent être utilisés pour la caractérisation et la classification supervisée (section 10.3).

10.1 Extraction de motifs émergents et motifs émergents forts

Le travail suivant a été effectué en collaboration avec Arnaud Soulet *du GREYC. Nous avons plus particulièrement participé à la formalisation du cadre sur le calcul des motifs émergents forts à partir des motifs fermés. Ce cadre nous sera utile pour les expérimentations menées au chapitre 13. La référence [Soulet et al., 2004b] donne les détails complets de la méthode, en particulier les preuves et les expérimentations et [Soulet et al., 2005] étend la notion de motif émergent fort à toute contrainte fondée sur l'utilisation de la fréquence.*

La recherche des motifs émergents [Li *et al.*, 1999] se situe dans un contexte supervisé. Ce

sont des motifs fréquents dans une classe (une partie de la base) et infréquents dans le reste de la base. On qualifie de *taux de croissance* GR le rapport entre les fréquences dans les deux parties de la base. Ces motifs présentent une vision des différences entre deux jeux de données. Leurs usages en classification sont reconnus [Dong et Li, 1999]. Nous les avons aussi utilisés à des fins de caractérisation de classe, par exemple lors de nos expériences avec la société PHILIPS concernant la détection de plaques de silicium défectueuses [Soulet *et al.*, 2004a].

L'extraction des motifs émergents constitue un défi, car leur énumération naïve échoue rapidement. Considérée sous la forme d'une contrainte, l'émergence ne fournit pas de bonne propriété, comme la monotonie ou l'antimonotonie, voire la convertibilité. Nous avons montré que la représentation condensée des motifs fermés est aussi une représentation condensée pour les motifs émergents [Soulet *et al.*, 2004b]. Nous avons également introduit la notion de motif émergent *fort*, qui concerne les motifs qui ont le meilleur taux de croissance.

10.1.1 Motifs émergents : définition

Afin de disposer d'un contexte supervisé, nous considérons maintenant l'exemple la base de données de la table 10.1. Elle schématise un exemple de base de données génétiques, où six gènes (*i.e.* attributs) sont exprimés différemment dans quatre cellules (*i.e.* objets). Chaque gène appartient à une (et une seule) classe notée par l'attribut c_1 ou c_2.

	Classes		Attributs					
Objets	c_1	c_2	a_1	a_2	a_3	a_4	a_5	a_6
o_1	×		×	×	×			
o_2	×		×			×	×	×
o_3		×	×	×		×	×	
o_4		×		×		×	×	×

TAB. 10.1 – Exemple d'une base de données supervisée.

Les motifs émergents sont caractérisés par leur taux de croissance (nous notons r_i pour r_{c_i}, la sous base qui contient les objets de la classe c_i) :

Définition 37 (Taux de croissance) *Pour une base de données r, le taux de croissance pour la classe i d'un motif X est :*

$$GR_i(X) = \frac{|r \backslash r_i|}{|r_i|} \cdot \frac{\mathcal{F}(X, r_i)}{\mathcal{F}(X, r \backslash r_i)}. \tag{10.1}$$

Le rapport est normalisé pour comparer équitablement les taux de croissance lorsque les classes sont déséquilibrées. Un motif est dit émergent si son taux de croissance dépasse *1*. Quand

un motif est présent uniquement dans r_i (*i.e.* $\mathcal{F}(X, r\backslash r_i) = 0$), X est qualifié de *jumping emerging pattern* (JEP) et $GR_i(X) = \infty$. Dans notre exemple, le motif a_1 est deux fois présent dans la classe c_1, tandis qu'il n'est présent qu'une fois dans c_2 : son taux de croissance est *2*. Le motif $a_1a_2a_3$ est un JEP pour la première classe car il n'est pas présent dans r_2.

10.1.2 Calcul des motifs émergents à partir des motifs fermés

Nous avons vu à la section 2.2 que les motifs fermés fréquents constituent une représentation condensée des motifs fréquents, car la fréquence d'un motif est celle de sa fermeture. La même propriété concerne le taux de croissance (h est l'opérateur de fermeture Galois).

Proposition 18 *Soit X un motif ne contenant pas d'attribut de classe :*

$$GR_i(X) = GR_i(h(X))$$

Cette propriété est prouvée dans [Soulet *et al.*, 2004b].

Sur notre exemple, nous pouvons éviter de calculer le taux de croissance de a_1a_4 car il n'est pas fermé. Seul celui de sa fermeture $a_1a_4a_5$ est calculé. Les motifs fermés fréquents suffisent donc pour synthétiser l'ensemble des motifs émergents fréquents, avec leur taux de croissance : ils en constituent **une représentation condensée exacte**. Elle est exacte car le taux de croissance précis de X est donné par celui de $h(X)$.

10.1.3 Motifs émergents forts

Le nombre de motifs émergents peut être rédhibitoire pour leur utilisation. En pratique, il est judicieux de ne conserver que les plus fréquents et possédant le meilleur taux de croissance possible. Mais atteindre ces deux objectifs simultanément peut être problématique. D'un côté, si le seuil de fréquence est trop bas, les motifs découverts sont trop spécifiques (trop longs). De l'autre côté, si le seuil est trop élevé, les motifs ont un taux de croissance trop faible. Ainsi nous avons proposé les motifs émergents *forts* (*strong emerging pattern* ou SEP) qui constituent un compromis entre les exigences de fréquence et de taux de croissance.

Définition 38 (Motif émergent fort) *Un motif X, ne contenant pas l'attribut de classe c_i, est appelé motif émergent fort pour la classe i si Xc_i est fermé dans r.*

La proposition suivante indique que les motifs émergents forts ont le meilleur taux de croissance possible sur une classe d'équivalence des supports :

Proposition 19 *Soit X un motif ne contenant pas l'attribut de classe c_i. Le motif émergent fort $h(Xc_i)$ a un meilleur taux de croissance que X, i.e. $GR_i(X) \leq GR_i(h(Xc_i)\backslash c_i)$.*

Tous les motifs fermés dans r ne sont pas des motifs émergents forts. Par exemple, $a_1a_4a_5$ est fermé mais son taux de croissance vaut seulement 1 et interdit son émergence. Le motif émergent fort correspondant pour la classe c_1 est $a_1a_4a_5a_6$ ($GR = \infty$: c'est un JEP). Notons qu'il y a des JEP qui ne sont pas émergents forts : a_2a_4 est un JEP pour la seconde classe, mais le motif fort correspondant est $a_2a_4a_5$.

Comparés aux motifs émergents « simples », les motifs émergents forts ont deux avantages : ils sont faciles à calculer à partir des motifs fermés (en filtrant ceux qui contiennent une valeur de classe) et ils représentent les motifs qui ont le meilleur taux de croissance parmi les motifs de même support. Dans [Soulet *et al.*, 2005], nous avons montré que cette proposition peut être étendue à toute mesure fondée sur la fréquence, car les motifs fermés résument la fréquence de tous les motifs de leur classe d'équivalence.

10.2 Extraction des SEP dans les larges jeux de données

Dans les larges jeux de données, l'extraction des SEP est une tâche difficile car celle-ci repose sur l'extraction des motifs fermés. Similairement à l'approche que nous avons menée pour l'extraction des motifs fréquents dans de tels contextes, nous proposons dans cette section un procédé d'extraction des SEP dans les larges jeux de données à l'aide de la transposition.

Nous donnons maintenant notre méthode pour calculer les motifs émergents forts dans une base large à l'aide des étapes suivantes :

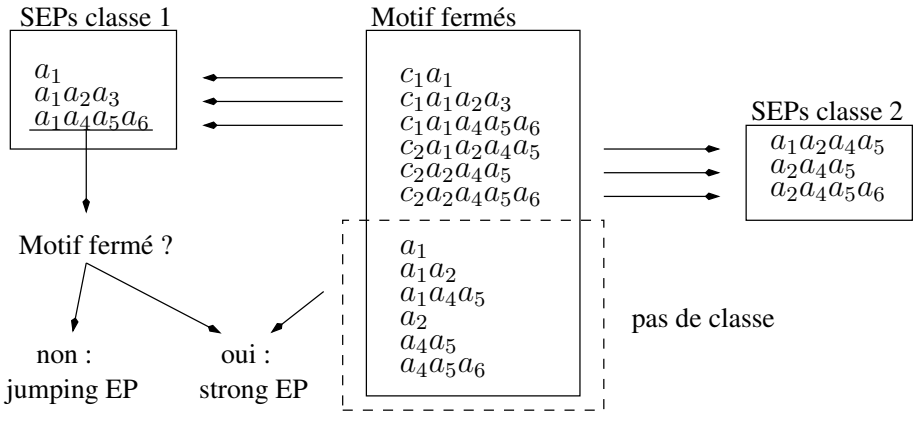

FIG. 10.1 – Processus d'extraction des motifs émergents forts à partir des motifs fermés dans l'exemple de la table 10.1.

1. extraire les concepts dans la base transposée pour obtenir les motifs fermés de la base ;
2. les SEP pour une classe sont les motifs fermés qui contiennent une valeur de classe ;

3. pour calculer les taux de croissance, il faut différencier les SEP qui sont aussi des JEP. Nous avons donné dans [Soulet *et al.*, 2004b] une nouvelle caractérisation pour les jumping emerging patterns : X est un JEP pour la classe i si et seulement si l'attribut de classe est dans la fermeture de X. Cette information est disponible dans la liste des motifs fermés.

La partie de cette méthode qui calcule les SEP à partir des fermés est schématisée à la figure 10.1. Le chapitre 13 montre l'utilisation pratique de cette méthode dans le cas des données génomiques. Les SEP sont extraits d'une matrice d'expression de 27 679 gènes [Rioult, 2004b] et analysés par les experts en biologie.

10.3 Règles d'association généralisées pour la caractérisation et la classification

Dans un contexte supervisé, les règles d'association généralisées peuvent être sources de règles de caractérisation de classe, ou de règles de classification.

10.3.1 Règles de caractérisation

Les règles d'association classiques sont souvent utilisées pour des objectifs de caractérisation, comme les règles concluant sur un attribut de la classe. Les prémisses fournissent des indications potentiellement utiles pour expliquer l'appartenance à une classe plutôt qu'à une autre. [Crémilleux et Boulicaut, 2002] exploite cette caractéristique, combinée à une propriété de minimalité de la prémisse. Nous étendons ici ce principe à l'usage des règles d'associations généralisées informatives à base de motifs k-libres.

Les avantages de ces règles généralisées informatives sont multiples :

- leurs prémisses sont robustes en présence de valeurs manquantes ;
- leur redondance est minimisée par l'utilisation de conclusions minimales ;
- leur formalisme, assimilable à une disjonction, permet des manipulations d'attributs de part et d'autre de l'« implication ».

La manipulation des attributs de part et d'autre de l'implication permet l'introduction de la notion de règle *positive* ou *négative*[6] et illustre la souplesse de ces règles. Supposons par exemple que $X \rightarrow \vee Y$ est exacte. Si Y contient un attribut de classe ($Y = Y'c_i$), on obtient une règle positive $X\overline{Y'} \rightarrow c_i$. Cette règle caractérise pour la classe c_i les objets contenant X mais aucun attribut de Y'. Si l'attribut de classe est présent dans X ($X = X'c_i$), on obtient la règle négative $X'\overline{Y} \rightarrow \overline{c_i}$. Elle caractérise des objets qui n'appartiennent pas à c_i.

[6] [Antonie et Zaïane, 2004a, Antonie et Zaïane, 2004b] définissent les règles négatives en limitant la taille de la conclusion à un attribut.

Les règles d'association généralisées positives et négatives sont utilisées ci-dessous pour la construction de règles de classification.

10.3.2 Classification supervisée

Il existe de plusieurs méthodes pour effectuer de la classification supervisée à partir d'associations et nous avons choisi CMAR[7] [Li *et al.*, 2001, Li, 2001] (Classification based on Multiple class-Association Rules) pour nos développements sur les motifs k-libres. CMAR est l'une des propositions les plus récentes et abouties sur la classification à base de règles. En particulier, cette méthode définit très précisément les procédures de sélection de règle. Elle utilise à l'origine des règles d'association classiques ($k = 1$). La redondance est évitée en ne conservant que les règles à prémisse minimale. Elles sont ensuite pondérées par une mesure de χ^2. Un dernier élagage de règles fonctionne suivant le principe de couverture : on ne garde que les premières règles qui couvrent tous les objets d'apprentissage. Un vote permet de classer un nouvel objet.

Nous avons implémenté une méthode de classification CMAR qui combine l'utilisation des règles d'associations généralisées informatives positives et négatives définies à la section précédente. Ce travail est en cours de développement mais montre des résultats encourageants. Le tableau 10.2 indique les scores comparés de notre méthode avec les méthodes classiques (C4.5 [Quinlan, 1993], CBA [Liu *et al.*, 1998a] et CMAR) sur quelques bases de référence de l'UCI [Blake et Merz, 1998]. Nos expériences montrent que 3 est une valeur intéressante pour la profondeur des règles. Au delà, les scores ne sont peu améliorés.

Il reste un travail de réglage à mener pour adapter à ces règles généralisées les mesures et les modes de sélection classiques en classification et en particulier pour gérer les règles négatives. De plus, les performances de notre classifieur se dégradent lorsqu'il y a beaucoup de classes. Cette méthode de classification représente malgré tout un usage original des motifs k-libres et de leurs fermetures.

Avec Frédérick Houben (GREYC) qui travaille sur la découverte automatique des structures formelles des langues à partir de corpus brut [Houben, 2004], nous avons mis cette méthode de classification en pratique [Houben et Rioult, 2005]. Les mots sont étiquetés par leur position dans la phrase et selon qu'il s'agit d'un mot vide ou d'un mot plein [Houben, 2004]. Les mots vides sont les mots grammaticaux (déterminants, prépositions, pronoms) ou des auxiliaires. Les mots pleins sont les autres mots.

Un exemple de règle positive obtenue indique que si le type du mot étudié est vide, qu'il est précédé d'un signe de ponctuation, qu'il est suivi d'un mot de type indéterminé et qu'il influe sur les terminaisons alors on peut conclure que le mot est de la classe « pronom personnel ». Un

[7] Le lecteur intéressé se référera avec avantage à l'excellente page de Frans Coenen, `http://www.csc.liv.ac.uk/~frans/KDD/Software/CMAR/cmar.html` qui propose de nombreux détails et une implémentation en `java`.

Base	attr.	classes	objets	C4.5	CBA	CMAR	`k-miner`
Crx	15	2	690	84,9	84,7	84,9	82,3
Hepatic	19	2	155	80,6	81,8	80,5	81,8
Pima	8	2	768	75,5	72,9	75,1	73,3
Tic-tac	9	2	958	99,4	99,6	99,2	95,4
Iris	4	3	150	95,3	94,7	94	94,4
Wine	13	3	178	92,7	95	95	87
Lymph	18	4	148	73,5	77,8	83,1	85,5
Anneal	38	5	898	94,8	97,9	97,3	74,2
Auto	25	7	205	80,1	78,3	78,1	78,3
Glass	9	7	214	68,7	73,9	70,1	66
Zoo	16	7	101	92,2	96,8	97,1	78,3
Moyenne	-	-	-	85,24	86,67	86,76	81,50

TAB. 10.2 – Performances (%) de classification avec les règles d'association généralisées.

exemple de règle négative indique que si le mot courant est de type plein et que le mot suivant n'est pas de type plein, alors le mot courant ne peut pas être un déterminant. En fait, cette règle est naturelle : un déterminant est un mot vide par définition et il est suivi d'un adjectif ou d'un nom, c'est-à-dire de mots pleins, dans la plupart des cas.

Ces règles, parmi d'autres, sont donc en accord avec les connaissances que nous avons du domaine et confirment que notre démarche est cohérente. Les perspectives linguistiques de cette méthode sont encourageantes.

10.3.3 Discussion

Les méthodes de classification à base de règles nécessitent un long et complexe travail de réglage et de mise au point. Afin d'éviter le sur-apprentissage, on s'intéresse pour un taux de classification à produire les règles les plus générales.

Les conclusions des règles généralisées étant des disjonctions, leur minimalité assure la généralité par rapport à leurs prémisses et pallient le problème de la redondance. Néanmoins, ces règles n'ont pas de propriété simple relativement à la minimalité de leurs prémisses. Mais nous pensons que la bordure négative des k-libres offre une perspective prometteuse pour la définition de règles basées sur des minimaux non k-libres.

Cette bordure fournit un réservoir pour obtenir les motifs à partir desquels des règles de classification peuvent être construites. Elle rassemble les motifs minimaux non k-libres ; selon le sens de spécialisation, ce sont donc les premiers motifs au sein desquels apparaît une corrélation. Par définition de la bordure négative, leurs sous ensembles sont libres et exempts de corrélation.

Illustrons les propriétés d'un motif $Z \in \mathcal{B}d^{-}(k-libres)$. Il existe pour Z un découpage en XY tel que la règle $X \rightarrow \vee Y$ est exacte dans la base. Les règles de classification positives ou négatives peuvent être obtenues par ce procédé. Elles possèdent des propriétés de minimalité, non pas sur la prémisse et la conclusion, mais sur leur union Z. Ce motif est le plus petit qui contienne l'association entre X et Y. L'intérêt de cette formalisation est que la définition de cette bordure ne fait appel à aucun paramètre de seuil, par exemple de fréquence ou de confiance. Remarquons que la nécessité de ces seuils pour la fouille de données est un reproche régulièrement formulé.

Chapitre 11

Développements logiciels

Ce chapitre présente les outils logiciels que nous avons réalisés pour mener notre travail de recherche. La section 11.1 donne les principes généraux de développement et la section 11.2 précise les outils utilisés. Le processus de fouille de données est ensuite découpé selon le flux d'information : les sections 11.3 à 11.5 détaillent les méthodes de prétraitement, les algorithmes d'extraction de motifs et les procédés de post-traitement. La section 11.6 livre quelques perspectives.

11.1 Principes généraux de développement

Nous avons développé nos outils logiciels au fil de l'avancée de nos travaux plus théoriques. Cela signifie que nous avons réalisé quand cela était nécessaire les prototypes dont nous avions besoin pour valider nos idées à l'échelle. Les chaînes de traitement sont donc sous forme de scripts élémentaires qui réalisent le prétraitement, l'extraction de motifs et l'interprétation. Nous avons toujours cherché à respecter des compromis entre la performance, le temps de développement, l'utilisation du matériel disponible et le besoin de solutions dédiées. Le domaine des bases de données incomplètes a en particulier nécessité l'adaptation des algorithmes connus pour la mise au point de méthodes dédiées.

La difficulté reconnue de la fouille de données concerne l'extraction des motifs. Cet aspect a concentré une part importante de nos efforts pour réaliser des programmes efficaces. Les outils théoriques pour la fouille de données concernent les manipulations « élémentaires » d'ensembles de motifs : minimaux, maximaux, complémentaires, traverses minimales. Assemblées, ces briques logicielles permettent d'implémenter une large variété d'algorithmes et de gérer un processus d'extraction de connaissance du début à la fin. Nos scripts et prototypes forment un ensemble homogène sur lequel toute modification peut être entreprise. Cet aspect confère une grande liberté dans la mise au point.

Enfin, pour éviter de surcharger les algorithmes d'extraction de motifs, nous avons mis au

point un programme qui étiquette les motifs selon une base de données. Il permet par exemple, à partir de motifs 1-libres, de calculer le concept correspondant, de le mesurer, de l'écarter selon certains critères. Fonctionnant sous forme de flux et ne consommant aucune ressource, il évite de stocker des fichiers importants de résultats complets. L'extraction de motifs étant la tâche la plus pénalisante, elle n'a lieu qu'une fois et ne doit pas être surchargée par des opérations dispensables. Avec cet outil de mesure, les calculs coûteux en espace sont reportés en aval de la chaîne de traitement.

Nos algorithmes d'extraction sont utilisés régulièrement au GREYC au cœur de processus de fouille de données [Houben, 2004, Durand *et al.*, 2004, Durand *et al.*, 2003, Hébert et Crémilleux, 2005]. Jusqu'ici, les limitations pour les expériences n'ont pas concerné les temps d'exécution, mais plutôt les ressources en mémoire et en espace disque. En effet, sous `Linux` 32-bits, l'adressage mémoire est limité à 3 GB, car chaque processus dispose de l'adressage complet possible (4 GB) et réserve 1 GB pour la gestion de l'espace. D'autre part, les fichiers de résultats pour l'extraction de motifs font couramment plusieurs giga-octets. Nous n'avons pas utilisé de techniques de compression, mais largement mis à profit les enchaînements de processus grâce à des *pipes*, pour conserver le minimum de résultats intermédiaires.

11.2 Outils de développement utilisés

Les outils utilisés sont libres et disponibles sur tous les systèmes d'exploitation. Lorsque les besoins d'efficacité et de complexité de conception le nécessitent, nous utilisons `C++` et le compilateur `g++`. Pour les traitements de flux (données ou motifs), nous utilisons les outils standards d'`UNIX` : `sed`, `awk`, `bash`. Pour assurer la portabilité, l'ensemble de ces programmes est regroupé au sein d'une distribution `GNU`, ce qui permet de l'installer sur n'importe quelle machine.

11.2.1 C++

Le `C++` s'impose dès lors que l'on recherche la performance. Pour le calcul des motifs, cet argument est déterminant. Le modèle objet de `C++`, sa librairie standard et sa généricité permettent d'écrire du code performant de haut niveau. `java` fournit le même pouvoir d'abstraction, mais n'a pas la même réputation d'efficacité. De plus, lorsqu'une application `java` est lancée, il faut indiquer à la machine virtuelle la quantité de mémoire nécessaire, ce qui soumet le processus à un paramètre complexe à estimer.

La documentation des classes est réalisée avec `doxygen`[8]. Dans les sources `C++`, il suffit d'introduire un commentaire avec la notation `///` plutôt que `//` (ou `/*** ... ***/` plutôt que `/** ... **/`). `doxygen` génère une arborescence `html` avec la hiérarchie de classe et les index

[8] http://www.doxygen.org

complets des variables et des fonctions. Les formats `XML`, LaTeX, `RTF`, `MAN` sont également disponibles.

11.2.2 `sed`, `awk`, `bash`

Les outils standards d'`Unix` traitent les flux très simplement et efficacement. L'éditeur `sed` permet, à l'aide de commandes très simples déclenchées par des expressions régulières, d'effectuer des traitements élémentaires sur les lignes et les mots.

`awk` considère également les fichiers sous la forme d'une série de lignes. Un script se compose d'une partie d'initialisations, d'une partie de traitement de chaque ligne symbolisé par un sélecteur de choix utilisant des expressions régulières et termine par une section finale. `awk` est particulièrement adapté à des traitements simples sur des structures tabulaires ou des listes de motifs.

Enfin, `bash` permet d'appeler les différents scripts et programmes. Sa faible efficacité n'est pas un handicap lorsqu'il est réservé pour les tâches d'enchaînement de haut niveau.

11.2.3 Portabilité

Dans le but de garantir un large accès à nos réalisations, la portabilité est assurée par le système de distribution `GNU`. Il fonctionne par la génération automatique de `Makefile` et libère le programmeur de la lourde gestion des dépendances. Des procédures d'installation pour les systèmes compatibles `GNU` sont offertes.

11.3 Pré-traitement des données

Tout processus de fouille de données commence par la préparation des données : constitution d'une table par jointure de relations (effectué via un système de gestion de bases de données), sélection des attributs, correction de format, etc. Pour les méthodes à base de motifs, une table attribut/valeur est le format de départ. Des outils simples d'analyse univariée ont été développés, ainsi que des procédures de binarisation des attributs continus selon un nombre déterminé d'intervalles de même population (pour quatre intervalles, c'est la méthode des quartiles).

Les contextes booléens sont décrits par un format classique : chaque objet est codé sur une ligne qui liste les numéros de chaque attribut. Les valeurs manquantes sont précédées d'un signe « - ». Pour notre exemple récurrent de la table 5.1 page 75, nous obtenons le fichier suivant :

```
1 3 5
2 3 5
1 3 -5 -6 -7
1 4 6
```

```
2 3 6
-1 -2 3 6
1 4 7
2 -3 -4 7
```

Enfin, pour effectuer nos transformations de bases de données, nous avons réalisé des utilitaires qui calculent les bases opposée et transposée.

11.4 Algorithmes d'extraction de motifs

Nous décrivons ici les deux algorithmes d'extraction de motifs dans les bases de données incomplètes que nous avons réalisés : `MVminer`, qui extrait par niveaux les motifs δ-libres et les motifs fermés (voir chapitre 2) et `MV-k-miner`, qui calcule les motifs k-libres (cf. le chapitre 3 et la partie II). L'algorithme `MV-k-miner` est donné à la section 6.4.

11.4.1 `MVminer`

Le prototype exécutable `acminer` [Boulicaut *et al.*, 2000] nous a permis de valider nos résultats. Celui-ci a été développé au LIRIS par Arthur Bykowski. L'article sus-cité donne des détails précis sur l'algorithme `matched` de calcul de la δ-fermeture employé par `acminer`.

Après divers essais (tables de hachage, arbres équilibrés), nous avons décidé de stocker les motifs dans des arbres de préfixe (voir figure 11.1). Chaque nœud contient un numéro d'attribut et deux pointeurs sur un fils et un frère. Un motif est symbolisé par une feuille et la liste de ses attributs est constituée par l'énumération des valeurs depuis la racine. Sur la figure, la feuille entourée stocke le motif $a_1a_4a_7$. Cette structure réalise un compromis entre la facilité de programmation et des impératifs de parcours systématique de tous les motifs d'un niveau.

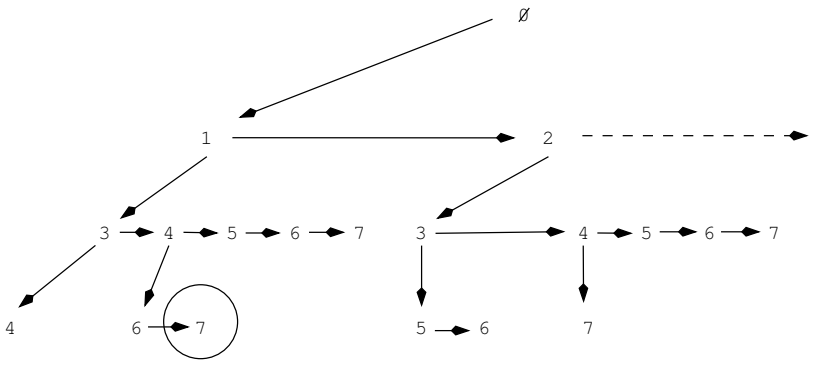

Fig. 11.1 – Arbre de préfixe.

Dans ce type d'arbre, les temps d'insertion et de recherche sont moins efficaces que les tables

de hachage et les arbres équilibrés. Mais pour nos problèmes, le hachage par exemple a peu de sens car le nombre d'attributs est généralement localement réduit. D'autre part, les structures équilibrées sont appréciables pour l'efficacité des accès. Mais le parcours d'un niveau du treillis (selon des motifs de longueur constante), impliqué dans les phases de génération de candidats, est complexe. Dans les arbres de préfixe, ces opérations sont triviales et la génération est effectuée par fusion d'une feuille et sa sœur, qui diffèrent seulement d'un pointeur.

Un unique arbre de préfixe gère l'ensemble des motifs δ-libres. Pour les phases d'examen de la base, qui sont nécessaires au calcul des supports exacts, cet arbre est parcouru et la base est interrogée pour chaque motif. De cette manière, la fermeture d'un motif est disponible avant de passer au motif suivant dans l'arbre. Le programme connaît à ce moment l'intension et l'extension : `MVminer` peut calculer les concepts[9] sans sur-coût de ressources. Une fois le concept fourni à l'utilisateur, la fermeture et le support n'ont pas besoin d'être stockés : seul l'arbre des motifs δ-libres est conservé en mémoire.

`MVminer` ne respecte donc pas l'usage qui veut que chaque objet de la base soit lu une unique fois et utilisé pour tous les candidats, pour limiter les accès à la base de données. L'expérience montre que lorsqu'il y a un « faible » nombre d'objets (quelques milliers), les deux approches ont des performances comparables. Notre technique permet de disposer d'un extracteur souple, qui peut indifféremment extraire les motifs δ-libres et leurs fermetures ou les concepts, sans consommation de ressources supplémentaires.

D'une façon plus générale sur l'implémentation des algorithmes, le lecteur intéressé peut se référer aux travaux de Christian Borgelt [Borgelt et Kruse, 2002, Borgelt, 2003].

`MVminer` permet également de calculer différents types de mesures sur les concepts calculés. Ces mesures servent pour la méthode de classification non supervisée `Ecclat` [Durand, 2004], développée par Nicolas Durand du GREYC, et utilisée au chapitre 12 pour la mise en évidence de groupes pronostiques de patients pour la maladie de Hodgkin. `Ecclat` ordonne et choisit les concepts selon cette mesure d'intérêt et construit un clustering avec chevauchement.

Les mesures des concepts concernent leur homogénéité, qui est le rapport entre l'aire du concept et l'aire définie par tous les objets du concept. Lorsque chaque objet est décrit par un même nombre d'attributs, cette mesure est équivalente au nombre d'attributs du concept. Nous avons donc proposé l'emploi d'une homogénéité selon les attributs, qui est égale au rapport de l'aire du concept et de l'aire définie par ses attributs. Ces mesures sont directement intégrées dans `MVminer`.

[9] Cette technique d'extraction de concepts à partir des motifs libres produit de nombreux doublons qu'il faut supprimer. Elle montre ses limites lorsqu'il y a beaucoup de concepts à trier.

11.4.2 `MV-k-miner`

Ce prototype implémente l'algorithme éponyme d'extraction par niveaux des motifs k-libres et de leur fermeture généralisée dans une base incomplète (voir algorithmes 4 et 5 section 6.4). Il utilise le même principe d'arbre de préfixe que développé précédemment. Cependant, à la différence de `MVminer`, la phase de calcul des supports exacts est effectuée en une seule passe sur la base.

L'extraction des k-libres dans une base de données est effectuée par un algorithme par niveaux qui exploite l'antimonotonie de la contrainte de k-liberté. Rappelons que pour MUSHROOM, le temps d'exécution minimum est obtenu pour $k = 2$ (cf. tableau 3.1 page 48). Lorsque k augmente, le calcul des différentes bornes pour la fréquence se complexifie. Cependant, les bases possèdent rarement des corrélations plus profondes que $k = 5$ et obtenir les motifs k-libres est raisonnable (quelques minutes de calcul).

La tâche difficile concerne le calcul des fermetures. Celle-ci est effectuée à l'aide des traverses minimales de longueur bornée (voir annexe 1). Cette étape détermine clairement la durée de l'exécution. Il faut cependant noter qu'en pratique la limitation introduite par le calcul des fermetures est peu pénalisante. Pour appliquer notre méthode dédiée aux valeurs manquantes, qui calcule les règles correctes entre la base et son opposée, nous avons besoin de règles comportant sensiblement autant d'attributs en prémisse qu'en conclusion. On peut considérer que la valeur $k = 4$ suffit en pratique. Au-delà de cette valeur, le nombre de motifs k-libres stagne. Dans ces conditions, le temps de calcul des fermetures est raisonnable.

11.5 Traitement des données et des motifs

Pour obtenir les règles correctes en présence de valeurs manquantes, le traitement des règles d'association généralisées informatives issues de la base opposée requiert un développement particulier. Pour les comparer aux règles de la base initiale, elles nécessistent d'être inversées et leurs prémisses doivent être fréquentes dans la base originale. On ne conserve que les règles reconstruites dont les conclusions sont minimales.

Nous avons pour cela défini un modèle d'arbre préfixe dont chaque nœud peut lui-même contenir un arbre : à chaque motif k-libre situé dans l'arbre est associé l'arbre constitué par les motifs de sa fermeture. Cette structure d'arbre préfixe est assez générique pour traiter des ensembles de motifs selon des contraintes de minimalité et de maximalité et pour les parcourir suivant des ordres variés. Seules les méthodes d'insertion doivent être adaptées.

11.6 Conclusion - perspectives

Pour l'extraction des règles généralisées informatives, la difficulté centrale reste le calcul des fermetures généralisées. Des algorithmes dédiés au calcul des traverses minimales de longueur bornée sont nécessaires et plus particulièrement sur de petites valeurs de k, par exemple $k \leq 5$.

Un effort logiciel important pour le développement de solutions en fouille de données a permis une grande souplesse pour les expérimentations et une autonomie importante. Nos outils permettent d'envisager une large panoplie de méthodes d'extraction de connaissance et d'exploration de données. Ces méthodes sont actuellement prises en charges par des briques logicielles, enchaînées par des scripts `bash`. Nos développements récents utilisent une visualisation graphique des chaînes de traitement, qui permet de réguler le déroulement des tâches.

Afin de faciliter la mise en œuvre de processus d'extraction de connaissances, nous avons conçu une maquette pour un gestionnaire de flux de données. Le processus est schématisé par un graphe réalisé avec `JgraphPad`, un logiciel de conception de flux. Notre utilitaire convertit ce graphe codé en XML vers un fichier `Makefile`. L'utilitaire `make` contrôle l'exécution en gérant les dépendances.

La figure 11.2 présente un exemple de processus. Il s'agit de la binarisation pour la base `iris`. Ce type de processus est généralement complexe, car il constitué de plusieurs utilitaires à enchaîner, qui génèrent des fichiers intermédiaires. Le graphe contient sept boîtes, identifiées par un nom de processus. Chaque boîte comporte un nombre variable de ports, qui fonctionnent en entrée ou en sortie, suivant qu'une flèche de flux y pénètre ou en sort. Les flèches sans source sont les dépendances du processus, les flèches sans destination sont les cibles. La transformation de l'exemple de la figure en `Makefile` est à la table 11.1.

À ce stade, l'outil est encore limité. D'une part, les exécutions sont réalisées pour chaque tâche élémentaire indépendamment des autres. Les optimisations par enchaînement de `pipe` sont impossibles et les fichiers intermédiaires sont stockés. D'autre part, le langage graphique utilisé est trop simple d'un point de vue conceptuel. Il manque de possibilités d'itérations et de factorisation des opérations qui rendraient les réalisations moins fastidieuses.

Cependant, cette interface apporte souplesse, interactivité et convivialité dans la définition des processus de fouille de données. Les changements de paramètres sont aisés et appliquer un processus complet sur différentes bases est très simple. De plus, cette visualisation du processus peut être mémorisée, facilement éditée et procure une représentation graphique du procédé plus lisible que les scripts `bash`.

Le développement de solutions pour la gestion graphique de processus doit être poursuivi pour améliorer les limites de notre maquette. Pour cela, une intégration de notre ensemble logiciel est

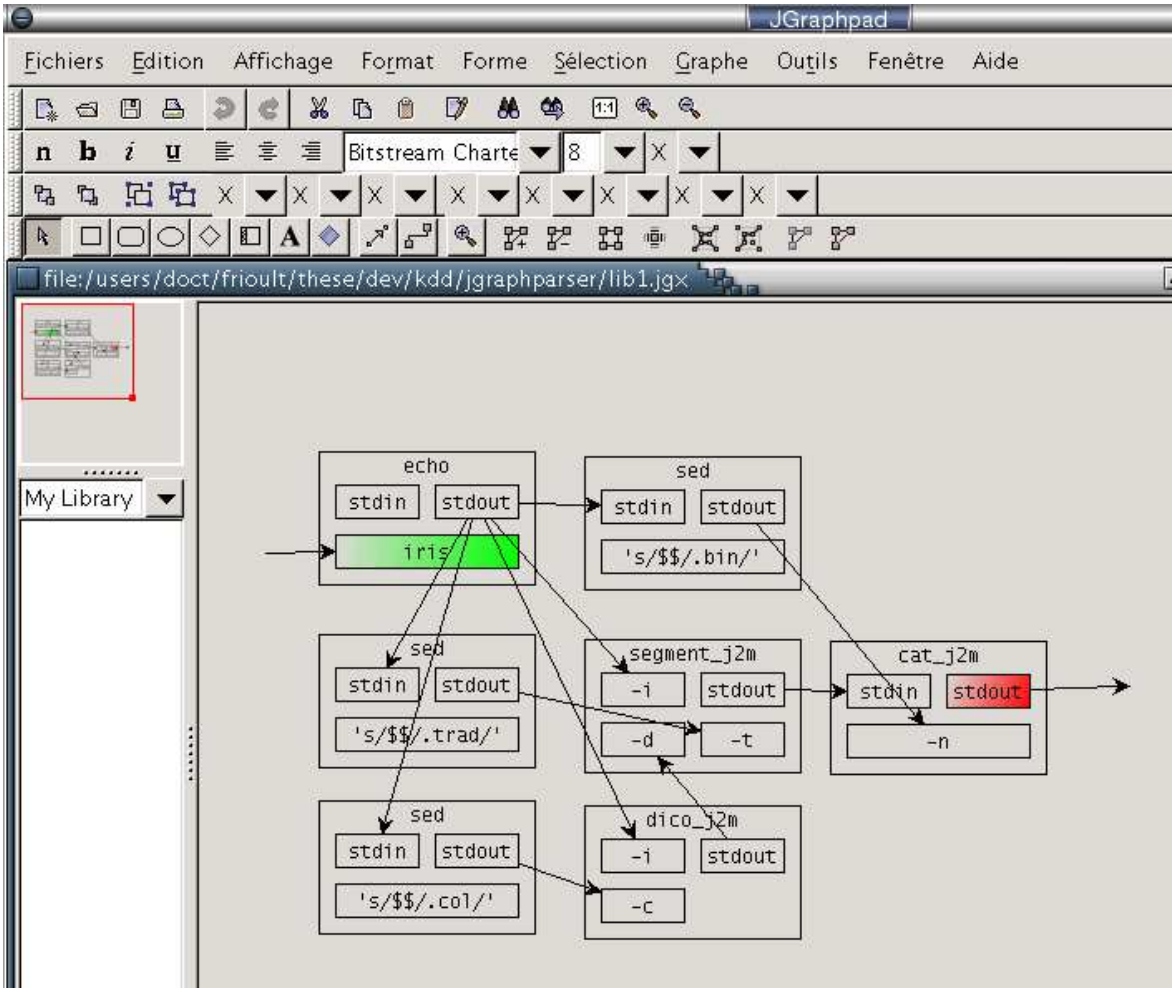

FIG. 11.2 – Édition graphique de processus avec `Jgraph`.

en cours dans la plate-forme `Lingua Stream` [10]. Développée au GREYC, `Lingua Stream` est dédiée au langage naturel et est très aboutie d'un point de vue de l'interface avec l'utilisateur. Le format pour les données et les processus est `XML`, et a montré son utilité pour la fouille de données [Meo et Psaila, 2003].

[10] http://www.linguastream.org

```
all : cat_j2m285
echo15 : iris
        echo iris > echo15
sed57 : echo15
        sed 's/$$/.col/' < echo15 > sed57
dico_j2m99 : echo15 sed57
        dico_j2m -i echo15 -c sed57 > dico_j2m99
segment_j2m155 : echo15 dico_j2m99 sed240
        segment_j2m -i echo15 -d dico_j2m99 -t sed240 > segment_j2m155
sed197 : echo15
        sed 's/$$/.bin/' < echo15 > sed197
sed240 : echo15
        sed 's/$$/.trad/' < echo15 > sed240
cat_j2m285 : segment_j2m155 sed197
        cat_j2m -n sed197 < segment_j2m155 > cat_j2m285
clean :
        rm -f echo15 sed57 dico_j2m99 sed240 segment_j2m155 sed197
```

TAB. 11.1 – `Makefile` pour le processus de la figure 11.2.

Chapitre 12

Recherche de facteurs pronostiques pour la maladie de Hodgkin

Nous rapportons dans ce chapitre des expériences relatives à la mise en évidence de facteurs pronostiques pour les stades localisés de la maladie de Hodgkin. Il s'agit d'un cancer des ganglions du système lymphatique. Le Docteur Michel Henry-Amar, responsable de l'unité de recherche clinique du centre de lutte contre le cancer François Baclesse à Caen, est un expert de cette affection.

Ce chapitre concerne une illustration pratique de nos résultats de la partie II, qui traite de l'extraction de connaissances dans des bases de données comportant des valeurs manquantes. Plus particulièrement, nous cherchons dans ces données des règles d'association généralisées correctes. La section 12.1 présente brièvement la maladie de Hodgkin et les thérapeutiques utilisées sont expliquées à la section 12.2. La section 12.3 décrit les données mises à notre disposition et les caractéristiques des valeurs manquantes rencontrées. La section 12.4 conclut ce chapitre et liste les associations intéressantes que nous avons mises en évidence à partir de ces données.

12.1 Maladie de Hodgkin

Le système lymphatique joue un rôle primordial pour le système immunitaire. Il est composé de ganglions lymphatiques reliés par des vaisseaux. Ce système véhicule la lymphe, les anticorps et les cellules spécialisées dans la défense immunitaire. Les ganglions filtrent le liquide lymphatique. On les retrouve organisés en chapelet le long des gros troncs vasculaires. Ils sont palpables au niveau du cou, des aisselles, de l'aine ; les examens radiologiques sont nécessaires pour évaluer leur présence au niveau du thorax ou de l'abdomen. Lorsqu'ils sont envahis, ils peuvent présenter des dimensions variables, allant de celles d'une tête d'épingle à celles d'une orange.

Les cellules cancéreuses colonisent les organes lymphatiques (ganglions, rate) où elles s'agrègent

en amas de dimensions variables. La maladie (ou lymphome) de HODGKIN, du nom du médecin qui l'a décrite à Londres en 1832, est une prolifération maligne des cellules du système immunitaire, appelée lymphome malin. On distingue les lymphomes malins non hodgkiniens et les lymphomes malins hodgkiniens.

Cette maladie est une pathologie peu fréquente, deux à trois nouveaux cas par an pour 100 000 habitants. Grâce aux données des registres français de cancer (réseau Francim), son incidence a été estimée à de $2,1$ pour la femme et de $2,6$ pour l'homme en 2000, soit environ 1400 nouveaux cas par an. Elle affecte principalement des jeunes adultes, entre 16 et 30 ans. Cette maladie est au 23^{e} rang des décès par cancer et représente 0,1 % de la mortalité par cancer en France.

Nous avons pu disposer des données du groupe lymphome de l'*European Organization for Research and Treatment of Cancer* (EORTC) qui conduit des essais cliniques depuis 1964. Les données concernent 6651 patients adultes atteints par la maladie de HODGKIN. Ces malades ont été inclus dans neuf essais cliniques consécutifs [Cosset *et al.*, 1992]. Leur objectif était, au moyen de la comparaison de plusieurs prises en charge thérapeutiques, de traiter les patients au moindre « coût » : en maximisant la probabilité de guérison, tout en minimisant le risque d'effets secondaires [Tubiana *et al.*, 1989]. Aujourd'hui, les taux de survie à dix ans des patients atteints d'un stage limité de la maladie dépassent 90 %. Ils atteignent 80 à 85 % dans les stades disséminés. La recherche de facteurs pronostiques reste d'actualité, le but étant de limiter le recours à des thérapies agressives pour les patients ayant un pronostic *a priori* favorable.

Parmi ces neuf études, trois ont été retenues par l'expert (le D^{r} HENRY-AMAR), les essais H7-H8-H9 conduits entre 1988 et 2004 parce que basés sur la même définition de groupes pronostiques *a priori*. Au total, 4058 malades atteints d'un stade localisé de la maladie ont été inclus, dont 3904 ont pu être utilisés dans le cadre de ce travail. Ces derniers avaient été suivis suffisamment longtemps pour permettre une analyse pronostique concernant les échecs (progression de la maladie ou échec primaire, rechute), ceux-ci survenant dans leur grande majorité au cours des trois premières années. Les premiers résultats de ces essais ont été publiés [Noordijk *et al.*, 1994, Hagenbeek *et al.*, 2000, Fermé *et al.*, 2000, Noordijk *et al.*, 2005].

12.2 Traitements

Au cours de la période 1964-1988, la prise en charge thérapeutique de cette maladie a beaucoup évolué. Le groupe lymphome de l'EORTC s'est appuyé sur l'expérience acquise pour définir deux groupes de malades à pronostic *a priori* différent : un groupe « favorable » et un groupe « défavorable », pour lesquels la stratégie thérapeutique appliquée est différente.

Cinq paramètres démographiques, cliniques et biologiques ont été associés pour définir ces deux groupes : l'âge (< 50 ou ≥ 50 ans), le nombre d'aires ganglionnaires envahies (1 à 3

contre 4 ou 5), l'envahissement médiastinal (médiastin indemne ou envahissement limité [ratio médiastin / thorax (MTR) < 35 %] contre $MTR \geq 35$ %), la présence de signes généraux (SG) (fièvre, sueurs nocturnes, perte de poids de plus de 10 % au cours des 6 derniers mois) et la vitesse de sédimentation globulaire (VS) [Tubiana *et al.*, 1989]. Un patient est inclus dans le groupe `défavorable` si son âge est supérieur à 50 ans, ou si $MTR \geq 35$ %, ou s'il présente 4 ou 5 aires ganglionnaires envahies, ou absence de SG et $VS \geq 50mm/1^e heure$, ou présence de SG et $VS \geq 30mm/1^e heure$. Tous les autres patients sont inclus dans le groupe `favorable`.

12.3 Description des données et analyse des valeurs manquantes

Les données sont décrites par 30 attributs multi valués : deux indications démographiques (sexe, âge), 18 résultats d'examens cliniques et 10 biologiques. Chaque attribut est transformé en attributs booléens sous le contrôle de l'expert. Par exemple, pour l'âge, deux attributs ont été définis : moins et plus de 50 ans. Au total, nous avons construit 78 attributs booléens.

La table 12.1 liste pour chaque attribut le procédé de discrétisation et la fraction de valeurs manquantes. Les 3904 patients concernent trois séries indépendantes (les trois essais $H7$-$H8$-$H9$). Dans la première série (816 objets soit 21 % du total), les données concernant les ganglions cervicaux hauts n'ont pas été recueillies. Ils ont été confondus avec les ganglions cervicaux bas. Les données relatives à leur envahissement et dimensions sont donc manquantes. Il s'agit d'un problème classique de fusion de données relatives à plusieurs protocoles. Dans la situation présente, il explique le nombre important de valeur manquantes sur les attributs concernés.

Les valeurs manquantes sur les autres attributs, entre 2 et 6 %, sont classiques des données médicales et caractérisent les examens qui n'ont pas été pratiqués ou dont les résultats n'ont pas été transmis. Les éléments figurés du sang (polynucléaire, lymphocytes, monocytes) sont tous mesurés à partir du même échantillon de sang. Leur absence simultanée indique que les analyses correspondantes n'ont pas été effectuées.

Au total, les données sont complètes pour 2004 patients. Dans cette configuration, les méthodes consistant à écarter les objets incomplets ont peu de sens. En particulier, elles excluraient les patients de l'essai $H7$. Il y a au total 11 995 relations attribut/objet manquantes, soit 4 % de valeurs manquantes.

Les données fournies par l'expert sont rarement complètes pour la dimension des ganglions. En effet, lorsque le ganglion n'est pas envahi, le praticien ne mesure pas sa dimension. Ce type de données incomplètes suit le modèle *missing at random* (cf. section 5.1.1 page 70) car elles dépendent de valeurs observées. Lors de premières expériences, ces données ont été considérées comme manquantes, ce qui occasionnait un taux de valeurs manquantes sur les attributs de dimensions qui avoisinait les 75 %. Dans ce cas, seules les données de 11 patients étaient complètes. Dans les expériences suivantes, elles ont été remplacées par une dimension de ganglion nulle.

attribut	nom	binarisation	valeurs manquantes
`age`	âge (années)	$\leq 50,\ > 50$	0 %
`sexe`	sexe	homme, femme	_ %
`chd`	cervical haut droit envahi	oui, non	23 %
`chddim`	dimension du chd (mm)	$0,]0, 10],]10, 50], > 50$	25 %
`chg`	cervical haut gauche envahi	oui, non	23 %
`chgdim`	dimension du chg (mm)	$0,]0, 10],]10, 50], > 50$	25 %
`cbd`	cervical bas droit envahi	oui, non	2 %
`cbddim`	dimension du cbd (mm)	$0,]0, 10],]10, 50], > 50$	7 %
`cbg`	cervical bas gauche envahi	oui, non	2 %
`cbgdim`	dimension du cbg (mm)	$0,]0, 10],]10, 50], > 50$	8 %
`axd`	auxiliaire droit envahi	oui, non	2 %
`axddim` 31	dimension de l'axd (mm)	$0,]0, 10],]10, 50], > 50$	3 %
`axg`	auxiliaire gauche envahi	oui, non	2 %
`axgdim` 31	dimension de l'axg (mm)	$0,]0, 10],]10, 50], > 50$	4 %
`med`	médiastin envahi	oui, non	2 %
`hild`	hile droit envahi	oui, non	2 %
`hilg`	hile gauche envahi	oui, non	2 %
`mtr`	rapport médiastin / thorax	$< 35, [35, 45[, \geq 45$	7 %
`ext`	extension par contiguïté	oui, non	2 %
`sg`	signes généraux	oui, non	2 %
`vs`	vitesse de sédimentation ($mm/1^e heure$)	$\leq 30,]30, 49], > 49$	3 %
`hb`	hémoglobine ($g/dl \times 10$)	$\leq 105, > 105$	3 %
`gb`	globules blancs ($\times 100$)	$\leq 60,]60, 120], > 120$	3 %
`polyn2`	polynucléaires neutrophiles	$\leq 3000, > 3000$	9 %
`lympho2`	lymphocytes	$\leq 1500, > 1500$	8 %
`mono2`	monocytes	$\leq 200, > 200$	9 %
`plaq`	plaquettes ($\times 1000$)	$\leq 100,]100, 600], > 600$	3 %
`pa`	phosphatases alcalines (UI/l)	$\leq 200, > 200$	6 %
`ldh`	lactico deshydrogénase (UI/l)	$\leq 300, > 300$	6 %
`histo2`	histologie	4 types	0 %

TAB. 12.1 – Attributs pour l'étude de la maladie de HODGKIN.

12.4 Expériences

12.4.1 Recherche d'associations généralisées informatives

Lors d'expériences précédentes [Rioult et Crémilleux, 2003a, Rioult et Crémilleux, 2004] concentrées sur les données du protocole $H7$, nous avons extrait des motifs δ-libres ($k = 1$). Nous avons constaté que des connaissances du domaine comme `médiastin envahi` $\Longleftrightarrow$ $MTR > 0$ n'étaient pas retrouvées lorsque les valeurs manquantes sont ignorées et remplacées par des valeurs absentes. Cela illustre les effets induits par ce type de stratégie, soulignés à la section 5.4 page 75.

Nous appliquons désormais la méthode décrite à la section 7.3 page 99 pour mettre en évidence des règles d'association généralisées correctes (définition 31 page 100), qui sont les règles informatives communes à la base et à son opposée. La profondeur des règles est limitée à 3 attributs, pour une fréquence minimale de 20 objets. La stratégie S_1 (une valeur manquante est considérée présente, cf. section 7.1) est utilisée pour le calcul des fermetures. Les règles relatives à la construction de la base de données sont retrouvées (voir table 12.2) : lorsque la dimension d'un ganglions est non nulle, il y a envahissement. Il est rassurant de constater que notre méthode détecte ces connaissances malgré les valeurs manquantes sur les dimensions de ganglions.

La table 12.3 rassemble les règles les plus simples qui ne décrivent pas un mode de construction des données, mais plutôt une connaissance anatomique. À part la première, ces règles concluent sur l'envahissement du médiastin lorsque les hiles gauche ou droit sont envahis. Cette connaissance est confirmée par l'expert. En effet, ces ganglions sont anatomiquement très voisins et on constate fréquemment un envahissement disséminé de cette zone.

Lors de nos expériences initiales, où un grand nombre de dimensions de ganglions étaient manquantes, ces règles ont également été retrouvées.

12.4.2 Classification non supervisée

La mise en évidence de groupes de patients homogènes intéresse fortement les médecins. Dans le cas de la maladie de Hodgkin, cette opération permet de discerner des comportements communs à certains patients et propose des stratégies pronostiques.

Plus précisément, nous utilisons la méthode `Ecclat` [Durand, 2004], développée par Nicolas Durand du GREYC. Les concepts extraits de la base sont mesurés suivant leur homogénéité (le rapport entre l'aire du concept et l'aire définie par tous les objets). Puis ils sont ordonnés et choisis selon cette mesure d'intérêt pour construire un clustering avec chevauchement. Ces concepts sont extraits en présence de valeurs manquantes. Pour cela, les motifs mv-1-libres sont calculés selon les définitions du chapitre 6 et leurs fermetures sont calculées suivant la stratégie S_1. La table 12.4 indique un exemple de répartition déterminée par trois concepts. Le recouvrement

prémisse	conclusion	fréquence
$chddim \in]0, 10]$	`chd` envahi	80
$chddim \in]10, 50]$	`chd` envahi	327
$chddim > 50$	`chd` envahi	62
$chgdim \in]0, 10]$	`chg` envahi	87
$chgdim \in]10, 50]$	`chg` envahi	351
$chgdim > 50$	`chg` envahi	68
$cbddim \in]0, 10]$	`cbd` envahi	232
$cbddim \in]10, 50]$	`cbd` envahi	1193
$cbddim > 50$	`cbd` envahi	205
$cbgdim \in]0, 10]$	`cbg` envahi	240
$cbgdim \in]10, 50]$	`cbg` envahi	1387
$cbgdim > 50$	`cbg` envahi	250
$axddim \in]0, 10]$	`axd` envahi	114
$axddim \in]10, 50]$	`axd` envahi	315
$axgdim \in]0, 10]$	`axg` envahi	131
$axgdim \in]10, 50]$	`axg` envahi	381
$mtr \in [30, 49]$	`med` envahi	574
$mtr \geq 50$	`med` envahi	321

TAB. 12.2 – Règles de construction des données découvertes.

moyen entre les groupes est de 221 patients, le groupe poubelle contient 1 107 patients.

Les patients sont *a priori* répartis parmi trois classes pronostiques par l'expert et la survenue d'événement au cours du traitement (rechute, décès) est connue. `Ecclat` calcule la répartition sans connaître ces groupes *a priori*, qui est évaluée et visualisée selon ces critères. Les groupes homogènes qui contiennent des patients de pronostic ou de réactions différents intéressent fortement l'expert, car ils permettent d'envisager des caractérisations nouvelles à des fins pronostiques.

prémisse	conclusion	fréquence
`homme` $\wedge$ $polyn2 <= 3000$	$chgdim \in]10, 50]$	23
`femme` $\wedge$ `cbg` non envahi $\wedge$ `hilg` envahi	`med` envahi	61
`femme` $\wedge$ `hild` envahi $\wedge$ $vs \in]30, 49]$	`med` envahi	45
`femme` $\wedge$ `hilg` envahi $\wedge$ $vs \in]30, 49]$	`med` envahi	41
`femme` $\wedge$ `hilg` envahi $\wedge$ $histo2 = 2$	`med` envahi	49
`hild` envahi $\wedge$ $hb \leq 105$	`med` envahi	50
`hild` envahi $\wedge$ $pa > 200$	`med` envahi	60
`hilg` envahi $\wedge$ $vs > 49$	`med` envahi	173
`hilg` envahi $\wedge$ $hb \leq 105$	`med` envahi	44
`hilg` envahi $\wedge$ $gb > 120$	`med` envahi	138
`hilg` envahi $\wedge$ $pa > 200$	`med` envahi	65
`hilg` envahi $\wedge$ $histo2 = 3$	`med` envahi	49
`hild` envahi $\wedge$ $sg = non \wedge gb > 120$	`med` envahi	53

TAB. 12.3 – Connaissances anatomiques découvertes.

Description	Nb de patients	Nb de patients classés
`med=0 hild=0 hilg=0 vs<=30 plaq<=600`	920	920
`med=1 sg=0 vs<=30 hb>105 plaq<=600`	919	919
`med=1 gb<=120 polyn2>3000`	1622	957

TAB. 12.4 – Exemple de classification supervisée avec `Ecclat`.

Chapitre 13

Extraction de motifs émergents forts dans des données génomiques

Ce chapitre relate une collaboration avec les biologistes Sylvain Blachon et Olivier Gandrillon du CGMC (Centre de Génétique Moléculaire et Cellulaire, UMR 5534 du CNRS et de l'Université Claude Bernard à Villeurbanne). Elle concerne des expériences menées sur des données d'expression de gènes et le séquençage du matériel génétique de cellules impliquées dans diverses situations biologiques : prélèvement d'un organe déterminé, ou culture cellulaire *in-vitro*. Les biologistes étant intéressés par l'identification de fonctions cancéreuses, nous utilisons des motifs émergents forts (SEP) (voir chapitre 10) pour caractériser la classe des situations correspondantes.

Ces données rassemblent peu d'objets mais de très nombreuses expressions de gène, et l'application directe des algorithmes de fouille de données n'est pas possible [Rioult *et al.*, 2003a]. Aussi nous utilisons la méthode que nous avons proposée à la section 10.2 pour faire face à ce type de données : le principe de la transposition des bases de données comportant un grand nombre d'attributs est appliqué sur des données d'expression de gène.

Après avoir détaillé les données considérées à la section 13.1, la section13.2 illustre l'intérêt de la transposition pour ce contexte. La section 13.3 relate l'analyse biologique conduite par les experts.

13.1 Données biologiques

Les expériences portent sur des données d'expression de gènes humains, obtenues par la méthode `SAGE` (Serial Analysis of Gene Expression) [Velculescu *et al.*, 1995]. Ces données sont publiques et disponibles sur le site `SAGE` Genie[11]. Elles ont été peu exploitées, essentiellement à

[11] `cgap.nci.nih.gov/SAGE`

cause des dimensions des données. La matrice que nous utilisons décrit 90 situations avec 27 679 gènes ; elle a été préparée par Sylvain Blachon et Olivier Gandrillon. Notons qu'elle est aussi utilisée pour les Discovery Challenge d'ECML/PKDD 2004 et 2005[12].

Le séquençage des situations biologiques (le terme `SAGE` est « librairie ») indique une liste des paires de bases d'acides aminés (A, C, T, ou G). L'identification `SAGE` est réalisée à l'aide d'*étiquettes* ou *tag* de dix paires de bases qui identifient les gènes connus. Il est donc plus précis d'indiquer que nous avons travaillé sur 90 objets représentant les situations, et 27 679 étiquettes `SAGE`.

La qualité des données du transcriptome suscite de nombreuses questions aux biologistes, que ces données soient issues de méthodes fondées sur l'hybridation moléculaire (puces à ADN) ou basées sur le séquençage (comme la technique `SAGE`). Des erreurs de séquençage peuvent intervenir dans la production des données SAGE et la correspondance entre les tags (paires de bases) et les gènes n'est pas toujours établie avec le même degré de confiance. Les interprétations biologiques doivent donc être menées prudemment. Pour les puces à ADN, de nombreux paramètres et facteurs de correction influencent le résultat et rendent ainsi plus délicates certaines interprétations. Cependant, même si la technique `SAGE` est économiquement plus coûteuse que les puces à ADN, elle permet d'obtenir des informations sans *a priori* sur les gènes étudiés, même s'ils sont peu exprimés. Ceci est un avantage par rapport à la technique des puces à ADN qui se limitent à l'expression des gènes impliqués dans l'expérience et ne peut découvrir de nouveaux gènes.

Les situations biologiques peuvent être classées de différentes manières : selon leur organe d'origine, leurs modes de développement (*in-vivo* ou et *in-vitro*) et leur affection par un cancer. Les modes de développement de ces cellules sont intéressants pour les biologistes car les gènes s'y expriment différemment. La mise en évidence des contrastes entre les cellules cancéreuses et les cellules saines a été privilégiée parce que c'est le problème favori des biologistes. Des détails sur les autres classes figurent dans [Rioult, 2004b].

Nous avons construit un contexte booléen en considérant qu'un gène est exprimé dans une situation si son taux d'expression dépasse les 75 % du maximum d'expression pour toutes les situations. Le lecteur intéressé trouvera dans [Becquet *et al.*, 2002] une discussion sur les procédures de codage de la sur-expression de ces données biologiques.

Dans cette présentation, un « petit » jeu de données (74×822) extrait de la matrice entière sera utilisé comme référence pour certains calculs pour permettre des comparaisons car ce contexte peut être exploré selon ses deux orientations. Sa portée biologique ne sera pas discutée.

[12] http://lisp.vse.cz/challenge/ecmlpkdd2004

13.2 Extraction des SEP avec la base transposée

Dans un premier temps, cette section montre que, grâce à notre approche, il est possible d'extraire les motifs émergents forts dans ce type de données. Puis nous donnons des résultats quantitatifs sur l'ensemble des motifs extraits.

13.2.1 Extraction des concepts

Rappelons que l'extraction des SEP dans les larges jeux de données requiert le calcul des motifs fermés (cf. section 10.2). La table 13.1 fournit des mesures sur les extractions de concepts réalisées dans les deux jeux de données, sans ou avec transposition. Elle donne le nombre de candidats pour le test de fréquence, et le nombre de motifs qui ont passé cette étape et sont donc 1-libres. Le nombre de motifs fermés - ou de concepts - et le temps d'éxécution terminent cette liste.

base r	mesures	extraction dans r	extraction dans ${}^t r$
74 × 822	candidats	474 244	20 246
	motifs 1-libres	38 349	8 290
	motifs fermés	1 617	1 617
	temps	12"	1"
90 × 27679	candidats	extraction	137 146
	motifs 1-libres	en échec	58 328
	motifs fermés	par manque	13 660
	temps	de mémoire	5'47"

TAB. 13.1 – Mesures de l'extraction dans r et sa transposée ${}^t r$.

Les expériences sont réalisées sur un Pentium 800MHz avec 256MB de mémoire sous Linux. On peut considérer que ces caractéristiques sont insuffisantes et limitent les capacités de calcul. La suite montrera que cette intuition est inexacte.

L'extraction dans le contexte transposé est exceptionnellement efficace. Dans la petite matrice 74×822, un candidat sur 12 est conservé lors de l'extraction dans r, 2,5 dans ${}^t r$. Concernant la proportion de motifs fermés devant le nombre de motifs 1-libres, elle est de un pour 23 dans r et seulement un pour 5 dans ${}^t r$.

Les résultats complets pour la grande matrice 90×27679 sans transposition ne peuvent être fournis car l'extraction a échoué faute de mémoire. Même s'il n'y a finalement que 12636 gènes présents dans au moins deux situations, il y a tout de même 79 millions de candidats de longueur 2. Après 93 minutes d'examen de la base, la génération des candidats de longueur 3 à l'aide des 200 000 motifs 1-libres de longueur 2 échoue.

Notons toutefois que la limitation de la mémoire de la machine de test a une incidence limitée. Sur une machine 32-bits Linux dotée de 4GB de mémoire, ce qui laisse 3GB pour l'utilisateur, l'extraction a également échoué [Rioult *et al.*, 2003b]. À ce stade, ce ne sont pas les quelques GB offerts par les machines récentes qui pourront améliorer cette performance. Dans la transposée, l'extraction de l'ensemble des concepts est réalisée en six minutes.

La figure 13.1 représente les temps comparés d'exécution de notre algorithme d'extraction des concepts selon la grande matrice 90×27679 ou sa version transposée. La fréquence minimum varie de 1 à 15. Le temps *transposé* est le temps nécessaire à l'extraction de l'intégralité des concepts et le filtrage des plus fréquents. Cette opération nécessite un temps quasi constant et montre clairement sa supériorité lorsque le seuil descend en dessous de 4 objets.

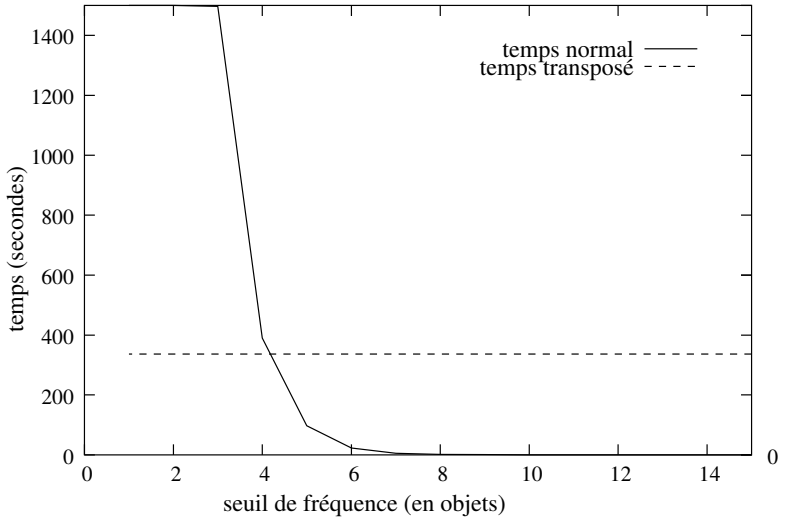

FIG. 13.1 – Extraction de concepts fréquents dans la grande matrice 90×27679.

13.2.2 Motifs émergents forts

Nous avons trouvé 5 277 SEPs relatifs à la classe cancer, 991 pour la classe non-cancer. Leur distribution selon la longueur est indiquée à la table 13.2. L'examen de ces SEPs un par un est fastidieux et des présélections peuvent être immédiatement effectuées suivant leur longueur, leur fréquence et leur taux de croissance. D'autres techniques de filtrage, du ressort des biologistes, peuvent être appliquées (voir section suivante).

13.3 Analyse biologique

Pour extraire des informations biologiques pertinentes des motifs disponibles, les experts ont effectué une sélection parmi les SEP fournis. *Le* motif finalement étudié a été choisi selon des

	Classe cancer		
fréquence	nombre de SEPs	longueur moyenne	dev.stand. longueur
1	59	548	209
2	1615	11	9,4
3	2112	2	1,2
4	716	1,2	0,4
5	306	1	0,1
6	185	1	0,1
7	53	1	0
8	27	1	0
9	12	1	0
10	7	1	0
11	2	1	0
12	2	1	0
13	1	1	0

TAB. 13.2 – Distribution des SEPs.

considérations « syntaxiques ». Nous rapportons ici l'analyse biologique effectuée par les experts sur ce concept qui associe un motif de gènes émergent fort pour la classe cancer et un motif de situations biologiques.

Une représentation graphique de propriétés de ce motif permet de mieux appréhender la nature des phénomènes biologiques capturés par un SEP. Finalement, la littérature disponible sur les situations biologiques et les gènes permet d'approfondir l'interprétation biologique.

13.3.1 Processus de sélection des SEPs

Les biologistes se sont concentrés sur les SEPs caractéristiques des situations cancéreuses, qui sont également des JEPs, c'est-à-dire qu'ils sont présents dans cette classe et absents du reste de la base. En ce sens, leur interprétation biologique est facilitée. Les SEPs contenant des gènes dont la fonction contient les termes « kinase[13], hormone, p53, EGF » ont été choisis pour leur implication connue dans les processus cancéreux. Cette sélection conserve 114 motifs.

Parmi eux, 43 concepts contiennent deux situations biologiques, 65 en contiennent trois. Les cinq derniers concernent quatre situations, et contiennent tous le mot clé « kinase » dans la descriptions d'un de leurs gènes. Parmi eux, le concept contenant le plus de gènes a été choisi.

[13]Enzyme assurant le transfert d'un phosphate provenant de l'adénosine triphosphate (ATP) sur un accepteur qui est ainsi activé. Ce mécanisme est fréquemment perturbé dans les cancers.

13.3.2 Présentation du concept sélectionné

Il s'agit du concept suivant (étiquette SAGE ou gène - situations) :

$$(\{864,\ 19258,\ 19378\},\ \{HCT116,\ ES2-1,\ OVT-8,\ HS766T\})$$

Il associe les situations biologiques suivantes (*cell line* indique une lignée cellulaire, *bulk* une culture) :

HCT116 : adenocarcinoma colon `SAGE` CGAP `SAGE` library method cell line [Zhang *et al.*, 1997] ;

ES2-1 : ovary clear cell poorly differentiated carcinoma CGAP `SAGE` library method cell line [Hough *et al.*, 2000] ;

OVT-8 : ovary serous adenocarcinoma bulk CGAP `SAGE` library method bulk [Hough *et al.*, 2000] ;

HS766T : pancreas adenocarcinoma cell line CGAP `SAGE` library method cell line.

Les gènes impliqués sont :

864 : (AACCTGGAGG) GCDH glutaryl Coenzyme A dehydrognase ;

19258 : (GTGGCCCGCA) AKAP8L A kinase (PRKA) anchor protein 8-like ;

19378 : (GTGGTAAGCA) Trans. seq. with weak similarity to protein ref :NP_060265.

La fonction biologique du deuxième gène est inconnue et fait l'objet des développements suivants.

Représentation graphique

Pour explorer le pouvoir d'expression des gènes capturé par le concept, le niveau d'expression de ses trois gènes a été représenté (figure 13.2). La partie A représente ces niveaux pour les 90 situations biologiques (les quatre situations pour le concept sont encadrées), et la partie B fournit un agrandissement pour trois situations. On y distingue le niveau d'expression des trois gènes.

À la lecture de ce graphique, il apparaît que le SEP sélectionné capture un phénomène très local. Chaque gène individuel montre des variations d'expression à la fois dans les cellules cancéreuses et non cancéreuses. Néanmoins, la sur-expression simultanée ne se produit que dans quatre cellules cancéreuses et aucune cellule non cancéreuse. Bien que cela puisse être interprété comme une pure coïncidence, une analyse plus précise révèle que les gènes 864 et 19378 sont en général fortement exprimés dans les cellules cancéreuses. Il est cependant très clair que ce motif de gènes ne caractérise pas toutes les cellules cancéreuses.

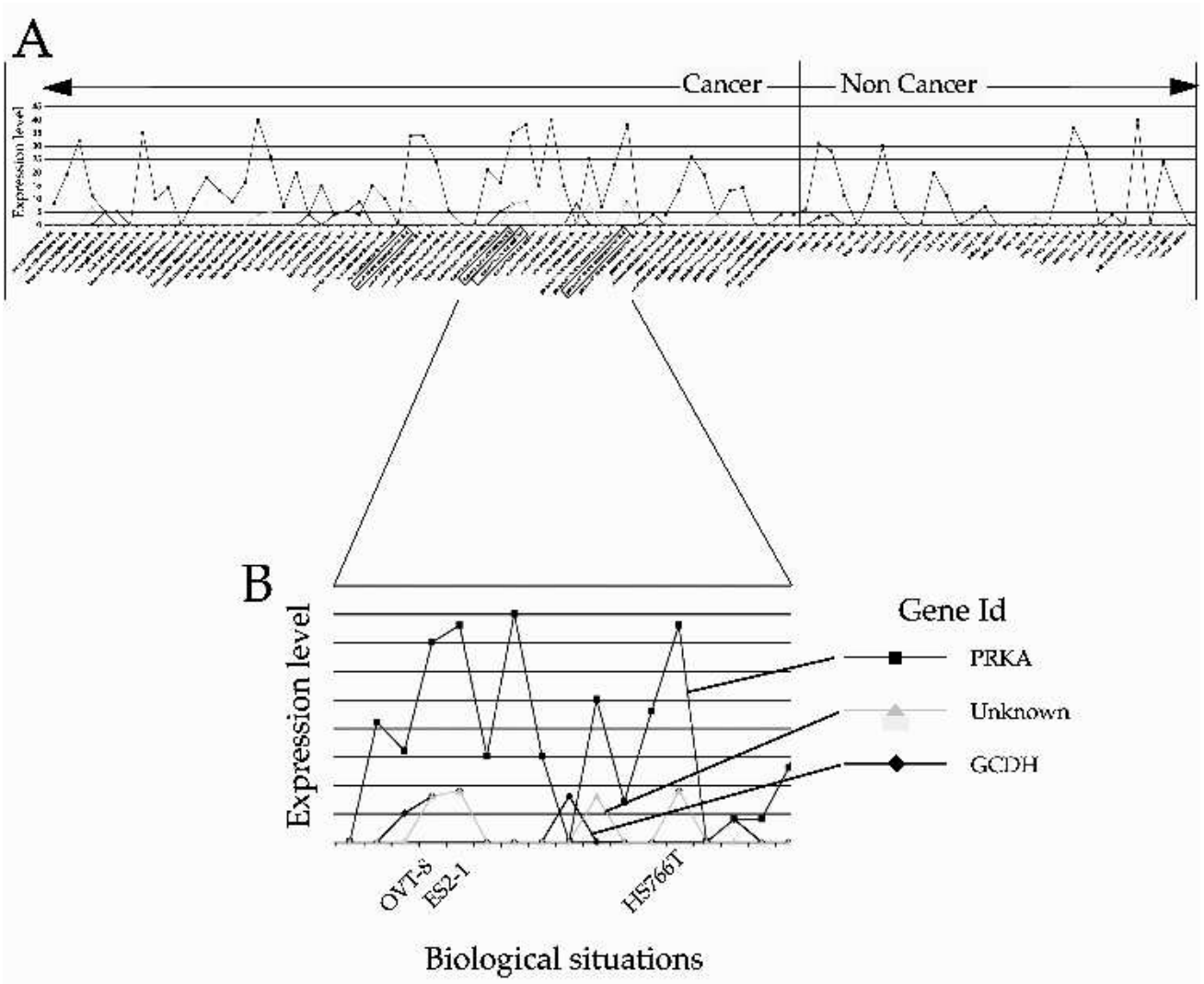

FIG. 13.2 – Représentation graphique du concept ({864, 19258, 19378}, {$HCT116$, $ES2-1$, $OVT-8$, $HS766T$})

13.3.3 Analyse du contenu biologique du SEP

Les experts ont tout d'abord examiné les quatre situations biologiques recensées par le concept. À part le caractère cancéreux lié aux conditions expérimentales, ils n'ont pas trouvé d'autre homogénéité flagrante. Trois d'entre elles concernent cependant des formes de cancer « adenocarcinoma[14] », alors que seulement 27 % de l'ensemble des situations sont de ce type. Deux situations biologiques du concept sont relatives à des cancers des ovaires (10 % pour toutes les situations de la base). La sur-expression simultanée des gènes 864, 19258 et 19378 semble se produire préférentiellement sur des adenocarcinomas ou des tumeurs d'origine ovarienne.

[14]Tumeur maligne qui se développe aux dépens des tissus glandulaires.

Identification des gènes

La caractérisation de la fonction des gènes est une tâche toujours en progrès. Ce processus consiste en l'assignation non ambiguë de chaque identifiant SAGE à la transcription des gènes dont ils dérivent. Ce travail n'est pas encore abouti pour les gènes humains et dépend des informations du site `SAGE`.

Nous discutons maintenant de l'identification des gènes impliqués dans le SEP, à partir des définitions `SAGE`. La qualité de ces identifications doit être appréhendée prudemment.

L'étiquette 864 possède une identification interne au gène CDH. Ces étiquettes internes proviennent soit de d'épissage alternatif, soit de digestion NlaIII insuffisante. Comme cette étiquette apparaît dans un assez grand nombre de librairies, la digestion NlaIII insuffisante est assez improbable. La possibilité existe donc que cette étiquette identifie une séquence mARN, et les interprétations biologiques doivent être menées avec prudence.

L'étiquette 19258 peut être identifiée en AKAP8L avec un plus fort niveau de confiance, par une séquence mARN située à l'extrême fin $3'$ du transcript (confirmé par la présence d'un signal et d'une queue PolyA). Les experts ont focalisé leur attention sur cette étiquette.

Enfin, l'étiquette 19378 provient de courtes séquences, obtenue auprès du programme ORESTES [Camargo et al., 2001]. Son identification ne peut donc être qualifiée de manière définitive.

Recherches bibliographiques

Une recherche dans la base `PubMed` (contenant plus de 14 millions de reférences dans le domaine biomédical depuis les années 1950) avec les mots clés « A kinase (PRKA) anchor protein 8-like » n'a retourné aucun résultat. La seule source d'information disponible pour l'étiquette 19 258 est disponible sur le site `http://www.ncbi.nlm.nih.gov/LocusLink/` qui répertorie les *locus*[15]. La protéine AKAP8L contient un domaine réservé similaire à celui de la protéine AKAP95 A-kinase ou PKA. AKAP95 est impliquée dans la condensation chromosomiale mitotique. Cette condensation est une étape critique de la mitose et ses dérégulations ont des conséquences certaines sur l'apparition de cellules cancéreuses.

Les experts ont alors lancé une recherche pour « A-kinase anchoring protein cancer » qui a retourné 11 résultats. Chaque article a été examiné. L'un d'entre eux était spécialement exaltant à la lumière des analyses précédentes. Il montrait que l'expression de AKAP3 mARN (très proche de notre protéine AKAP8L) était sur-exprimé dans des ovaires cancéreux et jamais dans des ovaires sains [Hasegawa *et al.*, 2004]. La conclusion de cette étude biologique permet de proposer de nouvelles expériences pour les biologistes impliqués dans les tumeurs des ovaires : vérifier l'expression d'un gène faiblement caractérisé, dénommé AKAP8L.

[15] Emplacement qu'occupe un gène ou un allèle sur un chromosome ou, par extension, sur la carte factorielle qui le représente.

13.4 Conclusion

La fouille de ces données génomiques et son interprétation biologique mettent en évidence le rôle d'un gène de la famille des A-kinases dans la génération ou la conservation de certains types de cancers, parmi lesquels les cancers des ovaires. On peut supposer que les deux autres gènes contenus dans notre SEP jouent un rôle dans le phénotype cancéreux final. Il serait très instructif d'aligner les régions fondatrices de ces gènes pour identifier les facteurs de transcription responsable de la sur-expression simultanée.

Rappelons que notre méthode fondée sur la transposition et les propriétés de la connexion de Galois permet de repousser les limites de la faisabilité des extractions, car la base est impossible à exploiter dans ses dimensions originales avec les techniques classiques de la fouille de données. Pour les biologistes, cet aspect est très important car ils ne sont plus obligés de limiter leurs données ou de sélectionner les gènes. Ils peuvent utiliser leur base sans restriction et choisir les motifs *après* la phase d'extraction. [Rioult *et al.*, 2003b, Besson *et al.*, 2004] montrent que la transposition permet d'enquêter sur différentes méthodes de discrétisation des signaux d'expression, qui fournissent des matrices praticables de densités variables. La transposition de bases de données a ouvert de nouveaux horizons pour les biologistes.

L'exemple d'analyse biologique d'un motif que nous avons relaté ici montre le caractère fastidieux du processus d'analyse, et suggère aux biologistes de nouveaux mode d'expérimentation impliquant l'informatique pour l'interprétation des résultats, en particulier lors des phases de recherche d'information[16].

[16] Ce travail se poursuit dans le cadre de l'ACI Bingo MD, qui regroupe quatre laboratoires : CGMC, Eurise, GREYC et LIRIS.

Bilan et perspectives

Bilan

Notre problématique de recherche porte sur l'extraction de connaissances dans des bases de données comportant des valeurs manquantes ou un grand nombre d'attributs. Au cours de ce travail, nous nous sommes attaché à proposer des adaptations formelles des techniques de fouille de données fondée sur les motifs ainsi que de nouvelles méthodes dans ces contextes difficiles, fréquemment rencontrés dans les processus réels d'extraction de connaissances.

Valeurs manquantes

Notre première contribution concerne l'impact des valeurs manquantes sur les représentations condensées de motifs et sur la production de règles d'association généralisées. Après avoir exposé les dégâts occasionnés par les données manquantes sur les motifs des représentations condensées, nous avons proposé un principe de désactivation partielle et temporaire qui quantifie l'écart entre la fréquence d'un motif généralisé dans la base incomplète et sa fréquence réelle dans la base complète. Cette technique permet de borner la fréquence d'un motif et de définir dans une base incomplète une propriété de mv-k-liberté qui est *compatible* et *consistante* avec la notion analogue dans toute base complète.

La consistance de la mv-k-liberté est un point central de notre travail. Elle assure que les motifs mv-k-libres calculés dans une base *avec valeurs manquantes* sont k-libres dans toute base complète dont elle est issue. La définition de la mv-k-liberté est compatible avec la k-liberté. Cela signifie que l'algorithme `MV-k-miner`, appliqué dans une base *sans valeurs manquantes*, calcule de façon équivalente les motifs k-libres.

Nous avons défini les règles d'association généralisées informatives comme étant des règles non redondantes, dont la conclusion est la plus générale possible. Nous avons montré que ces règles peuvent être obtenues de façon fiable dans les bases de données incomplètes. La précision du calcul est mesurée à l'aide de bases de référence.

L'avantage de notre approche réside dans l'absence de traitement d'imputation appliqué aux valeurs manquantes. Grâce à la consistance des motifs k-libres, les règles obtenues indiquent des

connaissances valides dans toute base complète dont la base incomplète est issue.

Grand nombre d'attributs

Notre deuxième contribution est relative aux bases de données comportant un grand nombre d'attributs. Pour pouvoir extraire les motifs fréquents, nous avons combiné les propriétés de la connexion de Galois et les algorithmes d'extraction de concepts pour calculer les motifs fermés. Entre une base et sa version transposée, les concepts coïncident, moyennant inversion. Cette correspondance permet de choisir l'orientation de la matrice et d'effectuer les calculs dans le contexte dont l'orientation est la plus favorable pour finalement réussir des extractions, infaisables sinon. Cette perspective ouvre de nouveaux horizons pour l'analyse des données biologiques.

Notre approche a été étendue à d'autres contraintes que la fréquence et nous avons proposé un cadre formel pour la transposition des contraintes. La correspondance de Galois est utilisée avec la contrainte transposée, qui est équivalente sur les objets à la contrainte initiale sur les attributs. Relaxée, cette contrainte identifie les motifs fermés pour les classes d'équivalence contenant des motifs contraints. Une phase de regénération est nécessaire pour obtenir l'ensemble des motifs contraints.

L'utilisation des techniques de transposition permet d'appliquer les algorithmes classiques de la fouille de données dans des contextes jusqu'ici inabordables. Elle repousse les limites de la faisabilité et dispense les biologistes d'une quelconque préselection des gènes étudiés.

Usage des motifs k-libres et domaine applicatif

Le dernier axe a trait à la conception et au développement de méthodes d'exploration des données ainsi que leur mise en œuvre dans des contextes applicatifs réels. Ces travaux s'appuient sur les résultats obtenus aux axes précédents. Si la production de règles généralisées informatives dans les bases incomplètes illustre un usage des motifs k-libres, ils peuvent aussi être exploités et développés pour effectuer des tâches de classification supervisée. Ils fournissent une connaissance profonde des données, sans nécessiter l'emploi des motifs fréquents.

Dans le domaine médical, nous avons étudié des données incomplètes relatives à la maladie de Hodgkin. Les règles généralisées correctes obtenues confirment des connaissances spécifiques du domaine. Malgré de nombreuses valeurs manquantes, l'information obtenue valide les contraintes présentes par construction dans les données ainsi que des connaissances anatomiques.

Enfin, les motifs émergents forts ont été extraits d'une matrice très large de données d'expression de gène SAGE. Ces motifs caractérisent les états cancéreux constatés dans ces situations biologiques. Un motif émergent fort a été plus particulièrement examiné par les experts, qui ont ainsi pu préciser l'appartenance d'un gène particulier à une famille impliquée dans le cancer des ovaires.

Perspectives

Nos perspectives de recherche découlent d'abord des prolongements de nos travaux sur les contextes difficiles et de façon plus générale sur les usages des motifs k-libres et les processus d'extraction de connaissances.

Données difficiles

Les résultats théoriques obtenus sur l'extraction de motifs dans les données incomplètes doivent être mieux quantifiés. Il serait en particulier utile de mesurer leur adéquation lors de la mise en œuvre de méthodes d'exploration. En revanche, comme nous l'avons souligné, il semble vain de vouloir améliorer d'un point de vue théorique le compromis offert par la consistance de nos définitions de la k-liberté.

La transposition de contraintes est une approche prometteuse pour l'extraction de motifs sous contraintes dans les larges jeux de données. Néanmoins, l'efficacité de l'étape de regénération des motifs doit être améliorée.

La transposition de base de données laisse les valeurs manquantes invariantes. Ce constat peut être mis à profit pour extraire des motifs fiables dans les données incomplètes, en comparant les résultats d'extractions effectuées dans une base et sa base transposée. À la différence de l'opposition de base pour les motifs k-libres, il n'y a cependant pas de relation immédiatement profitable concernant les concepts en présence de valeurs manquantes.

Usages des motifs k-libres

Les motifs k-libres méritent une large promotion, d'autant que nous avons des techniques pour les extraire dans des bases de données incomplètes. Bien sûr, ils sont réputés pour fournir très efficacement une représentation condensée des motifs fréquents, mais leur intérêt se situe aussi au niveau de leurs usages. Nous pensons que leur apport en classification supervisée sera particulièrement fécond. Une voie peut consister à tirer profit du pouvoir d'expression des règles d'association généralisées pour adapter les méthodes connues en classification à base d'associations. En effet, une règle généralisée autorise la production de règles positives et négatives, qui indiquent alors une classe ou l'interdisent.

Une autre voie concerne la classification non supervisée. Les méthodes actuelles utilisent les concepts comme base pour les groupes. Ces méthodes peuvent être adaptées pour démarrer la répartition à partir des motifs k-libres. Un motif k-libre est porteur d'une absence de corrélation entre les attributs qu'il rassemble. Il existe de plus une dualité avec la base opposée, où ces motifs deviennent des éléments de fermeture, ce qui induit la présence d'une corrélation. Pour la constitution de groupes homogènes, cette notion peut être exploitée avec bénéfice.

Dans ce mémoire, nous avons beaucoup parlé des motifs k-libres et cité les motifs δ-libres. Intuitivement, il semble séduisant de combiner ces deux notions pour définir les motifs δ-k-libres, conjuguant tolérance d'exceptions et efficacité d'extraction. Malheureusement, ces deux notions coexistent difficilement avec les fermetures généralisées. Considérons la règle généralisée $a_1 \rightarrow a_2 \vee a_3$. Elle signifie que soit a_2, soit a_3 est toujours présent avec a_1. Si a_2 est peu présent, moins de δ fois, doit-on le traiter comme une exception ?

Perspectives générales

Il est bien connu que la binarisation d'un attribut continu est une tâche difficile. Lors d'une collaboration avec la société PHILIPS, nous avons eu à étudier des plaques de silicium défectueuses. Cette caractéristique était fournie par une mesure continue de qualité entre 0 et 100. Comment effectuer convenablement la séparation entre les plaques défectueuses et les plaques correctes ? De façon un peu paradoxale, les valeurs manquantes pourraient s'avérer intéressantes. Pourquoi ne pas séparer l'intervalle de variation en trois intervalles consécutifs ? Les valeurs basses et hautes sont codées, les valeurs intermédiaires sont laissées manquantes. L'utilisation des valeurs manquantes introduit une notion de flou dans l'interprétation des données. Sur une telle base incomplète, l'extraction des motifs selon notre méthode fournit des connaissances cohérentes avec toute base complète correspondante. Ainsi, les choix effectués lors des phases de prétraitement influent modérément sur la qualité de l'information obtenue. Cette technique de discrétisation partielle n'empêche pas de découvrir des connaissances correctes.

Enfin, nous pensons que les propriétés utilisées dans ce travail sur la fouille de données pourraient profitablement être revisitées dans le cadre de théories plus générales. Par exemple, l'inversion des règles dans la base opposée n'est pas une surprise pour un logicien. Les notions topologiques sur les structures discrètes qualifient profondément les rapports entre les motifs. La logique, les hypergraphes et les structures algébriques permettraient assurément de formuler les notions manipulées au cours de ce travail avec plus de profondeur et de recul.

Annexes

Chapitre 1

Calcul de traverses minimales

Les algorithmes classiques de calcul des traverses minimales sont expliqués dans cette annexe. Ce problème est bien identifié pour sa difficulté algorithmique et a donné lieu à la présentation de différentes solutions. Ces contributions sont *a priori* difficilement comparables entre elles, car la complexité exacte des algorithmes n'est pas disponible. Lors de nos recherches, nous avons donc implémenté plusieurs méthodes, dans le but de les comparer et de retenir la mieux adaptée à notre problématique.

Rappelons que les traverses minimales sont impliquées dans les procédures de génération des candidats pour les algorithmes d'extraction de motifs. La fermeture généralisée d'un motif k-libre est obtenue à l'aide des traverses minimales de longueur bornée par k (cf. section 3.4). Nous étudions donc dans ce chapitre les adaptations nécessaires aux algorithmes de calcul des traverses minimales pour résoudre ce problème.

La section 1.1 détaille le premier algorithme connu, défini par Claude Berge. Ensuite, la solution de Fredman et Kachiyan est examinée à la section 1.2. Cette présentation se termine à la section 1.3 par l'exposition de l'algorithme de Stavropoulos et Kavvadias.

1.1 L'algorithme de Berge

L'algorithme le plus classique est celui de Berge [Berge, 1989]. Il calcule une solution temporaire en ajoutant chaque hyper-arête de façon incrémentale. Si $H_l = \{\varepsilon_1, \ldots, \varepsilon_l\}$, alors $Tr(H_l \cup \{\varepsilon_{l+1}\}) = \{min\{t \cup \{v\}\} \mid t \in Tr(H_l)\ et\ v \in \varepsilon_{l+1}\}$: lorsqu'on ajoute une hyper-arête, les nouvelles traverses minimales sont obtenues en ne conservant que les minimums des anciennes traverses, auxquelles on a ajouté un item de la nouvelle arête. Outre sa complexité rédhibitoire (exponentielle en la taille de l'entrée et de la sortie), cet algorithme nécessite de stocker toutes les traverses intermédiaires et le résultat final n'est fourni qu'à l'issue de l'insertion de la dernière arête. Pendant de nos expériences, nous avons pu constater que cette solution était impraticable dès que les hypergraphes ont quelques dizaines de sommets.

1.2 L'algorithme de FREDMAN et KACHIYAN

La deuxième solution pour le calcul des traverses minimales est fournie par l'algorithme de FREDMAN et KACHIYAN [Fredman et Kachiyan, 1996], qui possède la meilleure complexité théorique connue (pour n sommets et b le nombre total d'arêtes en entrée *et* sortie, le coût est en $nb^{o(log^2 b)}$). Les auteurs motivent le calcul de traverses minimales comme la solution au problème de la dualisation des formules booléennes monotones et c'est cette présentation intuitive que nous reprenons.

Ces formules sont construites à l'aide de *littéraux* qui sont des variables booléennes $x_1, \ldots, x_n$, ou leurs négations. Une *clause* est une disjonction de littéraux, tandis qu'un *terme* est une conjonction. Une formule est sous *forme normale conjonctive* (resp. *disjonctive*) si c'est une conjonction de clauses (resp. une disjonction de termes). Une formule est monotone si elle ne contient que des littéraux positifs.

Étant donnée une formule $f(x) = f(x_1, \ldots, x_n)$ sous forme normale conjonctive, il s'agit de calculer la formule duale correspondante $f^d(x) = \overline{f}(\overline{x}) = \overline{f}(\overline{x_1}, \ldots, \overline{x_n})$ sous forme normale conjonctive également. Pour cela, on obtient aisément f^d sous forme normale disjonctive en remplaçant chaque conjonction de f par une disjonction et vice-versa. Pour calculer la forme normale conjonctive de la formule duale, il s'agit finalement de développer la forme disjonctive pour constituer les clauses de f^d. Pour cela, on prendra un littéral dans chaque terme de $\overline{f}$ pour constituer une clause. Des simplifications apparaissent si l'on prend plusieurs fois le même littéral. Pour calculer les traverses minimales d'un hypergraphe, chaque traverse est construite en prenant un item dans chaque terme de $\overline{f}$: le résultat obtenu est identique à celui fourni par la dualisation des formules booléennes monotones : ce problème et celui du calcul des traverses minimales d'un hypergraphe sont parfaitement équivalents.

Par exemple, soit $f(x) = (x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3 \vee x_4)$. La formule duale correspondante, obtenue en échangeant chaque conjonction par une disjonction et vice-versa, est $f^d(x) = (x_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee (x_2 \wedge x_3 \wedge x_4) \vee (x_1 \wedge x_2 \wedge x_3 \wedge x_4)$. Si l'on développe scrupuleusement cette dernière expression pour la transformer en forme normale conjonctive, on obtient la série de clauses suivantes : $f^d(x) = (x_1 \vee x_1 \vee x_1 \vee x_2 \vee x_1) \wedge (x_1 \vee x_1 \vee x_1 \vee x_2 \vee x_2) \wedge \cdots \wedge (x_2 \vee x_3 \vee x_4 \vee x_4 \vee x_4)$ (il y a en tout $2 \times 3 \times 3 \times 3 \times 4 = 216$ clauses). Après les simplifications qui ne gardent que les clauses minimales, il ne reste que trois clauses : $f^d(x) = x_2 \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4)$!

La solution à ce problème, fournie par FREDMAN et KACHIYAN, consiste à déterminer de façon incrémentale si deux formules f et g sont mutuellement duales, c'est-à-dire si $f(x) = \overline{g}(\overline{x})$. La formule g est vide au début de l'algorithme. Cette méthode permet, si f et g ne sont pas duales, d'exhiber une clause disqualifiante qui sera ajoutée à g et le processus est itéré. Lorsqu'aucun disqualifieur n'existe, la solution finale pour la dualisation de f est trouvée et il s'agit de g.

La vérification de la dualité et la mise en évidence d'un disqualifieur sont effectuées grâce à la propriété suivante : en factorisant f et g selon une variable x_i, on fait apparaître des formules plus courtes f_0, f_1, g_0 et g_1 qui ne contiennent pas x_i. On obtient $f(x) = (x_i \wedge f_0(y)) \vee f_1(y)$ puis $g(x) = (x_i \wedge g_0(y)) \vee g_1(y)$ (y ne contient pas le littéral x_i). f et g sont duales si et seulement si f_1 et $g_0 \vee g_1$ le sont, ainsi que $f_0 \vee f_1$ et g_1. La taille du problème est ainsi réduite et permet d'appliquer récursivement ce procédé.

Nous avons implémenté cette solution et plus précisément une généralisation de ce principe dédiée à la dualisation des boîtes entières [Boros *et al.*, 2003], où les formules ne traitent plus seulement des variables booléennes, mais des variables entières bornées. Malheureusement, cette méthode est peu adaptée à notre cadre théorique qui nécessite le calcul des traverses minimales *de longueur bornée* pour exhiber les fermetures généralisées des motifs k-libres.

1.3 L'algorithme de Stavropoulos et Kavvadias

Le dernier algorithme étudié est fourni par Stavropoulos et Kavvadias, qui ont beaucoup écrit sur ce domaine, tant du point de vue de sa résolution [Kavvadias et Stavropoulos, 1999a, Kavvadias et Stavropoulos, 1999b] que de sa complexité [Kavvadias et Stavropoulos, 2001, Kavvadias et Stavropoulos, 2003b, Kavvadias et Stavropoulos, 2003a]. Leur solution consiste à calculer les traverses selon un principe incrémental. Chaque traverse est construite item par item, pour chaque nouvelle hyperarête de l'hypergraphe de départ. Ces traverses sont en fait constituées d'items généralisés, une liste des items présents dans toutes les arêtes insérées précédemment. Les traverses finales sont calculées par un produit cartésien des items généralisés successifs.

Considérons par exemple, sur un ensemble de sommets $V = \{x_1, \ldots x_6\}$, l'hypergraphe $H = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}\}$. Les traverses minimales sont exprimées à l'aide des items généralisés $\{x_1, x_2\}$, $\{x_3, x_4\}$ et $\{x_5, x_6\}$, ce qui signifie qu'une traverse particulière sera obtenue en prenant un item dans chaque item généralisé : c'est un produit cartésien.

La deuxième notion utilisée par les auteurs consiste à morceler les items généralisés lorsque la nouvelle arête présentée n'intersecte que partiellement. Par exemple, si $H = \{\{x_1, x_2, x_3\}, \{x_3, x_4, x_5\}, \ldots\}$ (x_3 est commun aux deux premières hyperarêtes), nous aurons deux calculs à mener : le premier contient les items généralisés $\{x_1, x_2\}$ et $\{x_4, x_5\}$, le second ne contient que leur intersection $\{x_3\}$. Suivant ce principe, le calcul des traverses minimales consiste à parcourir en profondeur un arbre dont chaque niveau symbolise l'ajout d'une nouvelle arête. Lorsque cette arête intersecte partiellement, deux branches sont créées.

Enfin, lorsque la nouvelle arête n'a aucun item commun avec les items généralisés temporaires, les items de cette arête viennent constituer un nouvel item généralisé. Pour cela, ils doivent être *appropriés*, c'est-à-dire que les traverses ainsi constituées doivent être minimales. Cette condition

permet de garantir la contrainte de minimalité des traverses. Pour illustrer ce principe, prenons l'hypergraphe $H = \{\{x_1, x_3\}, \{x_2, x_3\}, \{x_3\}\}$. La combinaison des deux premières arêtes qui intersectent partiellement, fournit deux branches à explorer avec les items généralisés $\{x_1\}$ et $\{x_2\}$ d'une part et $\{x_3\}$ d'autre part. Lorsque l'on veut ajouter les items de la dernière arête $\{x_3\}$, nous obtenons les items généralisés $\{x_1\}$, $\{x_2\}$ et $\{x_3\}$, qui définissent la traverse $\{x_1, x_2, x_3\}$. Mais en ôtant un item de cette traverse, par exemple x_2, l'ensemble restant $\{x_1, x_3\}$ constitue malgré tout une traverse de H : x_3 n'est pas approprié pour cette branche et elle est élaguée.

Nous donnons ci-dessous (algorithme 6) le synopsis complet de la proposition de Stavropoulos et Kavvadias. L'appel initial est effectué avec une liste d'items généralisés vides et avec l'hypergraphe dont on cherche les traverses minimales. L'adaptation au cas du calcul des traverses minimales de longueur bornée est triviale, puisqu'il suffit d'interrompre le parcours en profondeur de l'arbre dès que le nombre d'items généralisés dépasse la longueur maximale désirée.

Données : un ensemble d'items généralisés $\{X_1, \ldots, X_k\}$ et un ensemble d'arêtes $H = \{\varepsilon_1, \ldots, \varepsilon_m\}$
Résultat : les traverses de H
si $m = 0$ **alors**
  retourner le produit cartésien des X_i
fin
si $\exists i \mid X_i \subseteq \varepsilon_1$ **alors**
  appeler récursivement l'algorithme en ôtant ε_1
fin
si $\exists i \mid \varepsilon_1 \subseteq X_i$ **alors**
  appeler récursivement l'algorithme en ôtant ε_1 et en remplaçant X_i par ε_1
fin
pour $i \mid X_i \cap \varepsilon_1 = Y \neq \emptyset$ **faire**
  appeler récursivement l'algorithme en retirant ε_1 et en remplaçant X_i par Y;
  appeler récursivement l'algorithme en remplaçant ε_1 par $\varepsilon_1 \backslash Y$ et X_i par $X_i \backslash Y$;
fin
si *aucun i ne fournit les propriétés ci-dessus* **alors**
  appeler récursivement l'algorithme en ôtant ε_1 et en ajoutant aux X_i les items appropriés de ε_1
fin

Algorithme 6 – Algorithme de calcul des traverses minimales par Stavropoulos et Kavvadias.

1.4 Discussion

Des conclusions pertinentes sur l'efficacité comparée des différentes solutions sont difficiles à prononcer. Nos investigations ne sont pas exhaustives, car l'algorithme récent d'Eiter et Gottlob [Eiter *et al.*, 2002], qui ont beaucoup étudié les traverses minimales [Eiter et Gottlob, 1995], n'a pas été testé. Cette dernière contribution calcule les traverses en respectant l'ordre lexicographique selon un procédé incrémental polynomial. Pour cela, les arêtes de l'hypergraphe sont projetées selon un espace qui comprend de plus en plus de variables ; une relation de récurrence lie les calculs de traverses correspondants.

Lors de nos tests, nous avons pu constater que la solution de Berge est impraticable dans les cas réels et que l'implémentation de l'algorithme de Fredman et Kachiyan est plus performante que celle de Stavropoulos et Kavvadias. Néanmoins, l'adaptation de cette dernière au cadre des fermetures généralisées, nécessitant le calcul des traverses de longueur bornée, est triviale. Le prototype `MV-k-miner` utilise donc l'implémentation de Stavropoulos et Kavvadias.

En ce qui concerne le calcul des traverses minimales de longueur quelconques, Céline Hébert (GREYC) a récemment proposé dans [Hébert et Bretto, 2005] une méthode qui utilise la relation entre la bordure négative et les traverses minimales des complémentaires. Elle utilise pour cela une inversion de cette relation : les traverses minimales sont la bordure négative pour les motifs complémentaires. L'extraction des motifs présents dans la base constituée des hyper-arêtes complémentaires de l'hypergraphe de départ calcule aisément cette bordure négative : ce sont les motifs qui ne vérifient pas les critères d'élagages et ne sont pas de bons candidats. Cette approche montre une efficacité nettement supérieure à celle de Fredman et Kachiyan dans les bases très denses.

Chapitre 2

Preuves pour le chapitre 4

Cette annexe contient les preuves des théorèmes 1 page 57 à 8 page 60 du chapitre 4.

2.1 Résultats communs

Les preuves pour les modèles de Bernoulli et de Markov sont fondées sur la formule suivante, qui exprime exactement le nombre moyen de motifs fréquents :

Lemme 1 *Soit p_A la probabilité qu'un objet contienne un motif A ($A \subseteq \mathcal{A}$). Le nombre moyen de motifs γ-fréquents dans une base aléatoire avec n objets indépendants satisfait*

$$\sum_{A \subseteq \mathcal{A}} \sum_{i=\gamma}^{n} \binom{n}{i} {p_A}^i (1-p_A)^{n-i} \tag{2.1}$$

Preuve : Soit A un motif et i sa fréquence. La probabilité que A appartienne à i objets vaut ${p_A}^i$. Chacun des $(n-i)$ objets restant ne contient pas A, et cette probabilité vaux $(1-p_A)$. De plus, il y a $\binom{n}{i}$ fréquences possibles de taille i. La somme correspondante prouve la formule 2.1. □

L'étape principale consiste maintenant à transformer la somme interne de la formule 2.1 en une intégrale. Différentes approximations de cette intégrale sont réalisées selon les hypothèses d'un seuil de fréquence fixe ou proportionnel.

Lemme 2 *La somme interne à la formule 2.1 se simplifie en l'intégrale*

$$\sum_{i=\gamma}^{n} \binom{n}{i} {p_A}^i (1-p_A)^{n-i} = \gamma \binom{n}{\gamma} \int_0^{p_A} x^{\gamma-1} (1-x)^{n-\gamma} dx$$

Preuve : Le développement de $(1-x)^{n-i}$ donne

$$\sum_{i=\gamma}^{n} \binom{n}{i} x^i (1-x)^{n-i} = \sum_{i=\gamma}^{n} \binom{n}{i} \sum_{u=0}^{n-i} \binom{n-i}{u} (-1)^u x^{i+u}.$$

Le changement de variable $v = u + i$ et l'inversion des deux sommes donne l'égalité

$$\sum_{i=\gamma}^{n} \binom{n}{i} x^i (1-x)^{n-i} = \sum_{v=\gamma}^{n} \binom{n}{v} x^v \sum_{i=\gamma}^{v} \binom{v}{i} (-1)^{v-i}.$$

Une récurrence simple montre que la seconde somme se simplifie en

$$(-1)^{v-\gamma} \binom{v-1}{\gamma-1}.$$

L'inégalité précédente devient

$$\sum_{i=\gamma}^{n} \binom{n}{i} x^i (1-x)^{n-i} = \sum_{v=\gamma}^{n} \binom{n}{v} x^v (-1)^{v-\gamma} \binom{v-1}{\gamma-1}.$$

Les coefficient binomiaux se simplifient $\binom{n}{v}\binom{v-1}{\gamma-1} = \frac{\gamma}{v}\binom{n}{\gamma}\binom{n-\gamma}{v-\gamma}$ et le changement de variable $w = v-\gamma$ donne la nouvelle expression

$$\sum_{i=\gamma}^{n} \binom{n}{i} x^i (1-x)^{n-i} = \gamma \binom{n}{\gamma} \sum_{w=0}^{n-\gamma} \binom{n-\gamma}{w} (-1)^w \frac{x^{w+\gamma}}{w+\gamma}.$$

Pour conclure, remarquons que la seconde somme est nulle si $x = 0$ et que sa dérivée selon x est exactement $x^{\gamma-1}(1-x)^{n-\gamma}$. □

Le prochain lemme résume les différentes approximations utilisées dans l'intégrale.

Lemme 3 *Si γ est fixe, l'intégrale est approximée par*

$$\int_0^{p_A} x^{\gamma-1}(1-x)^{n-\gamma} dx \approx \frac{{p_A}^{\gamma}}{\gamma} - (n-\gamma)\frac{{p_A}^{\gamma+1}}{\gamma+1} \tag{2.2}$$

où $a \approx b - c$ signifie $b - c \leq a \leq b$.

Si $\gamma = \alpha \cdot n$ est proportionnel avec n, alors pour $p_A < \alpha$ et un grand nombre n d'objets,

$$\int_0^{p_A} x^{\gamma-1}(1-x)^{n-\gamma} dx \leq {p_A}^{\gamma-1}(1-p_A)^{n-\gamma}. \tag{2.3}$$

Si $\gamma = \alpha \cdot n$ est proportionnel avec n, alors pour $p_A > \alpha$ et un grand nombre n d'objets,

$$\int_0^{p_A} x^{\gamma-1}(1-x)^{n-\gamma} dx \approx \frac{1}{\gamma\binom{n}{\gamma}} - {p_A}^{\gamma-1}(1-p_A)^{n-\gamma}. \tag{2.4}$$

Preuve : Comme $(1-x)^{n-\gamma} \approx 1-(n-\gamma)x$, le remplacement de $(1-x)^{n-\gamma}$ dans l'intégrale par son approximation donne directement la formule 2.2.
La fonction $x^{\gamma-1}(1-x)^{n-\gamma}$ est croissante sur $[0, \frac{\gamma-1}{n-1}]$ et décroissante sur $[\frac{\gamma-1}{n-1}, 1]$. Ces variations, combinées avec l'égalité (prouvée par récurrence)

$$\int_0^1 x^{\gamma-1}(1-x)^{n-\gamma} dx = \frac{1}{\gamma\binom{n}{\gamma}}$$

suffisent pour prouver les formules 2.3 et 2.4. □

2.2 Preuve du théorème 3 : le cas proportionnel

Nous pouvons maintenant prouver que dans les deux modèles, un objet contient un motif A de longueur j avec une probabilité exponentiellement décroissante avec j.

Lemme 4 *Dans le modèle de* Bernoulli *de paramètre p, un objet contient un motif A de longueur j avec la probabilité p^j.*
Dans le modèle de Markovian, *il existe deux constantes $\kappa > 0$ et $\theta \in]0,1[$ telles qu'un objet contient un motif A de longueur j avec une probabilité inférieure à $\kappa\theta^j$.*

Preuve : La première assertion est évidente. Rappelons que le modèle de Markov est défini par une matrice de transition P à coefficients strictement positifs. Chaque objet est découpé en paquets de K attributs (K est l'ordre du processus). Le premier paquet est généré par la distribution initiale f_{init} et les suivants sont générés selon les transitions de probabilité. Il y a au moins j/K paquets contraints, *i.e.* j/K possèdent au moins un attribut. Si p_1 est la probabilité minimale parmi les transitions $p_{w\to(0,\ldots,0)}$, alors la probabilité que A soit contenu par un objet est inférieure à $p_A \leq (1-p_1)^{\lfloor j/K \rfloor - 1}$. □

La décroissance exponentielle des probabilités est le point clé pour prouver le théorème 3.

Preuve : [Théorème 3] Dans les deux modèles, il existe une constante j_0 telle que tout motif A d'une longueur strictement inférieure à j_0 a une probabilité p_A de présence dans l'objet strictement inférieure à r. La formule 2.3 entraîne donc l'inégalité

$$\sum_{|A|>j_0} \gamma\binom{n}{\gamma} \int_0^{p_A} x^{\gamma-1}(1-x)^{n-\gamma}dx$$

$$\leq \sum_{j=j_0+1}^{m} \binom{m}{j}\gamma\binom{n}{\gamma}(\kappa\theta^j)^{\gamma-1}(1-\kappa\theta^j)^{n-\gamma}$$

Soit w_j les coefficients de la somme. w_{j+1}/w_j est inférieur à w_{j_0+2}/w_{j_0+1}, et l'utilisation des équivalences et de l'hypothèse $\mathcal{H}_1$ prouve que w_{j_0+2}/w_{j_0+1} tend vers zéro. La somme est donc équivalente à w_{j_0+1} et une fois encore, à l'aide d'équivalences, on prouve que w_{j_0+1} converge vers zéro.
La seconde partie de la somme est bornée par $\sum_{j=1}^{j_0}\binom{m}{j}$, équivalent à $\frac{m^{j_0}}{j_0!}$. Ceci prouve le comportement polynomial du théorème 3. □

Dans le modèle de Bernoulli, des calculs similaires avec la probabilité exacte p^j donnent l'équivalence $F_{m,n,\gamma} \sim \binom{m}{j_0}$ avec $j_0 = \lfloor \frac{\log\alpha}{\log p} \rfloor$, pour α n'étant pas une puissance de p [Lhote *et al.*, 2005a].

2.3 Preuve du théorème 2 : le cas du seuil fixe

L'inégalité 2.2 entraîne que le nombre moyen de motifs fréquents est approximé par

$$F_{m,n,\gamma} \approx \binom{n}{\gamma}\sum_{A\subseteq\mathcal{A}} {p_A}^{\gamma} - \frac{\gamma(n-\gamma)}{\gamma+1}\binom{n}{\gamma}\sum_{A\subseteq\mathcal{A}} {p_A}^{\gamma+1}$$

L'estimation de la somme $S_\gamma = \sum_{A\subseteq\mathcal{A}} {p_A}^\gamma$ est nécessaire.

Modèle de Bernoulli **:** la probabilité p_A satisfait $p_A = p^j$ avec $j = |A|$ la longueur de A. En introduisant cette relation dans la somme, nous obtenons $\sum_{A\subseteq\mathcal{A}} {p_A}^\gamma = (1+p^\gamma)^m - 1$. En posant $\eta = (1+p^{\gamma+1})/(1+p^\gamma)$, le nombre moyen de motifs fréquents satisfait $F_{m,n,\gamma} = \binom{n}{\gamma}(1+p^\gamma)^m\,[1+O(n\eta^m)]$.

Modèle de Markov **(idée de la preuve) :** il est beaucoup plus compliqué d'estimer la somme. Soit p_t la probabilité de générer l'objet o dans le modèle de Markov (P, f_{init}). La somme S_γ satisfait les égalités

$$S_\gamma = \sum_{A\subseteq I}\left(\sum_{t:A\subseteq t} p_t\right)^\gamma = \sum_{A\subseteq I}\sum_{t_1,\ldots,t_\gamma:A\subseteq t_j} p_{t_1}\ldots p_{t_\gamma}.$$

En inversant la première somme avec les autres, on obtient la nouvelle formule

$$S_\gamma = \sum_{t_1,\ldots,t_\gamma} p_{t_1}\ldots p_{t_\gamma}(2^{|t_1\cap\ldots\cap t_\gamma|}-1) \tag{2.5}$$

où $|t_1\cap\ldots\cap t_\gamma|$ est le nombre d'attributs simultanément présents dans tous les objets $o_1,\ldots,o_\gamma$.

Rappelons que dans ce modèle, m satisfait $m = K\cdot m_1$. Chaque objet est décomposé en paquets $o_j = (o_{j,1},\ldots,o_{j,m_1})$ tels que $o_{j,i}\subseteq[1+(i-1)K, iK]$. La somme S_γ est réalisée sur tous les paquets. Elle admet une formule matricielle alternative.

Lemme 5 *Soit $\tilde{P}_\gamma = (p_{(v_1,\ldots,v_\gamma),(w_1,\ldots,w_\gamma)})$ la matrice dont les coefficients satisfont*

$$p_{(v_1,\ldots,v_\gamma),(w_1,\ldots,w_\gamma)} = p_{w_1\to v_1}\ldots p_{w_\gamma\to v_\gamma}2^{|v_1\cap\ldots\cap v_\gamma|},$$

avec w_i et v_i dans $\{0,1\}^K$. Le vecteur colonne $\tilde{f}_{init}$ est donné par

$$\tilde{f}_{init} = (f_{v_1}\ldots f_{v_\gamma}2^{|v_1\cap\ldots\cap v_\gamma|})_{(v_1,\ldots,v_\gamma)\in(\{0,1\}^K)^\gamma}$$

où $f_{init} = (f_v)_{v\in\{0,1\}^K}$.
La somme S_γ admet donc l'expression alternative

$$S_\gamma = <\tilde{P}_\gamma^{m_1-1}\tilde{f}_{init}|\mathbf{1}> \tag{2.6}$$

où $<\cdot|\cdot>$ est le produit scalaire et $\mathbf{1}$ est le vecteur colonne constant égal à 1.

Comme $\tilde{P}_\gamma$ est une matrice strictement positive, elle admet une unique valeur propre dominante $\lambda(\gamma)$. Il n'est pas difficile de prouver la propriété suivante de $\lambda(\gamma)$.

Lemme 6

1. *La valeur propre dominante $\lambda(\gamma)$ de $\tilde{P}_\gamma$ est strictement supérieure à 1 ;*
2. $\lambda(\gamma) > \lambda(\gamma+1)$.

Les propriétés spectrales de $\tilde{P}_\gamma$ fournissent une équivalence pour S_γ,

$$S_\gamma = c_1 \lambda(\gamma)^{m_1} + O(\mu^{m_1})$$

où c_1 est constant et relatif à $\tilde{f}_{init}$ et à son objet spectral dominant. De plus, μ est le module de la valeur propre sous dominante.

Maintenant que S_γ est estimée, le théorème 2 suit facilement.

2.4 Preuve du théorème 7

Ce théorème est prouvé uniquement pour le modèle de Bernoulli. Le lemme suivant donne la formule exacte du nombre moyen de motifs fermés.

Lemme 7 *Le nombre moyen de motifs fermés $C_{m,n,\in}$ dans un modèle de* Bernoulli *de paramètre p satisfait*

$$C_{m,n,\gamma} = \sum_{j=1}^{m} \binom{m}{j} \sum_{i=\gamma}^{n} \binom{n}{i} p^{ij} (1-p^i)^{m-j} (1-p^j)^{n-i}.$$

Preuve : Soit A un motif fermé et B son support ou extension. La probabilité que tous les attributs de A appartiennent à tous les objets de B est $p^{|A|\cdot|B|}$. Celle que les $(n-|B|)$ objets n'appartenant pas B ne contiennent pas A est $(1-p^{|A|})^{n-|B|}$. Finalement, la probabilité que les $(n-|A|)$ attributs absents de A n'appartiennent pas aux objets de B est $(1-p^{|B|})^{m-|A|}$. La somme sur tous les motifs fermés et leurs fréquences prouve le lemme. □

Preuve : [Théorème 7] Rappelons que le nombre moyen de motifs fréquents est donné par la formule 2.1. Dans le modèle de Bernoulli, elle devient

$$F_{m,n,\gamma} = \sum_{j=1}^{m} \binom{m}{j} \sum_{i=\gamma}^{n} \binom{n}{i} p^{ij} (1-p^j)^{n-i}. \tag{2.7}$$

La seule différence entre $F_{m,n,\gamma}$ et $C_{m,n,\gamma}$ est le coefficient $(1-p^i)^{m-j}$. Il est borné par 1 et $(1-p^\gamma)^m$. Mais si γ est supérieur à $(1+\epsilon)\frac{\log m}{|\log p|}$, sa borne inférieure satisfait $(1-p^\gamma)^m = 1 + O(m^{-\epsilon})$, ce qui est suffisant pour prouver l'équivalence. □

Chapitre 3

Résultats d'expérience pour la section 7.4

Cette annexe rapporte des expériences menées sur des données de test de l'UCI [Blake et Merz, 1998]. Suivant le protocole décrit à la section 7.4, les règles d'association généralisées correctes, communes à la base de test et sa version opposée, sont recensées. On dispose donc du nombre maximum de règles communes.

Ensuite, des valeurs manquantes sont introduites et la partie gauche des figures indique la proportion de règles correctes qui sont informatives dans la base complète. La partie droite indique la même quantité relativement au nombre maximum de règles communes entre r et ^{-}r. Sur chaque figure, deux courbes sont utilisées pour différencier les stratégies utilisées pour le calcul des fermetures généralisées. Rappelons que la stratégie S_1 considère qu'une valeur manquante est présente pour effectuer le calcul des traverses minimales, tandis que S_2 suppose qu'elle est absente.

Lorsque la courbe représentant la proportion du nombre de règles correctes de $mv(r)$ qui sont informatives dans r s'arrête ou diverge, c'est que l'algorithme ne trouve aucune règle correcte. Cela se produit fréquemment avec la combinaison $S_2 - S_2$.

Pour toutes ces bases sauf `iris` et `zoo`, la proportion de règles correctes est plus importante avec la combinaison $S_2 - S_2$ (partie gauche des figures). Dans tous les cas cependant, cette combinaison fournit une quantité plus faible de règles (partie droite des figures).

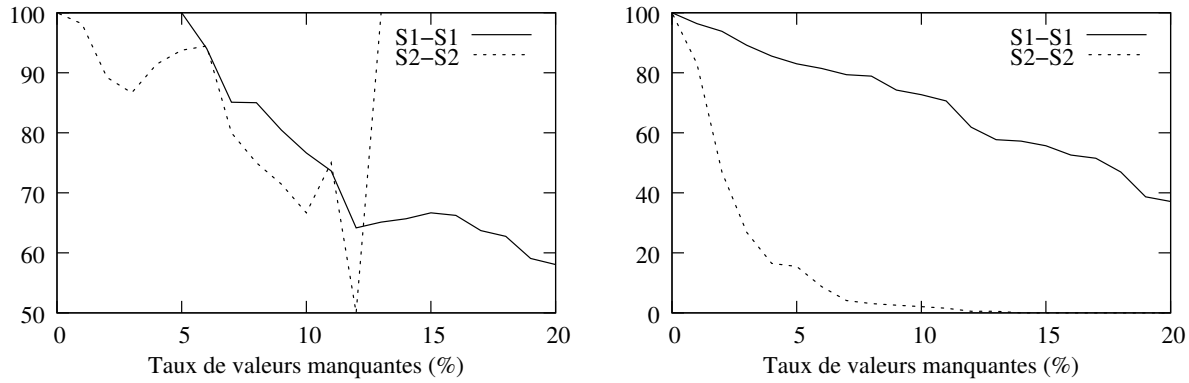

FIG. 3.1 – Proportion et quantité de règles correctes de mv(`iris`) qui sont informatives dans `iris` (100 % = 194 règles).

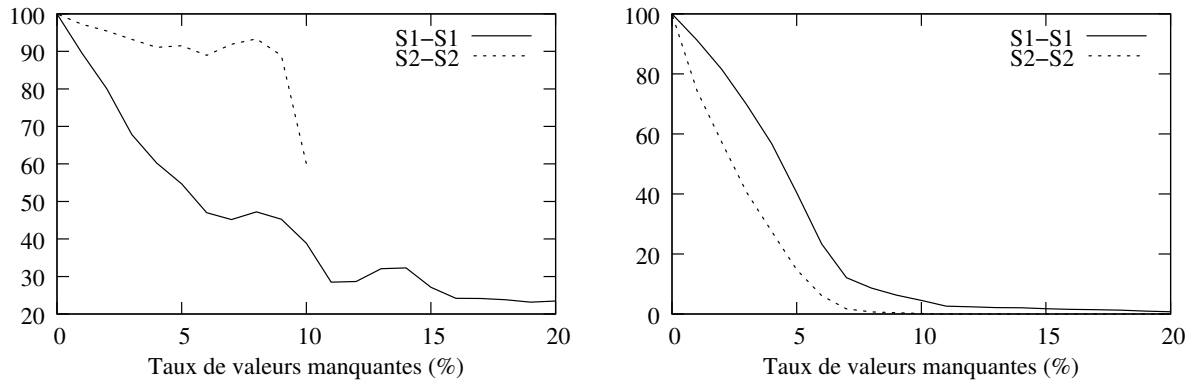

FIG. 3.2 – Proportion et quantité de règles correctes de mv(`liver – disorders`) qui sont informatives dans `liver – disorders` (100 % = 1969 règles).

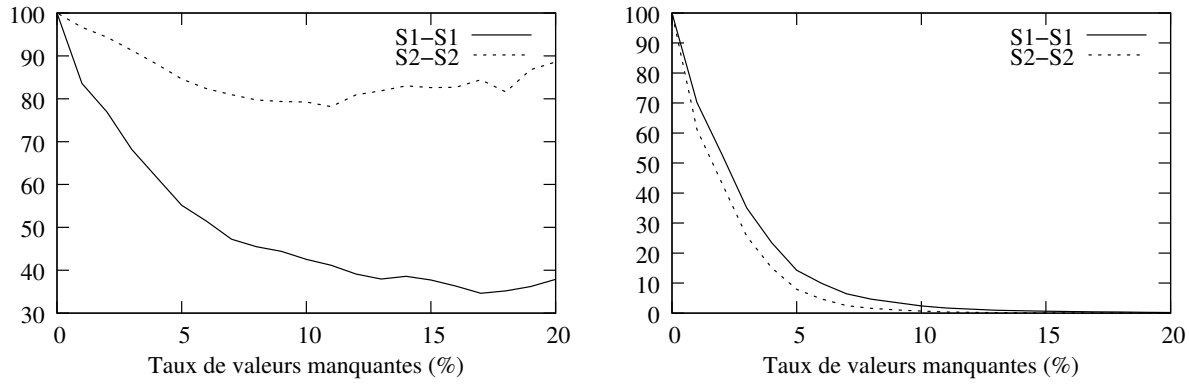

FIG. 3.3 – Proportion et quantité de règles correctes de mv(`lymphography`) qui sont informatives dans `lymphography` (100 % = 292 525 règles).

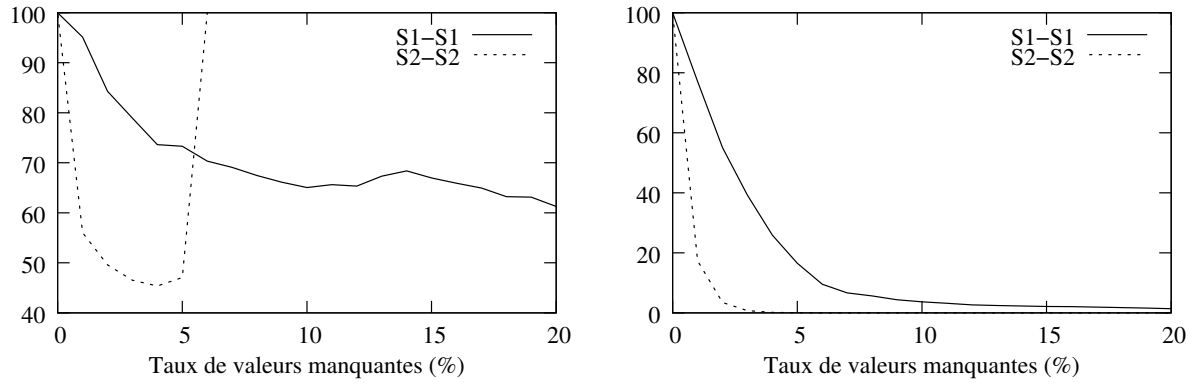

FIG. 3.4 – Proportion et quantité de règles correctes de $mv(\texttt{page} - \texttt{blocks})$ qui sont informatives dans $\texttt{page} - \texttt{blocks}$ (100 % = 24 494 règles).

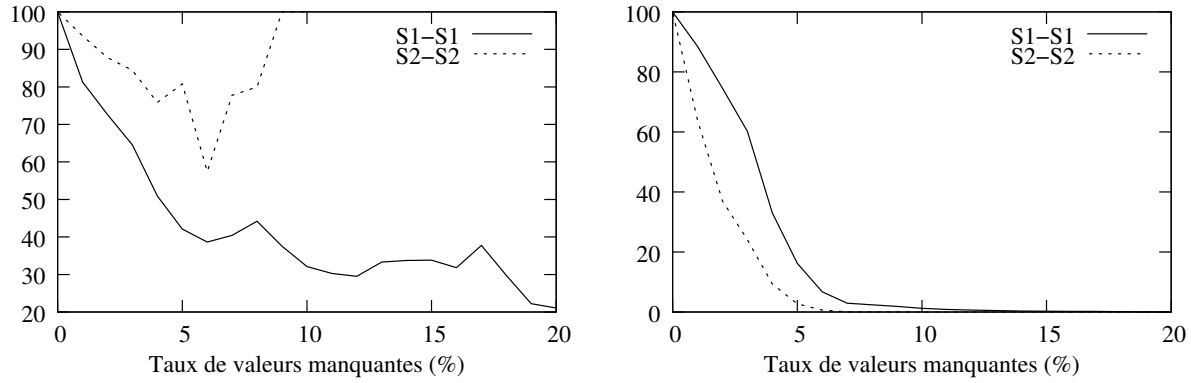

FIG. 3.5 – Proportion et quantité de règles correctes de $mv(\texttt{pima} - \texttt{indians} - \texttt{diabetes})$ qui sont informatives dans $\texttt{pima} - \texttt{indians} - \texttt{diabetes}$ (100 % = 9 540 règles).

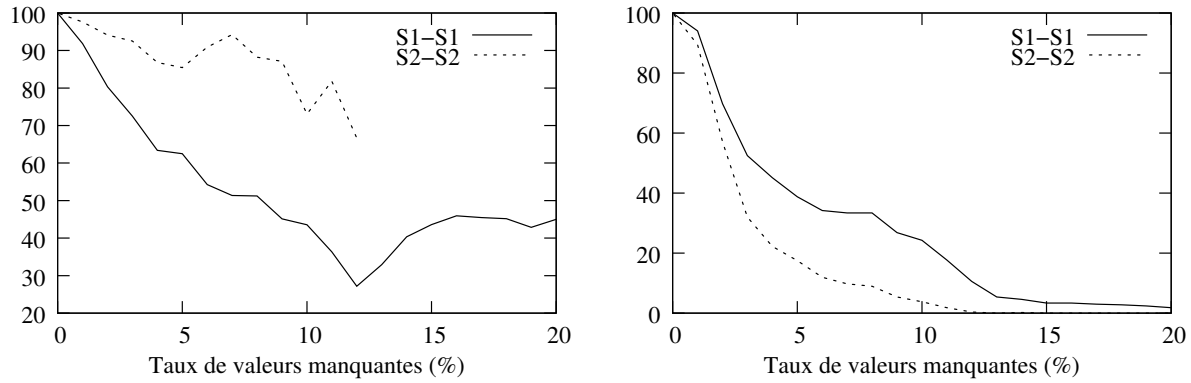

FIG. 3.6 – Proportion et quantité de règles correctes de $mv(\texttt{servo})$ qui sont informatives dans $\texttt{servo}$ (100 % = 503 règles).

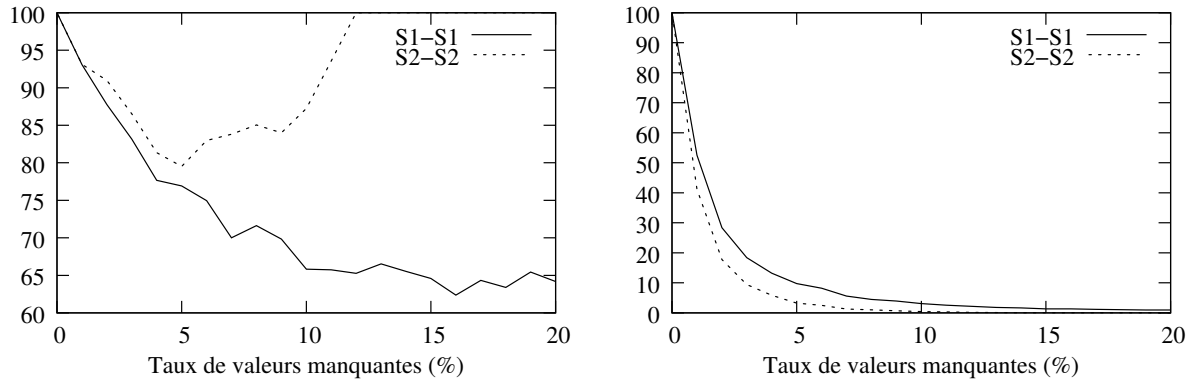

FIG. 3.7 – Proportion et quantité de règles correctes de $mv(\texttt{solar} - \texttt{flare})$ qui sont informatives dans $\texttt{solar} - \texttt{flare}$ (100 % = 9 193 règles).

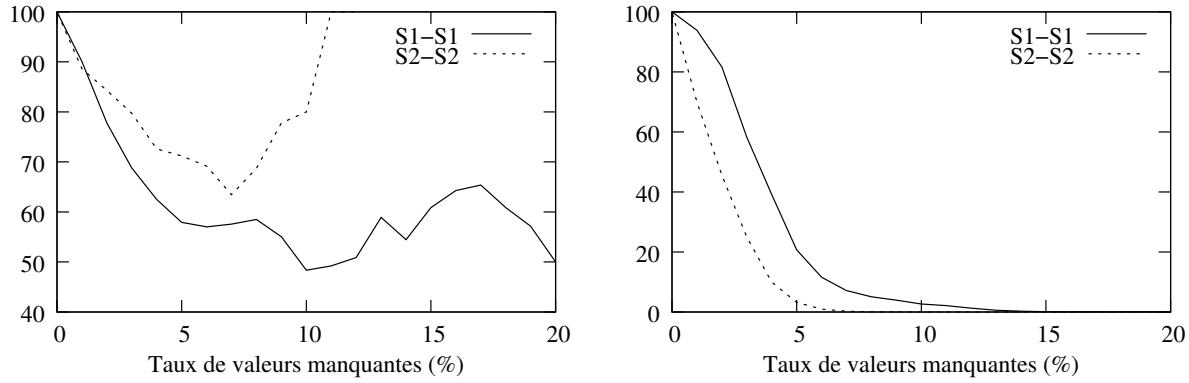

FIG. 3.8 – Proportion et quantité de règles correctes de $mv(\texttt{tic} - \texttt{tac} - \texttt{toe})$ qui sont informatives dans $\texttt{tic} - \texttt{tac} - \texttt{toe}$ (100 % = 22 638 règles).

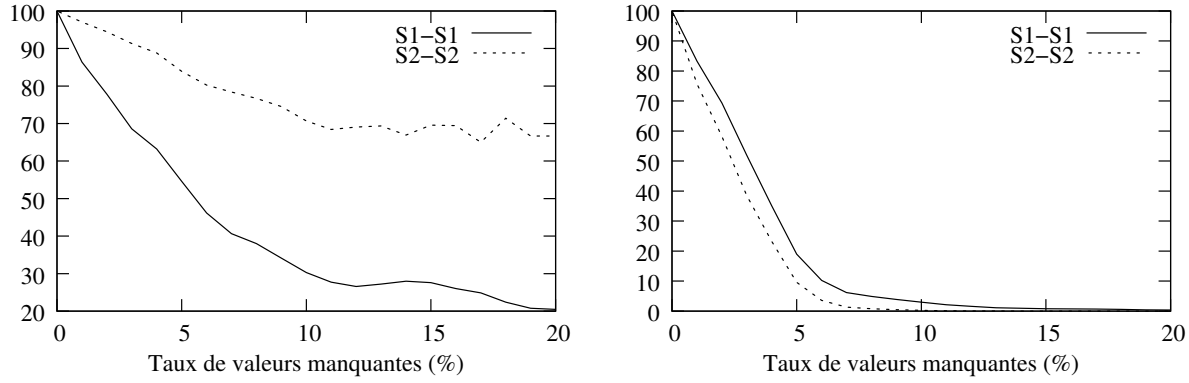

FIG. 3.9 – Proportion et quantité de règles correctes de $mv(\texttt{wine})$ qui sont informatives dans $\texttt{wine}$ (100 % = 633 611 règles).

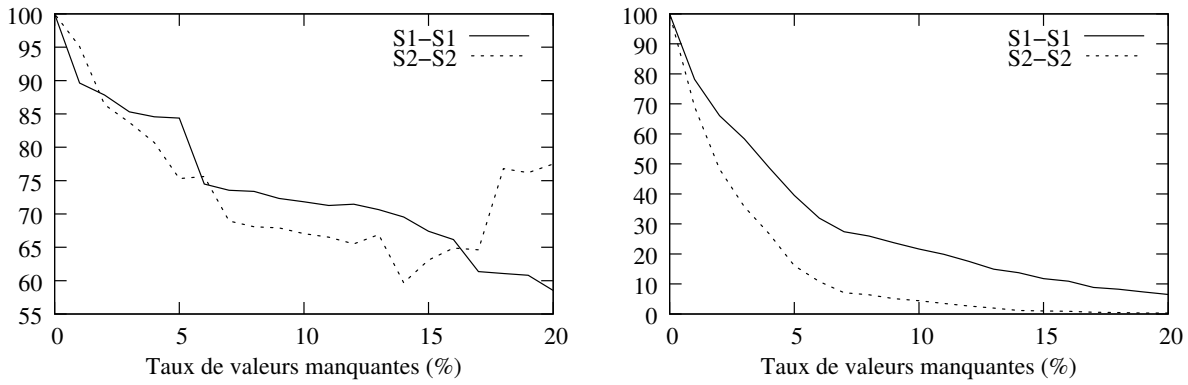

FIG. 3.10 – Proportion et quantité de règles correctes de mv(zoo) qui sont informatives dans zoo (100 % = 9 263 règles).

Bibliographie

[Agarwal *et al.*, 2001] Agarwal, R. C., Aggarwal, C. C. et Prasad, V. V. V. (2001). A tree projection algorithm for generation of frequent item sets. *In Journal of Parallel and Distributed Computing 61-3 Special issue on high-performance data mining*, pages 350–371.

[Agarwal, 2001] Agarwal, S. (2001). Learning from incomplete data. *In http ://www.cs.ucsd.edu/users/elkan/254spring01/sagarwalrep.pdf.*

[Agrawal *et al.*, 1993] Agrawal, R., Imielinski, T. et Swami, A. (1993). Mining association rules between sets of items in large databases. *In Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, USA*, pages 207–216.

[Agrawal *et al.*, 1996] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. et Verkamo, A. (1996). Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, pages 307–328.

[Agrawal et Srikant, 1994] Agrawal, R. et Srikant, R. (1994). Fast algorithms for mining association rules. *In Intl. Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile*, pages 487–499.

[Ahmed *et al.*, 2003] Ahmed, S., Coenen, F. et Leng, P. (2003). Strategies for partitioning data in association rule mining. *In The Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (SGAI-AI'03), Cambridge, UK.*

[Antonie et Zaïane, 2004a] Antonie, M.-L. et Zaïane, O. (2004a). An associative classifier based on positive and negative rules. *In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'04), Paris, France.*

[Antonie et Zaïane, 2004b] Antonie, M.-L. et Zaïane, O. (2004b). Mining positive and negative association rules : An approach for confined rules. *In Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04), Pisa, Italy*, pages 27–38.

[Armstrong *et al.*, 2002] Armstrong, W., Nakamura, Y. et Rudnicki, P. (2002). Armstrong's axioms. *In Journal of formalized mathematics, vol. 14.*

[Bastide, 2000] Bastide, Y. (2000). *Data Mining : algorithmes par niveaux, techniques d'implantation et applications*. Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand II, France.

[Bastide *et al.*, 2000] Bastide, Y., Taouil, R., Pasquier, N., Stumme, G. et Lakhal, L. (2000). Mining minimal non-redundant association rules using frequent closed itemsets. *In International Conference on Deductive and Object Databases (DOOD'00)*, pages 972–986.

[Bastide *et al.*, 2002] Bastide, Y., Taouil, R., Pasquier, N., Stumme, G. et Lakhal, L. (2002). Pascal : un algorithme d'extraction des motifs fréquents. *Technique et science informatiques*, 21(1):65–95.

[Bayardo, 1998] Bayardo, R. (1998). Efficiently mining long patterns from databases. *In ACM SIGMOD International Conference on Management, Seattle, Washington, USA*, pages 85–93.

[Bayardo *et al.*, 1999] Bayardo, R. J., Agrawal, R. et Gunopulos, D. (1999). Constraint-based rule mining in large, dense databases. *In Proc. of the 15th Int'l Conf. on Data Engineering, Sydney, Austrialia*, pages 188–197.

[Becquet *et al.*, 2002] Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.-F. et Gandrillon, O. (2002). Strong association rule mining for large gene expression data analysis : a case study on human SAGE data. *Genome Biology*, 12.

[Berge, 1989] Berge, C. (1989). *Hypergraphs*. North Holland, Amsterdam.

[Besson *et al.*, 2004] Besson, J., Rioult, F., Crémilleux, B., Rome, S. et Boulicaut, J.-F. (2004). *Informatique pour l'analyse du transcriptome*, chapitre Solutions pour le calcul d'ensembles fréquents dans des données biopuces, pages 231–254. Hermès.

[Birkhoff, 1967] Birkhoff, G. (1967). Lattice theory. *In American Mathematical Society, vol. 25*.

[Blake et Merz, 1998] Blake, C. et Merz, C. (1998). UCI repository of machine learning databases.

[Bonchi *et al.*, 2003a] Bonchi, F., Giannotti, F., Mazzanti, A. et Pedreschi, D. (2003a). Examiner : Optimized level-wise frequent pattern mining with monotone constraints. *In IEEE International Conference on Data Mining (ICDM'03), Melbourne, USA*, pages 11–18.

[Bonchi *et al.*, 2003b] Bonchi, F., Giannotti, F., Mazzanti, A. et Pedreschi, D. (2003b). Exante : Anticipated data reduction in constrained pattern mining. *In Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), Cavtat-Dubrovnik, Croatia*, pages 47–58.

[Borgelt, 2003] Borgelt, C. (2003). Efficient implementations of apriori and eclat. *In Workshop of Frequent Item Set Mining Implementations co-located with ICDM (FIMI'03), Melbourne, USA*.

[Borgelt et Kruse, 2002] Borgelt, C. et Kruse, R. (2002). Induction of association rules : Apriori implementation. *In 15th Conference on Computational Statistics, Berlin, Germany*, pages 395–400.

[Boros *et al.*, 2003] Boros, E., Elbassioni, K., Gurvich, V. et Khachiyan, L. (2003). An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals. *In 11th Annual European Symposium on Algorithms (ESA'03), Budapest, Hungary.*

[Boros *et al.*, 2002] Boros, E., Gurvich, V., Khachiyan, L. et Makino, K. (2002). On the complexity of generating maximal frequent and minimal infrequent sets. *In Symposium on Theoretical Aspects of Computer Science (STACS'02), Antibes - Juan les Pins, France*, pages 133–141.

[Bosc *et al.*, 2002] Bosc, P., Cholvy, L., Dubois, D., Mouaddib, N., Pivert, O., Prade, H., Raschia, G. et Rousset, M.-C. (2002). Les informations incomplètes dans les bases de données et en intelligence artificielle. *In Actes des 2è assises nationales du GRD i3.*

[Bosc *et al.*, 2004] Bosc, P., Liétard, L., Pivert, O. et Rocacher, D. (2004). *Gradualité et imprécision dans les bases de données.* Ellipses Marketting, Technosup.

[Boulicaut et Jeudy, 2000] Boulicaut, J. et Jeudy, B. (2000). Using constraints during set mining : Should we prune or not ? *In Journées Bases de Données Avancées (BDA'00), Blois, France.*

[Boulicaut et Bykowski, 2000] Boulicaut, J.-F. et Bykowski, A. (2000). Frequent closures as a concise representation for binary data mining. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00), Kyoto, Japan*, pages 62–73.

[Boulicaut *et al.*, 2000] Boulicaut, J.-F., Bykowski, A. et Rigotti, C. (2000). Approximation of frequency queries by means of free-sets. *In Principles of Data Mining and Knowledge Discovery (PKDD'00), Lyon, France*, pages 75–85.

[Boulicaut *et al.*, 2003] Boulicaut, J.-F., Bykowski, A. et Rigotti, C. (2003). Free-sets : a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, pages 5–22. Kluwer Academics Publishers.

[Boulicaut *et al.*, 1999] Boulicaut, J.-F., Klemettinen, M. et Mannila, H. (1999). Modeling kdd processes within the inductive database framework. *In Data Warehousing and Knowledge Discovery (DaWak'99), Florence, Italy*, pages 293–302.

[Bucila *et al.*, 2002] Bucila, C., Gehrke, J., Kifer, D. et White, W. (2002). Dualminer : a dual-pruning algorithm for itemsets with constraints. *In the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), Edmonton, Canada*, pages 42–51.

[Bykowski et Rigotti, 2001] Bykowski, A. et Rigotti, C. (2001). A condensed representation to find frequent patterns. *In ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Santa Barbara, USA*, pages 267–273.

[Bykowski et Rigotti, 2003] Bykowski, A. et Rigotti, C. (2003). Dbc : a condensed representation of frequent patterns for efficient mining. *Information Systems (IS), Elsevier Science*, 28(8):949–977.

[Calders et Goethals, 2002] Calders, T. et Goethals, B. (2002). Mining all non-derivable frequent itemsets. *In Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), Helsinki, Finland.*

[Calders et Goethals, 2003] Calders, T. et Goethals, B. (2003). Minimal k-free representations of frequent sets. *In Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), Cavtat-Dubrovnik, Croatia*, pages 71–82.

[Camargo et al., 2001] Camargo et al., A. (2001). The contribution of 700,000 orf sequence tags to the definition of the human transcriptome. *In Proc. Natl. Acad. Sci. USA, 98, Genie*, pages 12103–12108.

[Codd, 1979] Codd, E. (1979). Extending the data relational model to capture more meaning. *Transactions of ACM on database systems*, 4(4).

[Cosset *et al.*, 1992] Cosset, J. M., Henry-Amar, M., Meerwarldt, J. K., Carde, P., Noordijk, E. M., Thomas, J., Burgers, J. M. V., Somers, R., Hayat, M. et Tubiana, M. (1992). The EORTC trials for limited stage Hodgkin's disease. *Eur J Cancer*, 28A:1847–1850.

[Courtine, 2002] Courtine, M. (2002). *Changements de représentation pour la classification conceptuelle non supervisée de données complexes.* Thèse de doctorat, Université Paris VI, France.

[Crémilleux et Boulicaut, 2002] Crémilleux, B. et Boulicaut, J. (2002). Simplest rules characterizing classes generated by delta-free sets. *In* Springer, éditeur : *International Conference on Knowledge Based Systems and Applied Artificial Intelligence (Expert System), Cambridge, UK*, pages 33–46.

[De Raedt *et al.*, 2002] De Raedt, L., Jaeger, M., Lee, S. et Mannila, H. (2002). A theory of inductive query answering (extended abstract). *In IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan*, pages 123–130.

[De Raedt et Kramer, 2001] De Raedt, L. et Kramer, S. (2001). The levelwise version space algorithm and its application to molecular fragment finding. *In International Joint Conference on Artificial Intelligence (IJCAI'01), Seattle, USA*, pages 853–862.

[Demetrovics et Thi, 1995] Demetrovics, J. et Thi, V. (1995). Some remarks on generating armstrong and inferring functionnal dependencies relation. *Acta Cybernetica*, 12(2):167–180.

[Dexters et Calders, 2004] Dexters, N. et Calders, T. (2004). Theoretical bounds on the size of condensed representations. *In ECML-PKDD 2004 Workshop on Knowledge Discovery in Inductive Databases (KDID'04), Pisa, Italy*, pages 25–36.

[Dong et Li, 1999] Dong, G. et Li, J. (1999). Efficient mining of emerging patterns : discovering trends and differences. *In Knowledge Discovery and Data Mining (KDD'99), San Diego, USA*, pages 43–52.

[Durand, 2004] Durand, N. (2004). *Extraction de clusters à partir du treillis de concepts : application à la découverte de communautés d'intérêt pour améliorer l'accès à l'information.* Thèse de doctorat, Université de Caen Basse-Normandie, France.

[Durand *et al.*, 2004] Durand, N., Cleuziou, G. et Soulet, A. (2004). Discovery of overlapping clusters to detect atherosclerosis risk factors. *In ECML/PKDD'04 Discovery Challenge on Atherosclerosis Data, Pisa, Italy*, pages 32–43.

[Durand et Crémilleux, 2002] Durand, N. et Crémilleux, B. (2002). Ecclat : a new approach of clusters discovery in categorical data. *In International Conference on Knowledge Based Systems and Applied Artificial Intelligence (Expert System), Cambridge, UK*, pages 177–190.

[Durand *et al.*, 2003] Durand, N., Lancieri, L. et Crémilleux, B. (2003). Recommendation system based on the discovery of meaningful categorical clusters. *In International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2003), Oxford, UK*, pages 857–863.

[Dyreson, 1997] Dyreson, C. E. (1997). *Uncertainty Management in Information Systems*, chapitre A Bibliography on Uncertainty Management in Information Systems. Kluwer Academic Publishers.

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

[Eiter *et al.*, 2002] Eiter, T., Gottlob, G. et Makino, K. (2002). New results on monotone dualization and generating hypergraph transversals. *In Annual ACM Symposium on Theory of Computing (STOC'02), Montréal, Canada*, pages 14–22.

[Fayyad *et al.*, 1996] Fayyad, U., Piatetsky-Shapiro, G. et Smyth, P. (1996). Knowledge discovery and data mining : Towards a unifying framework. *In International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, USA*, pages 82–88.

[Fermé *et al.*, 2000] Fermé, C., Eghbali, H., Hagenbeek, A., Brice, P., Meder, J., Carde, P., Lederlin, P., Cosset, J., Gerrits, W., Coiffier, B., Mandard, A., Rieux, C. et Henry-Amar, M. (2000). Mopp/abv hybrid and irradiation in unfavorable supradiaphragmatic clinical stages i-ii hodgkin's disease : Comparison of three treatment modalities. preliminary results of the eortc-gela h8-u randomized trial in 995 patients. *42nd Annual Meeting of the American Society of Hematology*, 96(11).

[Fredman et Kachiyan, 1996] Fredman, M. et Kachiyan, L. (1996). On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(2):618–628.

[Fu, 2005] Fu, H. (2005). *Algorithmique des Treillis de Concepts : Application à la Fouille de Données*. Thèse de doctorat, Université d'Artois, France.

[Fu et Mephu Nguifo, 2003] Fu, H. et Mephu Nguifo, E. (2003). How well go lattice algorithms on currently used machine learning testbeds ? *In First International Conference on Formal Concept Analysis((ICFCA), Darmstadt, Germany.*

[Fu et Mephu Nguifo, 2004] Fu, H. et Mephu Nguifo, E. (2004). Étude et conception d'algorithmes de génération de concepts formels. *Revue des sciences et technologies de l'information série Ingéniérie des systèmes d'information (RSTI-ISI)*, 9:109–132.

[Ganter, 1984] Ganter, B. (1984). Two basic algorithms in concept analysis. *In Preprint 831, Technische Hochschule Darmstadt.*

[Ganter et Wille, 1989] Ganter, B. et Wille, R. (1989). *Applications of combinatorics and graph theory to the biological and social sciences*, chapitre Conceptual scaling, pages 139–167. F. Roberts (ed.), Springer, NewYork.

[Ganter et Wille, 1999] Ganter, B. et Wille, R. (1999). *Mathematical Foundations*, chapitre Concept Analysis. Springer, Berlin-Heidelberg-New York.

[Geerts *et al.*, 2001] Geerts, F., Goethals, B. et Van den Bussche, J. (2001). A tight upper bound on the number of candidate patterns. *In IEEE International Conference on Data Mining (ICDM'01), San Jose, USA*, pages 155–162.

[Giacometti *et al.*, 2002] Giacometti, A., Laurent, D. et Diop, C. (2002). Condensed representations for sets of mining queries. *In First International PKDD Workshop on Knowledge Discovery in Inductive Databases (KDID´02), Helsinki, Finland.*

[Godin *et al.*, 1995] Godin, R., Mineau, G., Missaoui, R. et Mili, H. (1995). Méthodes de classification conceptuelle basées sur les treillis de galois et applications. *Revue d'intelligence artificielle*, 9(2).

[Goethals et Van den Bussche, 2000] Goethals, B. et Van den Bussche, J. (2000). On supporting interactive association rule mining. *In Data Warehousing and Knowledge Discovery (DAWAK'00), London, UK*, pages 307–316.

[Grzymala-Busse, 2002] Grzymala-Busse, J. (2002). *Handbook of Data Mining and Knowledge Discovery*, chapitre Discretization of numerical attributes. Oxford University Press.

[Grzymala-Busse et Hu, 2001] Grzymala-Busse, J. et Hu, M. (2001). A comparison of several approaches to missing attribute values in data mining. *In RSCTC '00 : Revised Papers from the Second International Conference on Rough Sets and Current Trends in Computing, London, UK*, pages 378–385.

[Gunopulos *et al.*, 1997a] Gunopulos, D., Mannila, H., Khardon, R. et Toivonen, H. (1997a). Data mining, hypergraph transversals, and machine learning. *In ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97), Tucson, USA*.

[Gunopulos *et al.*, 1997b] Gunopulos, D., Mannila, H. et Saluja, S. (1997b). Discovering all most specific sentences by randomized algorithms. *In ICDT*, pages 215–229.

[Guénoche, 1993] Guénoche, A. (1993). Construction du treillis de galois d'une relation binaire. *Mathématique, Informatique et Sciences Humaines*, 121:23–34.

[Hagenbeek *et al.*, 2000] Hagenbeek, A., Eghbali, H., Fermé, C., Meerwaldt, J., Diviné, M., Raemaekers, J., Reman, O., Zagonel, V., Ferrant, A., Gabarre, J., Berger, F., Rieux, C. et M., M. H.-A. (2000). Three cycles of mopp/abv hybrid and involved-field irradiation is more effective than subtotal nodal irradiation in favorable supradiaphragmatic clinical stages i-ii hodgkin's disease : Preliminary results of the eortc-gela h8-f randomized trial in 543 patients. *42nd Annual Meeting of the American Society of Hematology*, 96(11).

[Hamrouni *et al.*, 2005] Hamrouni, T., Ben Yahia, S. et Slimani, Y. (2005). Prince : Extraction optimisée des bases génériques de règles sans calcul de fermetures. *In Congrès INFORSID 2005*, pages 353–368.

[Han *et al.*, 2000] Han, J., Pei, J. et Yin, Y. (2000). Mining frequent patterns without candidate generation. *In ACM SIGMOD International Conference on Management of Data (SIGMOD'00), Dallas, USA*, pages 1–12.

[Hasegawa *et al.*, 2004] Hasegawa, K., Ono, T., Matsushita, H., Shimono, M., Noguchi, Y., Mizutani, Y., Kodama, J., Kudo, T. et Nakayama, E. (2004). A-kinase anchoring protein 3 messenger rna expression in ovarian cancer and its implication on prognosis. *In Int Journal of Cancer, 108, 86-90.*

[Hébert et Bretto, 2005] Hébert, C. et Bretto, A. (2005). A level-wise algorithm for computing hypergraph transversals. *In ACM-SIAM Symposium on Discrete Algorithms (SODA'06), submitted.*

[Hébert et Crémilleux, 2005] Hébert, C. et Crémilleux, B. (2005). Mining frequent δ-free patterns in large databases. *In Discovery Science'05, accepted.*

[Hipp *et al.*, 2000a] Hipp, J., Güntzer, U. et Nakhaeizadeh, G. (2000a). Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explorations*, 2:58–64.

[Hipp *et al.*, 2000b] Hipp, J., Güntzer, U. et Nakhaeizadeh, G. (2000b). Mining association rules : deriving a superior algorithm by analysing today's approaches. *In Principles of Data Mining and Knowledge Discovery (PKDD '00), Lyon, France.*

[Houben, 2004] Houben, F. (2004). Mot vide, mot plein ? comment trancher localement. *In Récital'04.*

[Houben et Rioult, 2005] Houben, F. et Rioult, F. (2005). Généralisation d'étiquetage morpho-syntaxique par classification supervisée. *In Atelier Langues peu dotées, TALN-RECITAL'05, Dourdan, France*, pages 239–248.

[Hough *et al.*, 2000] Hough, C., Sherman-Baust, C., Pizer, E., Montz, F., Im, D., Rosenshein, N., Cho, K., Riggins, G. et Morin, P. (2000). Large-scale serial analysis of gene expression reveals genes differentially expressed in ovarian cancer. *In Cancer Res, 60, 6281-6287.*

[Imielinski et Mannila, 1996] Imielinski, T. et Mannila, H. (1996). A database perspective on knowledge discovery. *Communication of the ACM*, 39(11):58–64.

[Jami *et al.*, 2004] Jami, S., Jen, T., Laurent, D., Loizou, G. et Sy, O. (2004). Extraction de règles d'association pour la prédiction de valeurs manquantes. *In Colloque Africain sur la Recherche en Informatique (CARI).*

[Jami *et al.*, 1998] Jami, S., Liu, X. et Loizou, G. (1998). Learning from an incomplete and uncertain data set : The identification of variant haemoglobins. *In Workshop on Intelligent Data Analysis in Medicine and Pharmacology, European Conference on Artificial Intelligence, Brighton, UK.*

[Jaroszewicz et Simovici, 2002] Jaroszewicz, S. et Simovici, D. (2002). Support approximations using bonferroni-type inequalities. *In Principles of Data Mining and Knowledge Discovery (PKDD'02), Helsinki, Finland*, pages 212–224.

[Jeudy, 2002] Jeudy, B. (2002). *Optimisation de requêtes inductives : application à l'extraction sous contrainte de règles d'association.* Thèse de doctorat, INSA de Lyon.

[Jeudy et Rioult, 2005a] Jeudy, B. et Rioult, F. (2005a). Extraction de concepts sous contraintes dans des données d'expression de gènes. *In Conférence d'Apprentissage (CAp'05), Nice, France*, pages 265–280.

[Jeudy et Rioult, 2005b] Jeudy, B. et Rioult, F. (2005b). *Post-proceedings of the International Workshop on Knowledge Discovery in Inductive Databases (KDID'04) co-located with the ECML-PKDD'04*, chapitre Database Transposition for Constrained (Closed) Pattern Mining, pages 89–107. Springer.

[Kavvadias et Stavropoulos, 2001] Kavvadias, D. et Stavropoulos, E. (2001). The nondeterministic complexity of the fredman and kachiyan algorithm for the dualization of monotone dnfs. Rapport technique, Computer Technology Institute of Patras, Greece.

[Kavvadias et Stavropoulos, 1999a] Kavvadias, D. J. et Stavropoulos, E. C. (1999a). Evaluation of an algorithm for the transversal hypergraph problem. *In Algorithm Engineering, 3rd International Workshop, WAE '99, London, UK*, volume 1668, pages 72–84.

[Kavvadias et Stavropoulos, 1999b] Kavvadias, D. J. et Stavropoulos, E. C. (1999b). A new algorithm for the transversal hypergraph problem. Rapport technique Technical Report CTI TR990303, Computer Technology Institute, Patras, Greece.

[Kavvadias et Stavropoulos, 2003a] Kavvadias, D. J. et Stavropoulos, E. C. (2003a). Checking monotone boolean duality with limited nondeterminism. Rapport technique, University of Patras, Greece.

[Kavvadias et Stavropoulos, 2003b] Kavvadias, D. J. et Stavropoulos, E. C. (2003b). Monotone boolean dualization is in $co - np[\log^2 n]$. *Information Processing Letters*, 85(1):1–6.

[Kerdprasop *et al.*, 2003] Kerdprasop, N., Kerdprasop, K. et Pumrungreong, P. (2003). A comparative study of techniques to handle missing values in the classification task of data mining. *In 29th congress on Science and Technology of Thailand.*

[Kryszkiewicz, 2004] Kryszkiewicz, M. (2004). Reducing borders of k-disjunction free representations of frequent patterns. *In ACM Symposium on Applied Computing (SAC'04), Nicosia, Cyprus*, pages 559–563.

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

[Latkowski, 2003] Latkowski, R. (2003). On decomposition for incomplete data. *Fundam. Inf.*, 54(1):1–16.

[Lee et De Raedt, 2003] Lee, S. et De Raedt, L. (2003). An algebra for inductive query evaluation. *In Workshop on Knowledge Discovery in Inductive Databases (KDID'03) co-located with ECML/PKDD'03, Cavtat-Dubrovnik, Croatia.*

[Levene et Loizou, 1999] Levene, M. et Loizou, G. (1999). Database design for incomplete relations. *ACM Transactions on Database Systems*, 24(1):80–126.

[Lhote *et al.*, 2005a] Lhote, L., Rioult, F. et Soulet, A. (2005a). Average number of frequent and closed patterns in random databases. *In Conférence d'Apprentissage (CAp'05), Nice, France*, pages 345–360.

[Lhote *et al.*, 2005b] Lhote, L., Rioult, F. et Soulet, A. (2005b). Average number of frequent (closed) patterns in bernouilli and markovian databases. *In IEEE International Conference on Data Mining (ICDM'05), Houston, USA*, pages 713–716.

[Li *et al.*, 1999] Li, J., Zhang, X., Dong, G., Ramamohanarao, K. et Sun, Q. (1999). Efficient mining of high confidence association rules without support thresholds. *In Principles and Practice of Knowledge Discovery in Databases (PKDD'99), Prague, Czech Republic*, pages 406–411.

[Li, 2001] Li, W. (2001). *Classification based on multiple association rules.* Thèse de doctorat, Simon Fraser University, USA.

[Li *et al.*, 2001] Li, W., Han, J. et Pei, J. (2001). Cmar : Accurate and efficient classification based on multiple class-association rules. *In IEEE International Conference on Data Mining (ICDM'01), San Jose, USA.*

[Little et Rubin, 1985] Little, R. et Rubin, D. (1985). *Statistical analysis with missag data.* John Wiley and Sons.

[Liu *et al.*, 1998a] Liu, B., Hsu, W. et Ma, Y. (1998a). Integrating classification and association rules mining. *In International Conference on Knowledge Discovery and Data Mining (KDD'98), New York, USA*, pages 80–86.

[Liu *et al.*, 1997] Liu, B., Ku, L. et Hsu, W. (1997). Discovering interesting holes in data. *In International Joint Conference on Artificial Intelligence (IJCAI'97), Nagoya, Japan*, pages 930–935.

[Liu *et al.*, 1998b] Liu, B., Wang, K., Mun, L.-F. et Qi, X.-Z. (1998b). Using decision tree induction for discovering holes in data. *In Pacific Rim International Conference on Artificial Intelligence, Singapore*, pages 182–193.

[Mannila et Räihä, 1986] Mannila, H. et Räihä, K.-J. (1986). Inclusion dependencies : Application to logical database tuning. *In International Conference on Data Engineering, Los Angeles, USA.*

[Mannila et Toivonen, 1996] Mannila, H. et Toivonen, H. (1996). Multiple uses of frequent sets and condensed representations (extended abstract). *In International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, USA*, pages 189–194.

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

[Meo et Psaila, 2003] Meo, R. et Psaila, G. (2003). Toward xml-based knowledge discovery systems. *In IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan*, pages 665–668.

[Mephu Nguifo et Njiwoua, 2000] Mephu Nguifo, E. et Njiwoua, P. (2000). Glue : a lattice-based constructive induction system. *Intl. Journ. of Intelligent Data Analysis (IDA)*, 4(4):1–49.

[Michalski, 1983] Michalski, R. (1983). *Machine Learning. An artificial intelligence approach*, volume 1, chapitre A theory and methodology of inductive learning, pages 83–133. Tioga Publishing.

[Mitchell, 1980] Mitchell, T. (1980). Generalization as search. *Artificial Intelligence*, 18(2): 203–226.

[Nakache et Gueguen, 2000] Nakache, J.-P. et Gueguen, A. (2000). Analyse multidimensionnelle de données incomplètes. Rapport technique, CNRS/INSERM U88-IFR69.

[Nayak et Cook, 2001] Nayak, J. et Cook, D. (2001). Approximate association rule mining. *In Florida Artificial Intelligence Research Symposium, Key West, Florida, USA*, pages 259–263.

[Ng *et al.*, 1998] Ng, R., Lakshmanan, L., Han, J. et Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. *In ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, pages 13–24.

[Noordijk *et al.*, 2005] Noordijk, E., Thomas, J., Fermé, C., Eghbali, H., Diviné, M., Brice, P., Carde, P., Van 't Veer, M., van der Maazen, R. et Henry-Amar, M. (2005). First results of the eortc-gela randomized trials : The h9-f trial (comparing 3 radiation dose levels) and h9-u trial (comparing 3 different chemotherapy schemes) in patients with favorable or unfavorable early stage hodgkin's lymphoma (hl). *J Clin Oncol*, 23(16).

[Noordijk *et al.*, 1994] Noordijk, E. M., Carde, P., Mandard, A. M., Mellink, W. A. M., Monconduit, M., Eghbali, H., Tirelli, U., Thomas, J., Somers, R., Dupouy, N. et Henry-Amar, M. (1994). Preliminary results of the EORTC-GPMC controlled clinical trial H7 in early stage Hodgkin's disease. *Ann. Oncol.*, 5 (Suppl. 2):S107–S112.

[Pan *et al.*, 2003] Pan, F., Cong, G., Tung, A., Yang, J. et Zaki, M. (2003). Carpenter : Finding closed patterns in long biological datasets. *In International Conference on Knowledge Discovery and Data Mining, Washington, USA*, pages 637–642.

[Pasquier, 2000] Pasquier, N. (2000). *Data mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données.* Thèse de doctorat, Université de Clermont-Ferrand II, France.

[Pei *et al.*, 2001a] Pei, J., Han, J. et Lakshmanan, L. (2001a). Mining frequent item sets with convertible constraints. *In International Conference on Data Engineering, Heidelberg, Germany*, pages 433–442.

[Pei *et al.*, 2001b] Pei, J., Han, J., Lu, H., Nishio, S., Tang, S. et Yang, D. (2001b). H-mine : Hyper-structure mining of frequent patterns in large databases. *In IEEE International Conference in Data Mining (ICDM'01), San Jose, California*, pages 441–448.

[Pei *et al.*, 2000] Pei, J., Han, J. et Mao, R. (2000). Closet : An efficient algorithm for mining frequent closed itemsets. *In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'00), Dallas, USA*, pages 21–30.

[Petit *et al.*, 2004] Petit, J.-M., De Marchi, F. et Flouvat, F. (2004). Adaptive strategies for mining the positive border of sentences. *In Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany.*

[Purdom *et al.*, 2004] Purdom, P. W., Gucht, D. V. et Groth, D. P. (2004). Average-case performance of the apriori algorithm. *SIAM Journal on Computing*, 33(5):1223–1260.

[Quinlan, 1987] Quinlan, J. (1987). Unknown attribute values in induction. *In International Machine Learning Workshop Cornell, New York, USA*.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers.

[Ragel, 1999] Ragel, A. (1999). *Exploration des bases incomplètes. Application à l'aide au prétraitement des valeurs manquantes*. Thèse de doctorat, Université de Caen Basse-Normandie.

[Ragel et Crémilleux, 1998] Ragel, A. et Crémilleux, B. (1998). Treatment of missing values for association rules. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'98), Melbourne, Australia*, pages 258–270.

[Ragel et Crémilleux, 1999] Ragel, A. et Crémilleux, B. (1999). Mvc - a preprocessing method to deal with missing values. *Knowledge-Based Systems*, 12(5-6):285–291.

[Rioult, 2004a] Rioult, F. (2004a). Découverte de motifs fréquents dans les bases de données, un cadre formel pour les méthodes. *Revue des sciences et technologies de l'information série Ingéniérie des systèmes d'information (RSTI-ISI)*, 9:211–240.

[Rioult, 2004b] Rioult, F. (2004b). Mining strong emerging patterns in wide sage data. *In ECML/PKDD'04 Discovery Challenge, Pisa, Italy*, pages 127–138.

[Rioult *et al.*, 2003a] Rioult, F., Boulicaut, J.-F., Crémilleux, B. et Besson, J. (2003a). Using transposition for pattern discovery from microarray data. *In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03), San Diego, USA*, pages 73–79.

[Rioult *et al.*, 2005] Rioult, F., Constans, J.-M., Crémilleux, B. et Kauffmann, F. (2005). Données médicales hétérogènes : l'exemple des cytopathies mitochondriales. *In Atelier Fouille de données complexes de la conférence Extraction et Gestion des Connaissances (EGC)*, pages 85–88.

[Rioult et Crémilleux, 2003a] Rioult, F. et Crémilleux, B. (2003a). Condensed representations in presence of missing values. *In Symposium on Intelligent Data Analysis, Berlin, Germany*, pages 578–588.

[Rioult et Crémilleux, 2003b] Rioult, F. et Crémilleux, B. (2003b). Optimisation d'extraction de motifs : une nouvelle méthode fondée sur la transposition de données. *In Conférence d'Apprentissage, (CAp'03), Laval, France*, pages 299–313.

[Rioult et Crémilleux, 2004] Rioult, F. et Crémilleux, B. (2004). Représentation condensée en présence de valeurs manquantes. *In XXIIè congrès Inforsid, Biarritz, France*, pages 301–317.

[Rioult *et al.*, 2003b] Rioult, F., Robardet, C., Blachon, S., Crémilleux, B., Gandrillon, O. et Boulicaut, J.-F. (2003b). Mining concepts from large sage gene expression matrices. *In International Workshop on Knowledge Discovery in Inductive Databases KDID'03 co-located with ECML-PKDD'03, Cavtat-Dubrovnik (Croatia).*

[Rome *et al.*, 2003] Rome, S., Clément, K., Rabasa-Lhoret, R., Loizon, E., Poitou, C., Barsh, G. S., Riou, J., Laville, M. et Vidal, H. (2003). Microarray profiling of human skeletal muscle reveals that insulin regulates 800 genes during an hyperinsulinemic clamp. *In Journal of Biological Chemistry.*

[Sebag, 1994] Sebag, M. (1994). Une approche par contraintes de l'espace des versions. *In Reconnaissance des Formes et Intelligence Artificielle*, pages 275–283.

[Simon, 2000] Simon, A. (2000). *Outils classificatoires par objets pour l'extraction de connaissance dans les bases de données.* Thèse de doctorat, Université Henri-Poincaré, Nancy I, France.

[Soulet et Crémilleux, 2005] Soulet, A. et Crémilleux, B. (2005). An efficient framework for mining flexible constraints. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD'05), Hanoi, Vietnam.*

[Soulet *et al.*, 2004a] Soulet, A., Crémilleux, B. et Rioult, F. (2004a). Condensed representation of emerging patterns. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04), Sydney, Australia*, pages 127–132.

[Soulet *et al.*, 2004b] Soulet, A., Crémilleux, B. et Rioult, F. (2004b). Représentation condensée de motifs émergents. *In 4èmes journées d'Extraction et de Gestion des Connaissances (EGC'04), Clermont-Ferrand, France*, Revue des Nouvelles Technologies de l'Information, pages 265–276. Cepaduès Editions.

[Soulet *et al.*, 2005] Soulet, A., Crémilleux, B. et Rioult, F. (2005). *Post-proceedings of the International Workshop on Knowledge Discovery in Inductive Databases (KDID'04) co-located with the ECML-PKDD'04*, chapitre Condensed Representation of EPs and Patterns Quantified by Frequency-Based Measures, pages 173–190. Springer.

[Srikant et Agrawal, 1996] Srikant, R. et Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *In Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, Canada*, pages 1–12.

[Stadler et Stadler, 2002] Stadler, B. et Stadler, P. (2002). Basic properties of filter convergence spaces. *J.Chem.Inf.Comput.Sci.*, 42:577–585.

[Stumme *et al.*, 2002] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N. et Lakhal, L. (2002). Computing iceberg concept lattices with titanic. *Data & Knowledge Engineering*, 42(2):189–222.

[Toivonen, 1996] Toivonen, H. (1996). Sampling large databases for association rules. *In International Conference on Very Large Data Bases (VLDB'96), Mumbai, India*, pages 134–145. Morgan Kaufman.

[Tubiana *et al.*, 1989] Tubiana, M., Henry-Amar, M., Carde, P., Burgers, J., Hayat, M., Van der Schueren, E., Noordijk, E. M., Tanguy, A., Meerwaldt, J. H., Thomas, J., De Pauw, B., Cosset, J. M. et Somers, R. (1989). Towards comprehensive management tailored to prognosis factors of patients with clinical stages I and II in Hodgkin's disease. *Blood*, 73:47–56.

[Uno *et al.*, 2003] Uno, T., Asai, T., Uchida, Y. et Arimura, H. (2003). Lcm : An efficient algorithm for enumerating frequent closed item sets. *In Workshop of Frequent Item Set Mining Implementations co-located with ICDM (FIMI'03), Melbourne, USA*.

[Uno et Satoh, 2003] Uno, T. et Satoh, K. (2003). Detailed description of an algorithm for enumeration of maximal frequent sets with irredundant dualization. *In Workshop of Frequent Item Set Mining Implementations co-located with ICDM (FIMI'03), Melbourne, USA*.

[Velculescu *et al.*, 1995] Velculescu, V., Zhang, L., Vogelstein, B. et Kinzler, K. (1995). Serial analysis of gene expression. *Science*, 270:484–487.

[Wang *et al.*, 2003] Wang, J., Han, J. et Pei, J. (2003). Closet+ : Searching for the best strategies for mining frequent closed itemsets. *In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, USA*.

[Wille, 1982] Wille, R. (1982). *Ordered sets*, chapitre Restructuring lattice theory : an approach based on hierarchies of concepts, pages 445–470. I. Rival (ed.), Reidel, Dordrecht-Boston.

[Zaki et Hsiao, 2002] Zaki, M. et Hsiao, C.-J. (2002). Charm : An efficient algorithm for closed itemset mining. *In Second SIAM International Conference on Data Mining (SDM'02), Arlington, USA*.

[Zhang *et al.*, 1997] Zhang, L., Zhou, W., Velculescu, V., Kern, S., Hruban, R., Hamilton, S., Vogelstein, B. et Kinzler, K. (1997). Gene expression profiles in normal and cancer cells. *Science*, 276:1268–1272.

[Zheng et Toh Low, 1999] Zheng, Z. et Toh Low, B. (1999). Classifying unseen cases with many missing values. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PKDD'99), Beijing, China*, pages 370–374.

[Zighed *et al.*, 1999] Zighed, D., Rabaseda, S., Rakotomalala, R. et Fescheta, F. (1999). Discretization methods in supervised learning. *Encyclopedia of Computer Science and Technology*, 40:35–50.

[Zucker, 2001] Zucker, J. (2001). Changement de représentation, abstractions et apprentissage. Habilitation à diriger les recherches, Université Paris VI, LIP6 - Pôle IA, France.

Résumé L'extraction de motifs est une tâche centrale pour l'extraction de connaissances dans les bases de données. Cette thèse traite de deux cas génériques particulièrement courants dans de nombreuses applications : les bases de données entachées de valeurs manquantes ou comportant un grand nombre d'attributs. Premièrement, nous proposons un mécanisme de désactivation temporaire des objets incomplets, qui permet par des calculs dans une base incomplète de mettre en évidence des propriétés compatibles avec la base complète. Ces propriétés sont exploitées avec celles de la base opposée pour proposer une méthode originale de construction de règles d'association informatives généralisées. Deuxièmement, en utilisant un principe de transposition et les propriétés de la connexion de Galois, nous avons développé un cadre formel pour l'extraction de motifs contraints dans ces données, qui permet de choisir l'orientation de la base de données la plus favorable pour les algorithmes d'extraction. Les contraintes de recherche sont également transposables et permettent d'obtenir les motifs contraints en menant les extractions dans le contexte transposé. Enfin, l'utilisation de règles d'association généralisées à des fins d'apprentissage supervisé et de motifs émergents forts complète ces travaux pour des applications médicales et génomiques.
Mots clés : Exploration de données, Bioinformatique, Correspondances de Galois.

Title

Knowledge discovery in databases with missing values or with a large number of attributes.

Abstract

Knowledge Discovery in Databases is a recent field aiming at discovering new knowledge. Pattern mining is here a central task and this thesis tackles two generic cases : databases containing missing values or a large number of attributes. Firstly, we propose a temporary desactivation process of the incomplete objects, which allows to lead computations in an incomplete database and gives rise to properties compatible with the complete database. An original method for building informative and generalised association rules combines the properties of the opposite database. Secondly, we have developed a complete theoretical framework for the constrained mining of patterns using a transposition principle and the Galois connection properties. It enables to choose the most favourable orientation of the database. Search constraints are also transposable, and allow to get the constrained patterns by leading extractions in the transposed context. At the end, the use of generalised association rules for supervised learning and strong emerging patterns complete these works in both medical and genomic area.
Keywords : Data mining, Databases, Bioinformatics, Galois correspondances.

Discipline : Informatique

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