Extraction optimisée de règles d'association positives et négatives intéressantes

Pierre-Antoine Papon

N ° d'ordre : D.U. : 2702
EDSPIC : 761

UNIVERSITÉ CLERMONT-FERRAND II

ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR

THÈSE

présentée par

Pierre-Antoine PAPON

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : INFORMATIQUE

Extraction Optimisée de Règles d'Association Positives et Négatives Intéressantes

---

Soutenue publiquement le 09 juin 2016 devant le jury :

Rapporteurs :	
Ali KHENCHAF	Professeur des Universités - ENSTA Bretagne
Israël-César LERMAN	Professeur Émérite des Universités - IRISA/INRIA
Invité :	
Engelbert MEPHU NGUIFO	Professeur des Universités - Université Blaise Pascal
Directeur :	
Alain QUILLIOT	Professeur des Universités - Université Blaise Pascal
Co-encadrante :	
Sylvie GUILLAUME	Maître de Conférences - Université d'Auvergne

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse, Sylvie Guillaume et Alain Quilliot, pour m'avoir proposé ce sujet et m'avoir accompagné lors de toutes les étapes de ce travail.

Je souhaite également remercier chaleureusement mes rapporteurs de thèse, Ali Khenchaf et Israël-César Lerman qui ont dû passer de longs moments à me lire et à me comprendre, mais également Engelbert Mephu Nguifo qui m'a fait découvrir la fouille de données et qui a accepté d'évaluer mon travail en tant qu'invité.

Mes remerciements vont par ailleurs à tout le personnel du Limos et des départements informatique, et réseaux et télécommunications que j'ai côtoyé pendant ces dernières années. Merci de m'avoir accueilli parmi vous.

Merci également à tous les thésards que j'ai pu rencontrés : Arnaud[2], Baraa, Ben, Benjamin[4], Benoit[2], Cang, Chao, Charifa, Damien, Dhouha, Diyé, Djelloul, Firmin, Henri, Hicham, Ibrahim, Jean-Christophe, John, Jonathan[2], Kahina, Khaled, Laétitia, Lakhdar, Libo, Marc, Marie, Maxime, Nardjes, Nathalie, Nicolas, Pierre, Rabie, Rafael, Raksmey, Romain, Sabeur, Salsabil, Samuel, Sébastien, Thérèse, Vanel, Wajdi, Xavier, Yahia, Yamen. Oui je sais, certaines de ces personnes n'ont jamais été des doctorants, à vous de trouver ces 3 intrus.

Pour terminer et non par la moindre, je remercie ma chère et tendre Flore pour l'amour qu'elle me porte depuis toutes ces années.

Résumé

L'objectif de la fouille de données consiste à extraire des connaissances à partir de grandes masses de données. Les connaissances extraites peuvent prendre différentes formes. Dans ce travail, nous allons chercher à extraire des connaissances uniquement sous la forme de règles d'association positives et de règles d'association négatives. Une règle d'association négative est une règle dans laquelle la présence ainsi que l'absence d'une variable peuvent être utilisées. En considérant l'absence des variables dans l'étude, nous allons élargir la sémantique des connaissances et extraire des informations non détectables par les méthodes d'extraction de règles d'association positives. Cela va par exemple permettre aux médecins de trouver des caractéristiques qui empêchent une maladie de se déclarer, en plus de chercher des caractéristiques déclenchant une maladie. Cependant, l'ajout de la négation va entraîner différents défis. En effet, comme l'absence d'une variable est en général plus importante que la présence de ces mêmes variables, les coûts de calculs vont augmenter exponentiellement et le risque d'extraire un nombre prohibitif de règles, qui sont pour la plupart redondantes et inintéressantes, va également augmenter.

Afin de remédier à ces problèmes, notre proposition, dérivée de l'algorithme de référence `Apriori`, ne va pas se baser sur les motifs fréquents comme le font les autres méthodes. Nous définissons donc un nouveau type de motifs : les motifs raisonnablement fréquents qui vont permettre d'améliorer la qualité des règles. Nous nous appuyons également sur la mesure M_G pour connaître les types de règles à extraire mais également pour supprimer des règles inintéressantes. Nous utilisons également des méta-règles nous permettant d'inférer l'intérêt d'une règle négative à partir d'une règle positive. Par ailleurs, notre algorithme va extraire un nouveau type de règles négatives qui nous semble intéressant : les règles dont la prémisse et la conclusion sont des conjonctions de motifs négatifs.

Notre étude se termine par une comparaison quantitative et qualitative aux autres algorithmes d'extraction de règles d'association positives et négatives sur différentes bases de données de la littérature. Notre logiciel `ARA` (*Association Rules Analyzer*) facilite l'analyse qualitative des algorithmes en permettant de comparer intuitivement les algorithmes et d'appliquer en post-traitement différentes mesures de qualité. Finalement, notre proposition améliore l'extraction au niveau du nombre et de la qualité des règles extraites mais également au niveau du parcours de recherche des règles.

Mots-clés : fouille de données, règles d'association positives et négatives, motifs raisonnablement fréquents, mesure M_G, méta-règle, règles à conjonctions de motifs négatifs, `ARA`, qualité des règles.

Abstract

The purpose of data mining is to extract knowledge from large amount of data. The extracted knowledge can take different forms. In this work, we will seek to extract knowledge only in the form of positive association rules and negative association rules. A negative association rule is a rule in which the presence and the absence of a variable can be used. When considering the absence of variables in the study, we will expand the semantics of knowledge and extract undetectable information by the positive association rules mining methods. This will, for example allow doctors to find characteristics that prevent disease instead of searching characteristics that cause a disease. Nevertheless, adding the negation will cause various challenges. Indeed, as the absence of a variable is usually more important than the presence of these same variables, the computational costs will increase exponentially and the risk to extract a prohibitive number of rules, which are mostly redundant and uninteresting, will also increase.

In order to address these problems, our proposal, based on the famous `Apriori` algorithm, does not rely on frequent itemsets as other methods do. We define a new type of itemsets : the reasonably frequent itemsets which will improve the quality of the rules. We also rely on the M_G measure to know which forms of rules should be mined but also to remove uninteresting rules. We also use meta-rules to allow us to infer the interest of a negative rule from a positive one. Moreover, our algorithm will extract a new type of negative rules that seems interesting : the rules for which the antecedent and the consequent are conjunctions of negative itemsets.

Our study ends with a quantitative and qualitative comparison with other positive and negative association rules mining algorithms on various databases of the literature. Our software `ARA` (*Association Rules Analyzer*) facilitates the qualitative analysis of the algorithms by allowing to compare intuitively the algorithms and to apply in post-process treatments various quality measures. Finally, our proposal improves the extraction in the number and the quality of the extracted rules but also in the rules search path.

Keywords : data mining, positive and negative association rules, reasonably frequent itemsets, M_G measure, meta-rules, rules with conjunctions of negative itemsets, `ARA`, quality of rules.

Table des matières

Liste des tableaux

Table des figures

Liste des algorithmes

Introduction

Savez-vous que les hommes qui achètent des couches pour bébés quand ils font leurs courses à Walmart les jeudis et samedis après-midis ont tendance à acheter de la bière ? Si vous avez entendu parler de cet exemple [Power, 2002], c'est que quelqu'un a déjà essayé de vous expliquer à quoi sert la fouille de données ou plus globalement l'extraction de connaissances à partir des données. En analysant les tickets de caisse, la chaîne de grande distribution Walmart aurait découvert une corrélation entre l'achat de couches-culottes par des hommes et l'achat de bière les jeudis et samedis après-midis. Walmart aurait donc réorganisé ses rayons en positionnant les packs de bière à côté des couches-culottes, ce qui aurait fait augmenter les ventes des deux produits. Bien que basé sur un fait réel, l'histoire a été enjolivée pour mettre en avant que l'analyse des données pouvait augmenter les profits en tirant parti des outils informatiques d'analyse de données. Même enjolivé, cet exemple illustre parfaitement l'importance de l'extraction de connaissances à partir des données pour les entreprises et comment elle peut leur permettre d'augmenter leurs ventes et d'obtenir un avantage sur leurs concurrents.

Les travaux de recherche effectués dans ce document s'inscrivent dans le domaine de l'extraction de connaissances à partir des données. Les connaissances extraites peuvent prendre différentes formes mais nous allons nous intéresser, dans ce manuscrit, seulement aux connaissances extraites sous la forme de règles d'association. L'extraction de règles d'association consiste à découvrir des relations entre les variables d'une base de données. Dans cette thèse, nous nous intéresserons à l'extraction de règles d'association et plus particulièrement à l'extraction de règles d'association négatives. L'ajout de la négation va permettre d'extraire des règles dans lesquelles la présence ainsi que l'absence d'une variable peuvent être utilisées. L'apport de ces règles négatives n'est pas négligeable puisqu'elles peuvent non seulement contenir des informations non présentes dans les règles positives mais permettent également d'élargir la sémantique des connaissances. Ainsi, en médecine, cela peut permettre de trouver les caractéristiques qui empêchent une maladie de se déclarer, en plus de chercher les caractéristiques déclenchant une maladie. En combinant l'extraction des règles positives avec celle des règles négatives, nous élargissons le champ des connaissances et par conséquent le champ des possibilités. Cependant cet ajout des règles négatives va être un défi en raison essentiellement des coûts de calculs qui vont augmenter exponentiellement, mais également en raison du nombre prohibitif de règles extraites qui sont pour la plupart redondantes et inintéressantes. Ces problèmes proviennent du fait que l'absence de variables est en général plus importante que la présence de ces mêmes variables. Par exemple, dans les bases de données de la grande distribution, chaque consommateur n'achète qu'un sous-ensemble des milliers d'articles

recensés dans le magasin. L'objectif de cette thèse est donc d'extraire de manière optimale l'ensemble des règles d'association positives et négatives intéressantes.

Dans le but de répondre à cette problématique, nous avons structuré ce manuscrit en six chapitres.

Le premier chapitre commence par présenter succinctement le domaine de l'extraction de connaissances à partir des données avant d'introduire le vocabulaire propre aux règles d'association et aux règles d'association négatives. Nous expliquons ensuite le problème d'extraction de règles d'association positives et négatives et justifions l'intérêt de prendre en compte ces règles négatives. L'algorithme `Apriori`, qui est l'algorithme de référence pour extraire les règles d'association positives, est également présenté et analysé en détail. Et enfin, nous présentons les différentes méthodes générant des règles négatives.

Le deuxième chapitre détaille les trois algorithmes majeurs d'extraction de règles d'association positives et négatives évoqués dans le premier chapitre. Ces trois algorithmes, possédant chacun ses avantages et ses inconvénients, ont la particularité d'être des dérivés d'`Apriori` et utilisent donc les mesures du support et de la confiance. Cependant chaque algorithme définit de manière différente ce qu'il considère comme une règle valide. Par conséquent les règles extraites seront différentes d'un algorithme à l'autre comme nous pourrons le voir sur un exemple fil-rouge. L'étude approfondie de ces trois méthodes met en avant essentiellement deux failles, à savoir un nombre encore important de règles inintéressantes et un parcours non optimisé de recherche des règles.

Le troisième chapitre regroupe les différentes propositions que nous faisons pour combler les deux failles mises en avant dans le précédent chapitre. Pour diminuer le nombre de règles inintéressantes, nous n'allons pas utiliser les motifs fréquents comme le font les autres méthodes, mais nous allons définir un nouveau type de motifs : les motifs raisonnablement fréquents. Nous utilisons ensuite la mesure M_G qui permet d'élaguer d'autres règles inintéressantes. Une autre contrainte est également ajoutée afin de renforcer notre souhait d'extraire les règles les plus pertinentes possibles. Cette dernière contrainte nous permet également d'extraire un nouveau type de règles négatives. Afin d'optimiser le parcours de recherche des règles, nous montrons que seule la moitié des règles doivent être étudiées. Nous utilisons également la propriété de la confiance abandonnée par certaines des méthodes présentées dans le deuxième chapitre. Et enfin, nous utilisons des méta-règles nous permettant d'inférer l'intérêt d'une règle négative à partir de la règle positive.

Le quatrième chapitre présente notre méthode pour extraire les règles d'association positives et négatives. Cette méthode repose sur les différentes améliorations que nous avons évoquées dans le troisième chapitre. Nous commençons par présenter les différentes contraintes que doivent respecter les règles pour être valides. Nous présentons ensuite les trois algorithmes utilisés dans notre méthode. Le premier algorithme permet de rechercher les motifs raisonnablement fréquents. Le deuxième va être utilisé pour rechercher les motifs minimaux raisonnablement fréquents. Et le dernier va permettre d'extraire les règles valides à partir des motifs raisonnablement fréquents et va vérifier que les règles négatives sont composées de motifs minimaux raisonnablement fréquents. Ce chapitre se conclut en déroulant notre méthode sur l'exemple fil-rouge.

Le cinquième chapitre contient les expérimentations quantitatives. Dans ce chapitre, nous commençons par présenter le logiciel `Weka` dans lequel nous avons implémenté les différents algorithmes étudiés dans le deuxième chapitre ainsi que notre méthode présentée dans le chapitre précédent. Les expérimentations se décomposent en trois parties. Dans

la première partie, nous mesurons l'impact des différents paramètres de notre méthode en les faisant varier un par un. Dans la deuxième partie, nous mesurons l'impact des améliorations que nous proposons. Et enfin dans la dernière partie, nous confrontons notre méthode à `Apriori` et aux trois autres méthodes d'extraction de règles d'association positives et négatives en les exécutant sur huit différentes bases de données de la littérature.

Le sixième chapitre présente une comparaison approfondie des différents algorithmes effectuée grâce à un logiciel que nous avons développé : `ARA`. Nous commençons donc par présenter ce logiciel nous permettant d'approfondir l'analyse quantitative et d'effectuer une étude qualitative des règles. Nous utilisons ensuite `ARA` sur les règles extraites à partir de l'exemple fil-rouge afin d'analyser les règles communes et spécifiques à chaque méthode. Nous poursuivons en effectuant une analyse qualitative des différentes méthodes. Afin d'effectuer cette analyse, nous devons tout d'abord définir certaines propriétés que doivent respecter les règles. Ces propriétés sont ensuite vérifiées via l'application dans `ARA` en appliquant différentes mesures de qualité sur les règles extraites par les différents algorithmes.

Ce manuscrit s'achève par une conclusion et des perspectives de recherche.

Chapitre 1

Extraction de règles

Sommaire

1.1 Introduction

Dans ce chapitre, nous commençons par définir le processus d'extraction de connaissances à partir des données. Notre objectif étant l'extraction de règles d'association positives et négatives, nous définissons tout d'abord le problème de l'extraction des règles positives. Nous présentons ensuite un algorithme naïf ainsi que l'algorithme de référence pour la résolution de ce problème en détaillant les deux étapes du processus. Nous nous focalisons par la suite sur le problème d'extraction des règles négatives et présentons un état de l'art sur les différentes méthodes existantes pour les différents types de règles négatives.

1.2 Extraction de connaissances à partir des données

Définition 1 - Extraction de connaissances à partir des données :
L'ECD désigne le processus non trivial d'extraction d'informations implicites, précédemment inconnues et potentiellement utiles à partir des données [Frawley et al., 1991].

Cette discipline, se situant à l'intersection de l'informatique et des statistiques, a vu le jour dans les années 1990 afin de répondre aux nouvelles problématiques d'analyse des données. Avec la croissance exponentielle de collecte et de stockage des données, les utilisateurs ont besoin de nouveaux outils afin d'extraire des informations utiles. Ces outils vont permettre à des utilisateurs métiers (*ou experts du domaine d'application*), ne possédant pas forcément de connaissances en informatique et/ou en statistiques, d'analyser leurs données et d'en extraire des connaissances auparavant inconnues et pouvant leur être utiles dans le présent (*comprendre*) ou dans le futur (*prédire*). Mais afin de mieux anticiper ce futur, il faut comprendre le passé puisque comme l'a dit Friedrich Nietzsche « *le futur appartient à celui qui a la plus longue mémoire* ».

L'ECD est un processus itératif comprenant plusieurs étapes qui s'étend de la **spécification du problème** jusqu'à **l'interprétation et l'évaluation des résultats**. La figure 1.1 est reprise très largement de la thèse de Frédéric Pennerath [Pennerath, 2009] et schématise le processus de l'ECD.

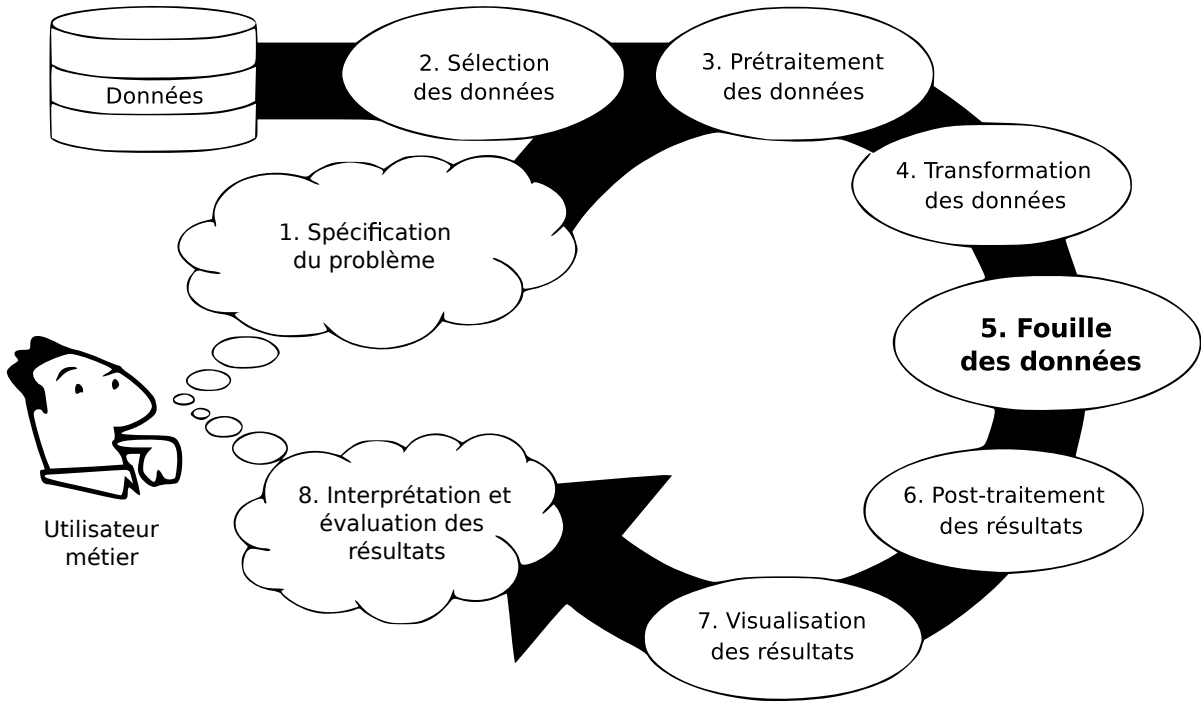

FIGURE 1.1 – Les différentes étapes de l'ECD

Comme c'est indiqué dans la figure 1.1, l'utilisateur métier va procéder en huit étapes afin de trouver une solution à son problème :

1. La première étape est la **spécification du problème** qui consiste à définir clairement la question ouverte visée par l'ECD.

2. La deuxième étape du processus est la **sélection des données**. Elle consiste à conserver uniquement les données qui vont permettre de répondre à la question. Il faut donc effectuer un tri afin de ne garder que le sous-ensemble de données réellement utile à la résolution du problème.

3. La troisième étape, correspondant au **prétraitement des données**, est une phase de nettoyage qui a pour objectif d'améliorer la qualité des données. Plusieurs méthodes vont être appliquées afin de supprimer le bruit, de compléter les données manquantes, de détecter et corriger les données incohérentes ou encore de gérer la présence de redondances.

4. La quatrième étape est la **transformation des données** puisque dans certains cas il faut formater les données pour pouvoir appliquer l'algorithme d'extraction de connaissances. En effet, pour s'exécuter, un algorithme nécessite un certain type de fichier (*fichier csv, fichier arff, base de données MySQL ...*), mais également un certain type de données (*binaires, continues, discrètes ...*). Par exemple dans le logiciel `Weka` [Hall et al., 2009], l'implémentation de l'algorithme d'extraction de règles d'association `FP-Growth` [Han et al., 2000] nécessite des données binaires pour s'exécuter tandis que celle de l'algorithme `Apriori` [Agrawal and Srikant, 1994] peut également travailler sur des données discrétisées.

5. La cinquième étape correspond à la **fouille des données** qui est l'étape centrale de l'ECD. Elle permet d'extraire des connaissances inconnues et utiles à partir des données en appliquant des algorithmes automatiques ou semi-automatiques.

6. Le **post-traitement** est la sixième étape et permet d'améliorer la qualité des résultats générés à l'étape 5. En effet, et notamment dans le cas des règles d'association, les résultats sont parfois retournés en quantité trop importante et rendent difficile l'analyse et l'identification de ceux qui sont réellement intéressants. Afin de résoudre ce problème, un post-traitement utilisant différentes mesures de qualité va aider l'expert à mieux évaluer les résultats et à supprimer la redondance présente dans ceux-ci.

7. La septième étape est la **visualisation des résultats**. Cette étape permet de représenter sous forme visuelle des résultats difficilement intelligibles afin de faciliter la tâche de l'utilisateur dans leur compréhension.

8. Et enfin la huitième étape est l'**interprétation et l'évaluation des résultats**. Cette dernière étape va consister à expliquer les résultats obtenus avec les données de départ et vérifier si une réponse a pu être apportée à la question ouverte. Si les conclusions de cette étape ne sont pas en accord avec ce qui est attendu, le processus peut revenir à n'importe quelle étape afin de continuer et d'affiner l'analyse en cours.

Dans le cadre de cette thèse, nous allons nous focaliser sur la cinquième étape du processus de l'ECD, à savoir la fouille de données. La fouille de données, plus connue sous son anglicisme data mining, consiste à extraire des connaissances inconnues et utiles à partir des données. Les connaissances extraites peuvent prendre différentes formes mais nous allons nous intéresser, dans ce manuscrit, seulement aux connaissances extraites sous la forme de règles d'association.

1.3 Extraction de règles d'association

L'extraction de règles d'association est l'une des techniques les plus populaires de la fouille de données. Ce problème, surnommé analyse du panier de la ménagère, a été introduit pour la première fois en 1993 [Agrawal et al., 1993] pour analyser des bases de données de la grande distribution. Depuis lors, ce problème a été intensément étudié pour son utilité dans de nombreux domaines d'application tels que les systèmes de recommandations, la bio-informatique ou encore les diagnostics médicaux.

L'extraction de règles d'association a pour objectif de découvrir des relations entre les variables de grandes bases de données. Une règle d'association dans la grande distribution pourrait être $Café,\ Thé \Rightarrow Sucre$ qui signifie que si le consommateur achète du café et du thé alors il y a de grandes chances qu'il achète également du sucre.

Après avoir posé le problème, nous présentons l'approche naïve ainsi que les problèmes qu'elle engendre. Nous étudions ensuite l'algorithme de référence pour l'extraction de règles d'association.

1.3.1 Définition du problème

Avant de définir le problème, nous devons expliquer le vocabulaire utilisé.

Définition 2 - Item, motif et *k*-motif :
Soit $\mathcal{I} = \{i_1, i_2, .., i_p\}$ un ensemble de p items i, où chaque item est une variable binaire de la base de données. Un ensemble d'items est appelé un motif, et plus spécifiquement on dit que X est un k-motif s'il est composé de k items : $X = \{i_1, i_2, .., i_k\}$.

Par convention, on utilise les minuscules pour représenter les items et les majuscules pour représenter les motifs. Par conséquent, x est un item et X est un motif (*qui peut être aussi un item*).

Définition 3 - Transaction :
Soit $\mathcal{D} = \{t_1, t_2, .., t_n\}$ un ensemble de n transactions t, où chaque transaction t est un ensemble d'items tel que $t \subseteq \mathcal{I}$. Une transaction t de $\mathcal{D}$ contient X, un ensemble d'items de $\mathcal{I}$, si $X \subseteq t$.

Définition 4 - Règle d'association :
Une règle d'association est une implication de la forme $X \Rightarrow Y$, où $X \subseteq \mathcal{I}$, $Y \subseteq \mathcal{I}$, et $X \cap Y = \varnothing$.

Une règle $X \Rightarrow Y$ indique que les transactions possédant le motif X ont tendance à posséder le motif Y. Cependant, il n'existe aucune relation de causalité entre X et Y : la présence de X ne cause pas la présence de Y.

Définition 5 - Prémisse, antécédent, conclusion et conséquent :
La partie gauche de la règle est appelée la prémisse ou l'antécédent et la partie droite est la conclusion ou le conséquent.

Pour une règle $X \Rightarrow Y$, X est donc la prémisse ou l'antécédent et Y est donc la conclusion ou le conséquent.

Définition 6 - Support, support relatif, support absolu, support minimum min_{sup} et motif fréquent :
Le support représente la fréquence de la règle ou la portée de la règle. Par extrapolation, on utilise la probabilité que la règle soit présente que l'on nomme support relatif par rapport au support absolu qui correspond aux nombres d'occurrences. La règle $X \Rightarrow Y$ a donc un support s si $s\%$ des transactions de $\mathcal{D}$ contiennent $X \cup Y$ également noté par XY simplification d'écriture. Autrement dit, le support correspond à la probabilité que la prémisse X et la conclusion Y soient vraies. Par $\mathcal{D}_{X \cup Y}$, nous indiquons l'ensemble de toutes les transactions qui contiennent $X \cup Y$, $\mathcal{D}_{X \cup Y} = \{t \in \mathcal{D} \mid X \cup Y \subseteq t\}$. Le support s de $X \Rightarrow Y$ est calculé comme $s = sup(X \Rightarrow Y) = sup(X \cup Y) = sup(XY) = P(XY) = \frac{|\mathcal{D}_{X \cup Y}|}{n}$. Un motif X qui respecte le support minimum (i.e. $sup(X) \geqslant min_{sup}$) est dit fréquent.

Définition 7 - Confiance et confiance minimum min_{conf} :
La confiance représente la force de la règle. La règle $X \Rightarrow Y$ a une confiance c si $c\%$ des transactions de $\mathcal{D}$ qui contiennent X contiennent également Y. Autrement dit, la confiance est la probabilité conditionnelle que la conclusion Y soit vraie sachant que la prémisse X est vraie : c'est-à-dire $P(Y|X)$. La confiance c de $X \Rightarrow Y$ est calculée comme $c = conf(X \Rightarrow Y) = \frac{sup(XY)}{sup(X)}$. Il existe également un seuil pour la confiance minimum min_{conf} afin de permettre aux utilisateurs de ne sélectionner que les règles plus pertinentes (i.e. $conf(X \Rightarrow Y) \geqslant min_{conf}$).

Définition 8 - Règle d'association valide :
Une règle d'association est dite valide si ses valeurs pour le support et pour la confiance sont supérieures aux seuils minimaux fixés par l'utilisateur : min_{sup} et min_{conf}.

Ainsi, si la règle $Café \Rightarrow Sucre$ a un support de 0,01 et une confiance de 0,90, alors cela signifie que 1% des consommateurs ont acheté du café et du sucre en même temps, et que parmi ceux qui ont acheté du café, 90% d'entre eux ont également acheté du sucre.

Le problème d'extraction de règles d'association à partir de $\mathcal{D}$ consiste à générer toutes les règles d'association valides. Cette approche est connue sous le nom d'approche support/confiance. Le tableau 1.1 récapitule les contraintes qu'une règle $X \Rightarrow Y$ doit respecter afin d'être considérée comme valide.

$X \Rightarrow Y$
$sup(X \Rightarrow Y) \geq min_{sup}$
$conf(X \Rightarrow Y) \geq min_{conf}$

Tableau 1.1 – Règles valides

La découverte de ces règles va avoir différents objectifs en fonction des données analysées. Dans l'exemple de la grande distribution donné précédemment, cela permet d'étudier le comportement des clients, et de mettre en place une meilleure stratégie marketing. En médecine, cela permet par exemple de détecter les patients à risque pour une maladie donnée. C'est pourquoi la recherche d'algorithmes efficaces de telles règles a été un problème majeur de cette communauté. Nous allons maintenant expliquer un algorithme naïf qui peut être employé pour l'extraction des règles d'association positives.

1.3.2 Algorithme naïf

Le problème de génération des règles d'association peut être décomposé en deux sous-problèmes :

1. Générer tous les motifs fréquents.
2. Générer toutes les règles d'association valides à partir des motifs fréquents.

Une approche naïve serait d'énumérer l'ensemble des motifs et de vérifier un à un le support de chaque motif, puis de générer l'ensemble des règles pour chaque motif fréquent et de vérifier une à une la confiance de chaque règle. Sur la figure 1.2, nous pouvons voir le treillis de l'ensemble des motifs que nous aurions à étudier pour une base comportant quatre items : A, B, C et D. Dans ce cas là, nous aurions donc 15 motifs à étudier puisqu'il faut vérifier le support de tous les motifs possédant au moins un item.

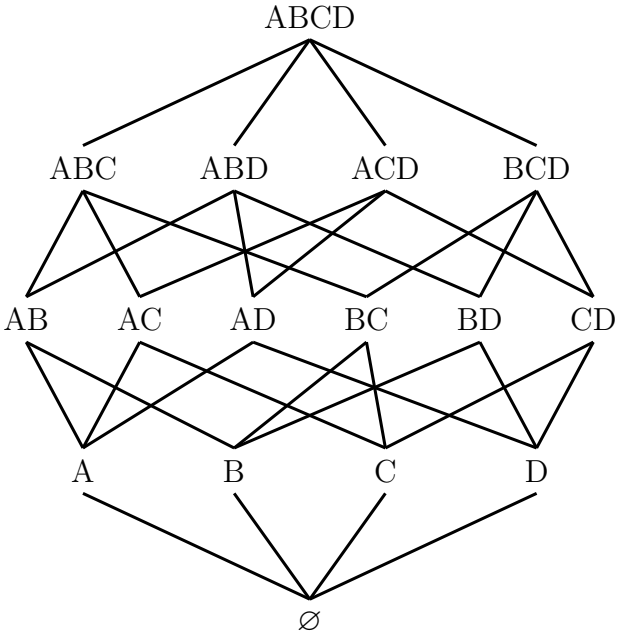

FIGURE 1.2 – Treillis des motifs

Cette méthode est cependant impossible à appliquer à cause des problèmes d'explosion combinatoire qu'elle engendre. En effet, pour une base de données comportant p items, le nombre maximal de motifs possibles est $\sum_{k=1}^{p} \binom{p}{k}$ ou plus simplement $2^p - 1$. Pour une base de données comportant 100 items, cela correspond à environ $1,27 \times 10^{30}$ motifs. L'identification des motifs fréquents est donc coûteuse en calcul et l'espace de recherche grandit exponentiellement en même temps que p. Le coût de calcul sera donc très important lorsque l'on s'intéressera aux bases de données comportant des milliers d'items, comme celles de la grande distribution.

Quant à la phase de génération des règles, le nombre de règles potentielles à analyser dépend du nombre de motifs fréquents mais également de leur taille. Ainsi pour chaque k-motif fréquent de taille supérieure ou égale à 2, il existe $2^k - 2$ règles potentielles. Dans le pire des cas, c'est-à-dire si tous les motifs sont fréquents, le nombre de règles à étudier représente : $\sum_{k=2}^{p} \binom{p}{k} \times (2^k - 2)$. En reprenant la base de l'exemple précédent comportant 100 items, le nombre de règles potentielles correspond à environ $5,15 \times 10^{47}$.

Même si la phase de génération des règles est importante, le principal goulot d'étranglement d'une telle approche provient de la génération des motifs candidats. Pas-

sons maintenant à l'algorithme de référence pour l'extraction de règles d'association positives qui comble en partie ces deux problèmes.

1.3.3 Algorithme Apriori

Comme la méthode naïve, l'algorithme `Apriori` [Agrawal and Srikant, 1994], va se décomposer en deux phases. Des optimisations vont être apportées dans les deux parties de l'algorithme, à savoir dans la recherche des motifs fréquents et dans la recherche des règles. Commençons par analyser ces optimisations et pour cela étudions les méthodes utilisées dans les deux étapes du processus. Nous déroulerons ensuite `Apriori` sur un exemple.

➤ Génération des motifs fréquents

L'algorithme `Apriori` utilise une propriété du support qui va permettre de ne pas parcourir tout l'espace de recherche et par conséquent va accélérer le processus d'extraction des motifs fréquents. Le support est une mesure anti-monotone.

Définition 9 - Mesure anti-monotone :
Une mesure $\mathcal{M}$ est dite anti-monotone si et seulement si : $\forall X, Y \subseteq \mathcal{I}$, *si* $X \subsetneq Y$ *et* $\mathcal{M}(Y)$ *alors* $\mathcal{M}(X)$.

Autrement dit, une mesure est anti-monotone si lorsqu'elle est vérifiée pour un motif, elle est forcément vérifiée pour un sous-ensemble englobant ce motif. Il existe également des mesures monotones comme nous le verrons par la suite. Profitons-en pour donner la définition.

Définition 10 - Mesure monotone :
Une mesure $\mathcal{M}$ est dite monotone si et seulement si : $\forall X, Y \subseteq \mathcal{I}$, *si* $X \subsetneq Y$ *et* $\mathcal{M}(X)$ *alors* $\mathcal{M}(Y)$.

Autrement dit, une mesure est monotone si lorsqu'elle est vérifiée pour un motif, elle est forcément vérifiée pour un sur-ensemble englobant ce motif.

Cette propriété définit donc que le support de tout sur-ensemble Y d'un motif X est inférieur ou égal au support de X, c'est-à-dire que $\forall\ Y \supsetneq X$, $sup(Y) \leqslant sup(X)$. Par conséquent, tous les sur-ensembles d'un motif non fréquent sont non fréquents. Par exemple, si C est non fréquent, aucun sur-ensemble de C ne peut être fréquent comme par exemple AC ou BC. Cette propriété va permettre d'élaguer un k-motif lorsqu'au moins un de ses sous-ensembles de taille *(k-1)* n'est pas fréquent. La figure 1.3 représente le treillis de l'ensemble des motifs à étudier si C n'est pas fréquent.

La vérification du support de C nous permet d'élaguer 7 motifs dans l'étude et qui sont les suivants : $\{AC, BC, CD, ABC, ACD, BCD, ABCD\}$. Cette propriété va donc avoir une incidence sur l'ordre dans lequel on génère les motifs. Pour éviter la vérification du support sur un maximum de motifs, il faut donc générer les motifs par ordre croissant de taille.

Nous venons de présenter la propriété du support qui permet d'accélérer le processus d'extraction des motifs fréquents. regardons maintenant comme elle est utilisée par `Apriori`. Précisons tout d'abord le prérequis que nécessite `Apriori` : les motifs doivent être écrits dans l'ordre lexicographique afin d'éviter de considérer plusieurs fois le même

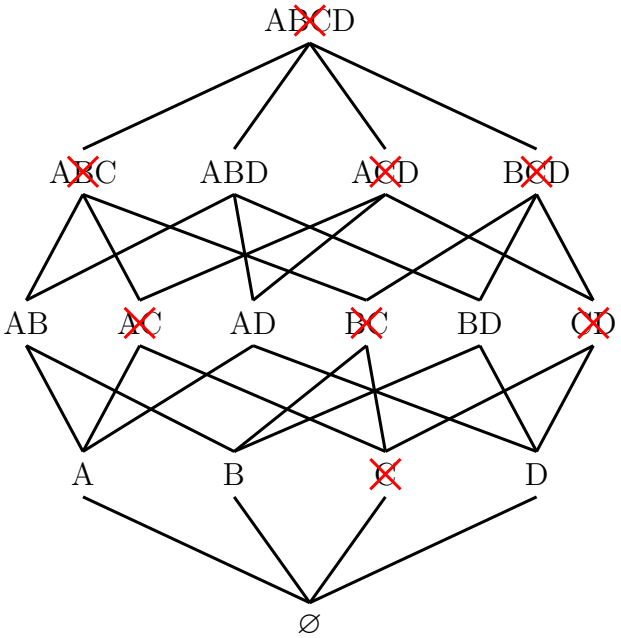

FIGURE 1.3 – Treillis des motifs potentiellement fréquents quand C n'est pas fréquent

motif. AB et BA représentent donc le même motif. Pour générer l'ensemble des motifs fréquents l'*algorithme 1*, nommé *fréquents* va être utilisé.

Algorithme 1 : *fréquents* - Génération des motifs fréquents

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}
Sortie : ensemble des motifs fréquents F

```
1  F = ∅
2  C_1 = {i ∈ I}
3  pour (k = 1; C_k ≠ ∅; k++) faire
4  |  F_k = ∅
5  |  pour tout motif candidat X ∈ C_k faire
6  |  |  s = support(D, X)
7  |  |  si s ⩾ min_sup alors
8  |  |  |  F_k = F_k ∪ {X}
9  |  C_{k+1} = candidats(F_k)
10 |  F = F ∪ F_k
11 retourner F
```

Cet algorithme commence (*ligne 1*) par initialiser l'ensemble C_1 des *1*-motifs candidats à l'ensemble de tous les items i de la base de données $\mathcal{D}$ passée en paramètre. Le processus suivant (*lignes 2 à 9*) va être réitéré jusqu'à ce que l'on n'obtienne plus de candidat ($C_k \neq \varnothing$) à partir de l'ensemble F_{k-1} des motifs fréquents X de niveau inférieur (*ligne 8*). En effet, la génération des candidats effectuée par la fonction *candidats* (*cf. algorithme 2*), repose sur la propriété anti-monotone du support et va donc s'effectuer uniquement à partir des motifs fréquents afin d'éviter de vérifier le support de certains candidats que l'on sait par avance trop faible. Pour un niveau k donné, on commence par initialiser l'ensemble des k-motifs fréquents F_k à l'ensemble vide (*ligne 3*). Ensuite

pour tous les candidats X de C_k (*ligne 4*), on calcule le support avec la fonction *support* (*ligne 5*) qui interroge la base de données. Puis, si le motif X est fréquent (*ligne 6*), on le stocke dans l'ensemble F_k qui servira, une fois l'ensemble des motifs X de C_k parcouru, à générer les candidats de niveau supérieur (*ligne 8*). La dernière étape (*ligne 9*) consiste à ajouter l'ensemble des k-motifs fréquents F_k à l'ensemble des motifs fréquents F. Lorsqu'il n'existe plus de candidats à analyser, on retourne l'ensemble des fréquents F (*ligne 10*).

Après avoir exposé la recherche des motifs fréquents, nous expliquons la fonction *candidats*, correspondant à l'*algorithme 2*, et qui permet de générer le prochain niveau de candidats.

Algorithme 2 : *candidats* - Génération des *(k+1)*-motifs candidats

Entrée : ensemble des k-motifs fréquents F_k
Sortie : ensemble des *(k+1)*-motifs candidats C_{k+1}

```
1 C_{k+1} = F_k ⋈ F_k
2 pour tout motif candidat potentiel X ∈ C_{k+1} faire
3     pour tout k-motif Y ⊂ X faire
4         si Y ∉ F_k alors
5             C_{k+1} = C_{k+1} \ X
6 retourner C_{k+1}
```

Cette fonction va prendre en paramètre l'ensemble des k-motifs fréquents F_k et va retourner les candidats à analyser à la prochaine itération. L'algorithme commence par générer l'ensemble C_{k+1} des candidats potentiels (*ligne 1*) en combinant l'ensemble F_k avec lui même. Deux k-motifs peuvent être combinés pour créer un nouvel *(k+1)*-motif candidat si et seulement s'ils ont en commun les *(k-1)* premiers items. Par exemple, AB et AC ont leur premier item A en commun et peuvent donc se combiner pour créer le motif ABC. Cependant, les motifs AC et BC ne peuvent pas se combiner car même si C est commun, il n'est pas le premier item. La seconde étape (*lignes 2 à 5*) est l'élagage des candidats dont tous les sous-ensembles ne sont pas fréquents. Pour se faire, il faut vérifier pour chaque *(k+1)*-motif X du nouvel ensemble C_{k+1} (*ligne 2*), que chaque sous-motif Y de taille k (*ligne 3*) composant X est bien fréquent. Si le sous-motif Y n'est pas fréquent (*ligne 4*) alors le motif X analysé doit être retiré de l'ensemble C_{k+1} (*ligne 5*). Une fois que tous les motifs X sont analysés, on retourne l'ensemble des *(k+1)*-motifs candidats C_{k+1} (*ligne 6*).

La première étape du processus étant finie, passons à la seconde étape qui consiste à générer les règles à partir des motifs fréquents.

➤ Génération des règles valides

Pour la génération des règles, on utilise une propriété de la confiance pour élaguer certaines règles sans avoir à calculer leur confiance.

Propriété 1 - Propriété de la confiance :
Cette propriété est définie comme suit :
$\forall (X, Y, Z)$ *tel que* $Z \subsetneq Y \subsetneq X$, $conf(Z \Rightarrow X \backslash Z) \leqslant conf(Y \Rightarrow X \backslash Y)$.

Preuve :

$$\begin{array}{lrcl} & conf(Z \Rightarrow X\backslash Z) & \leqslant & conf(Y \Rightarrow X\backslash Y) \\ \Leftrightarrow & \frac{sup(Z \Rightarrow X\backslash Z)}{sup(Z)} & \leqslant & \frac{sup(Y \Rightarrow X\backslash Y)}{sup(Y)} \\ \Leftrightarrow & \frac{sup(X)}{sup(Z)} & \leqslant & \frac{sup(X)}{sup(Y)}, \text{ puisque } Z \subsetneq Y. \end{array}$$

Cette propriété de la confiance nous permet donc de déduire deux choses :

1. si la confiance de la règle $Z \Rightarrow X\backslash Z$ n'est pas valide alors la confiance de la règle $Y \Rightarrow X\backslash Y$ ne le sera pas non plus.
2. la proposition contraposée, si la confiance de la règle $Z \Rightarrow X\backslash Z$ est valide alors la confiance de la règle $Y \Rightarrow X\backslash Y$ le sera également.

Exemple : si A a un support absolu de 2, le support de AB est inférieur ou égal à 2. Par conséquent, si la confiance de la règle $AB \Rightarrow C$ ne respecte pas le seuil minimum de confiance alors on sait par avance que la confiance de la règle $A \Rightarrow BC$ ne le sera pas non plus. En effet, pour calculer la confiance de ces deux règles, il faut diviser le support du motif ABC par le support de la prémisse. Le numérateur est donc commun pour les deux calculs et seul le dénominateur change, or le support de AB apparaissant au dénominateur pour le calcul de la confiance de la règle $AB \Rightarrow C$ ne peut être que plus petit ou égal à celui du support de A apparaissant au dénominateur pour le calcul de la confiance de la règle $A \Rightarrow BC$.

Cette propriété va avoir une incidence sur l'ordre d'étude des différentes règles pour un même motif. Pour éviter la vérification de la confiance sur un maximum de règles, il faut donc commencer par analyser les règles qui possèdent un motif en conclusion le plus petit possible. Regardons maintenant comment ces optimisations ont été utilisées dans l'algorithme `Apriori`.

Pour générer l'ensemble des règles, Agrawal *et al.* vont utiliser l'algorithme *règles* (*cf. algorithme* 3) qui recherche les règles possédant un seul item en conséquence, puis qui fait appel à l'algorithme *autresRègles* (*cf. algorithme* 4) pour générer les autres règles, c'est-à-dire celles possédant plusieurs items en conclusion.

L'algorithme *règles* (*cf. algorithme* 3) prend en paramètre l'ensemble des fréquents F ainsi que le seuil minimum de la confiance min_{conf}. Cet algorithme se déroule en deux phases. La première phase (*lignes 1 à 9*) consiste à générer, à partir des motifs fréquents, les règles possédant un seul item en conclusion. La seconde phase (*ligne 10*) fait appel au second algorithme afin de générer, pour chaque motif, les règles possédant plusieurs items en conclusion. Concernant la première phase, on commence par parcourir tous

Algorithme 3 : *règles* - Génération des règles d'association

Entrées : ensemble des motifs fréquents F, confiance minimum min_{conf}
Sortie : ensemble des règles d'association valides R

```
1  R = ∅
2  pour tout k-motif X ∈ F tel que k > 1 faire
3      E1 = ensemble des 1-motifs ⊂ X
4      pour tout Y ∈ E1 faire
5          c = conf(X\Y ⇒ Y)
6          si c ⩾ min_conf alors
7              R = R ∪ {X\Y ⇒ Y}
8          sinon
9              E1 = E1\Y
10     R = R ∪ autresRègles(X, E1, min_conf)
11 retourner R
```

les motifs fréquents de taille strictement supérieure à 1 (*ligne 2*) puisque l'on ne peut pas générer de règles comportant un seul item. On récupère dans l'ensemble E_1 tous les items qui composent le motif X en cours d'étude (*ligne 3*). Ensuite, pour chaque item Y de l'ensemble E_1 (*ligne 4*), on calcule la confiance de la règle $X \backslash Y \Rightarrow Y$ (*ligne 5*). Si la confiance de la règle est supérieure ou égale au seuil min_{conf} (*ligne 6*), alors la règle est ajoutée à l'ensemble des règles valides R (*ligne 7*). Si la règle n'est pas valide (*ligne 8*), le motif Y va être retiré de l'ensemble E_1 (*ligne 9*). Une fois tous les items Y parcourus, l'ensemble des règles possibles possédant un seul item en conclusion sera généré pour un motif donné et E_1 contiendra uniquement les conclusions qui ont permis de générer les règles valides. Ce nouvel ensemble E_1 sera utilisé dans la fonction *autresRègles* (*ligne 10*) pour générer les autres règles, c'est-à-dire celles possédant plusieurs items en conclusion. La fonction *autresRègles* repose sur la propriété anti-monotone du support et va nous éviter de calculer la confiance de certaines règles que l'on sait par avance trop faible.

La fonction récursive *autresRègles* (*cf. algorithme* 4) va retourner l'ensemble des règles valides possédant plusieurs items en conclusion pour chaque motif X passé en paramètre. Le second paramètre de la fonction est l'ensemble E_m des m-motifs conclusion Y pour lesquels la règle $X \backslash Y \Rightarrow Y$ est valide. L'algorithme commence par vérifier s'il est possible de générer d'autres règles à partir du motif X (*ligne 2*). En effet, il faut vérifier que la taille k du motif X passé en paramètre est strictement supérieure aux tailles $(m+1)$ des futures conclusions. Puis, on appelle la fonction *candidats* (*ligne 3*) afin de générer les conclusions de taille $(m+1)$ à partir des conclusions de taille m qui ont mené à des règles valides à l'itération précédente. Ce nouvel ensemble de conclusions est stocké dans l'ensemble E_{m+1}. Ensuite, pour chaque item Y de l'ensemble E_{m+1} (*ligne 4*) on calcule la confiance de la règle $X \backslash Y \Rightarrow Y$ (*ligne 5*). Si la confiance de la règle est supérieure ou égale au seuil min_{conf} (*ligne 6*) alors la règle est ajoutée à l'ensemble des règles valides R' (*ligne 7*). Si la règle n'est pas valide (*ligne 8*), le motif Y va être retiré de l'ensemble E_{m+1} (*ligne 9*). Une fois tous les items Y parcourus, l'ensemble des règles possibles possédant $(m+1)$ items en conclusion est généré pour un motif X donné et E_{m+1} contiendra uniquement les

conclusions qui ont permis de générer les règles valides. Ce nouvel ensemble E_{m+1} sera à son tour utilisé dans la fonction récursive *autresRègles* (*ligne 10*) pour générer les règles possédant (*m+2*) items en conclusion.

Algorithme 4 : *autresRègles* - Génération des règles d'association composées de plus d'un item en conclusion

Entrées : k-motif fréquent X, ensemble des m-motifs en conclusion E_m, confiance minimum min_{conf}
Sortie : ensemble des règles d'association valides R' possédant un *(m+1)*-motif en conclusion pour le motif X

```
1  R' = ∅
2  si k > m + 1 alors
3      E_{m+1} = candidats(E_m)
4      pour tout Y ∈ E_{m+1} faire
5          c = conf(X\Y ⇒ Y)
6          si c ⩾ min_conf alors
7              R' = R' ∪ {X\Y ⇒ Y}
8          sinon
9              E_{m+1} = E_{m+1}\Y
10     R' = R' ∪ autresRègles(X, E_{m+1}, min_conf)
11 retourner R'
```

➤ Discussion

`Apriori` est l'algorithme de référence pour l'extraction de règles d'association mais il possède cependant certaines faiblesses.

Une première faiblesse provient de la nécessité de récupérer le support des motifs dans la base de données. En effet, un passage est nécessaire pour chaque taille de motif. Par exemple, si le motif de plus grande taille possède 1000 items alors l'algorithme requiert 1000 balayages complets de la base de données. Ces balayages sont assez coûteux en terme d'entrées/sorties si la base de données ne peut pas être chargée en mémoire.

Une seconde faiblesse provient du dilemme pour choisir le seuil du support. En effet, l'extraction de règles avec un support élevé peut entraîner la perte des pépites de connaissances. Les pépites sont des règles avec un support faible mais possédant une confiance élevée. Ces règles sont intéressantes puisqu'elles apportent en général de l'information inattendue et surprenante à l'utilisateur. Néanmoins, le choix d'un support faible peut entraîner un nombre prohibitif de règles, dont la plupart sont inintéressantes et redondantes.

Depuis le célèbre algorithme `Apriori` [Agrawal and Srikant, 1994], il y a eu de nombreuses variantes et améliorations et notamment les algorithmes `Eclat` [Zaki et al., 1997] et `FP-Growth` [Han et al., 2000] qui comblent en partie ces faiblesses. `Eclat` va stocker pour chaque item la liste des transactions qui le contiennent dans un ensemble. Le support des motifs est ensuite calculé en utilisant les intersections d'ensemble. `FP-Growth` utilise une structure compacte appelée `FP-Tree` qui va permettre d'extraire les motifs fréquents sans générer de candidats. De plus, alors qu'`Apriori` interroge la base pour chaque niveau

de motifs candidats générés, `FP-Growth` nécessite seulement deux passages, ce qui accélère encore les traitements.

➤ Exemple

Déroulons maintenant l'algorithme `Apriori` sur un petit exemple (*cf. tableau* 1.2) afin d'éclaircir son fonctionnement. Cet exemple comporte 5 items : A, B, C, D et E ; et 4 transactions. Les 0 et les 1 représentent respectivement l'absence ou la présence d'un item dans la transaction. La première transaction contient donc les items A, C et D alors que les items B et E sont absents. Nous allons prendre les paramètres suivants : 0,25 pour le support minimum et 0,80 pour la confiance minimum.

A	B	C	D	E
1	0	1	1	0
0	1	1	0	1
1	1	1	0	0
0	1	0	0	1

Tableau 1.2 – Exemple de base de données

La première étape consiste à rechercher les motifs fréquents.

1) Extraction des motifs fréquents

1-Motifs fréquents :

Pour rechercher les motifs fréquents, la première étape consiste à récupérer l'ensemble des motifs de taille 1. Une fois les items A, B, C, D et E récupérés, la fonction *support* est ensuite utilisée afin de calculer les différents supports dans la base de données. Le support de chaque item est ensuite comparé à la valeur du support minimum afin de vérifier si c'est un motif fréquent. Pour un support de 0,25, un motif sera fréquent s'il apparaît au moins une fois puisque $0{,}25 \times 4 = 1$ (*support minimum* $\times$ *nombre de transactions*). Les items A, B, C, D et E sont donc fréquents. Le tableau 1.3 récapitule les 5 items fréquents avec leur support.

Item	Support	Item	Support	Item	Support
A	0,50	B	0,75	C	0,75
D	0,25	E	0,50		

Tableau 1.3 – Items fréquents de taille 1 accompagnés de leur support

2-Motifs candidats :

La prochaine étape consiste à générer les motifs candidats de taille 2 à partir des motifs fréquents de taille 1. Pour se faire il suffit de les combiner. Deux k-motifs fréquents

peuvent être combinés pour créer un nouvel *(k+1)*-motif candidat si et seulement ils ont en commun les *(k-1)* premiers items. Par conséquent A et B peuvent se combiner pour former AB car ils n'ont pas besoin d'avoir d'items en communs. Les motifs candidats de taille 2 sont référencés dans le tableau 1.4.

2-Motif					
AB	AC	AD	AE	BC	BD
BE	CD	CE	DE		

Tableau 1.4 – Motifs candidats de taille 2

2-Motifs fréquents :

Le support est ensuite calculé pour chaque motif candidat puis comparé au support minimum. Le tableau 1.5 restitue les motifs fréquents parmi les 2-motifs candidats du tableau 1.4.

2-Motif	Support	2-Motif	Support	2-Motif	Support
AB	0,25	AC	0,50	AD	0,25
BC	0,50	BE	0,50	CD	0,25
CE	0,25				

Tableau 1.5 – Motifs fréquents de taille 2 accompagnés de leur support

3-Motifs candidats :

Le tableau 1.6 donne les candidats potentiels de taille 3 générés à partir des 2-motifs fréquents du tableau 1.5. Les motifs candidats potentiels sont combinés que s'ils ont leur premier item en commun. AB et AC ont donc été combinés car ils ont A en commun.

3-Motif		
ABC	ABD	ACD
BCE	CDE	

Tableau 1.6 – Motifs candidats potentiels de taille 3

Le tableau 1.7 donne les candidats de taille 3 conservés à partir des 3-motifs candidats potentiels du tableau 1.6. ABC sera ensuite conservé car BC fait également partie des fréquents. En effet il faut vérifier que tous les sous-motifs soient fréquents. Lorsque AB et AD vont être combinés, ABD ne sera pas conservé car BD n'est pas fréquent.

3-Motif		
ABC	ACD	BCE

Tableau 1.7 – Motifs candidats de taille 3

3-Motifs fréquents :

On recherche ensuite les motifs fréquents parmi les motifs candidats. Les motifs fréquents de taille 3 sont renseignés dans le tableau 1.8.

3-Motif	Support	3-Motif	Support	3-Motif	Support
ABC	0,25	ACD	0,25	BCE	0,25

TABLEAU 1.8 – Motifs fréquents de taille 3 accompagnés de leur support

4-Motifs candidats :

L'ensemble des motifs fréquents n'est toujours pas vide, donc nous pouvons continuer la génération des candidats du prochain niveau. Cependant, l'algorithme de génération des motifs fréquents s'arrêtent puisque les motifs fréquents de taille 3 ne sont plus combinables.

2) Génération des règles valides

La prochaine étape va être la génération des règles à partir des motifs fréquents. Prenons à titre d'exemple, le motif fréquent ACD et cherchons les règles valides.

Règles possédant un seul item en conclusion :

La première étape de la recherche des règles va être de rechercher les règles possédant un seul item en conclusion. Le tableau 1.9 expose les règles possédant un seul item en conclusion accompagnées du calcul de la confiance.

Règle	Confiance
$AC \Rightarrow D$	$\frac{sup(ACD)}{sup(AC)} = \frac{0,25}{0,50} = 0,50$
$AD \Rightarrow C$	$\frac{sup(ACD)}{sup(AD)} = \frac{0,25}{0,25} = 1$
$CD \Rightarrow A$	$\frac{sup(ACD)}{sup(CD)} = \frac{0,25}{0,25} = 1$

TABLEAU 1.9 – Règles possédant un seul item en conclusion pour le motif ACD

Après l'étude des règles possédant un seul item en conclusion pour le motif ACD, les règles $AD \Rightarrow C$ et $CD \Rightarrow A$ respectent le seuil de confiance minimum ($min_{conf} = 0,80$) et vont être conservées.

Règles possédant plusieurs items en conclusion :

La seconde étape permet de générer les règles possédant plusieurs items en conclusion. Les conclusions A et C ont permis de générer deux règles valides. Par conséquent, pour le prochain niveau de recherche des règles, les deux conclusions vont être combinées afin de créer une nouvelle conclusion AC. La combinaison des conclusions se déroule comme

pour la combinaison des motifs lors de la recherche des motifs fréquents. La suite de l'étude est présentée dans le tableau 1.10.

Règle	Confiance
$D \Rightarrow AC$	$\frac{sup(ACD)}{sup(D)} = \frac{0,25}{0,25} = 1$

TABLEAU 1.10 – Règle possédant deux items en conclusion pour le motif ACD

La règle $D \Rightarrow AC$ possède une confiance supérieure au seuil minimum de confiance ($min_{conf} = 0,80$) et va par conséquent être conservée. Le même processus est effectué pour tous les autres motifs fréquents. Les résultats obtenus sont résumés dans le tableau 1.11.

Motif fréquent	Règle	Confiance	Motif fréquent	Règle	Confiance
AC	$A \Rightarrow C$	1	AD	$D \Rightarrow A$	1
BE	$E \Rightarrow B$	1	CD	$D \Rightarrow C$	1
ABC	$AB \Rightarrow C$	1	ACD	$AD \Rightarrow C$	1
ACD	$CD \Rightarrow A$	1	ACD	$D \Rightarrow AC$	1
BCE	$CE \Rightarrow B$	1			

TABLEAU 1.11 – Règles extraites sur la base d'exemple

En conclusion, `Apriori` génère 9 règles $X \Rightarrow Y$ que l'on nomme règles positives par opposition aux règles négatives que l'on cherche à extraire lorsque l'on s'intéresse à l'extraction des règles utilisant également l'absence de variables comme motif. Ces règles négatives sont présentées dans la section suivante.

1.4 Extraction de règles d'association négatives

Avant de justifier l'intérêt de l'extraction de ces règles négatives, nous définissons ce qu'est une règle négative. Nous présentons par la suite un algorithme naïf ainsi que les différentes méthodes existantes dans la littérature permettant de les extraire.

1.4.1 Définition du problème

L'extraction de règles d'association négatives permet donc d'extraire des règles dans lesquelles la présence ainsi que l'absence d'un item peuvent être utilisées. Ainsi au lieu de s'intéresser uniquement à la présence des items ou motifs X dans la règle, il faut également analyser l'absence de ces mêmes motifs, que l'on va noter $\overline{X}$. Un exemple de règle négative dans la grande distribution pourrait être que les clients qui achètent de la bière achètent généralement des chips mais pas de vin : $Bière \Rightarrow Chips$, $\overline{Vin}$ ou encore que les clients qui achètent de la bière mais pas de pizza, achètent généralement des chips : $Bière$, $\overline{Pizza} \Rightarrow Chips$.

Définition 11 - Règle d'association négative :
Une règle d'association négative est donc une règle d'association qui possède au moins un item négatif dans la prémisse et/ou dans la conclusion.

Cette prise en compte de l'absence d'un élément dans la règle va être un défi puisque le nombre de combinaisons possibles de motifs composés d'items positifs et/ou d'items négatifs augmente exponentiellement et par conséquent le nombre de règles valides augmente également. De plus, certaines règles extraites risquent d'être inintéressantes, puisqu'elles pourront être redondantes et présentent au sein des règles positives. En effet, dans le cas où une variable X possède seulement deux valeurs possibles $X = x1$ et $X = x2$, la négation $\overline{X = x1}$ est similaire à $X = x2$. Afin de limiter l'explosion combinatoire induite par l'ajout de la négation, la majorité des approches ont tendance à considérer l'ensemble des items composants les prémisses et les conclusions comme un unique motif soit entièrement positif soit entièrement négatif. Par exemple, un motif $\overline{Café, Thé}$ signifie l'absence du motif $Café, Thé$ dans les transactions. Ce motif englobe donc toutes les transactions qui contiennent du thé et pas de café ou du café et pas de thé ainsi que les transactions où le café et le thé sont absents simultanément.

Une règle d'association négative est donc une règle d'association où la prémisse et/ou la conclusion est composée d'un motif entièrement négatif. Autrement dit, en plus de chercher les règles $X \Rightarrow Y$ dites positives, il est nécessaire de regarder également les règles négatives suivantes : $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$. De ce fait, il peut y avoir potentiellement trois fois plus de règles négatives que de positives. Par conséquent, parmi l'ensemble des règles pouvant être extraites les trois quarts sont des règles négatives. Dans le pire des cas le nombre de règles positives et négatives pouvant être extraites est donc de $4 \times \sum_{k=2}^{p} \binom{p}{k} \times (2^k - 2)$ (*on rappelle que p est le nombre d'items et k la taille des motifs*). Un exemple de règle négative dans la grande distribution pourrait être $Café \Rightarrow \overline{Thé}$ afin de mettre en évidence que les consommateurs qui boivent du café n'achètent généralement pas de thé.

1.4.2 Intérêt des règles négatives

[Brin et al., 1997a] sont les premiers à avoir mis en évidence l'importance de l'extraction de ces règles négatives et indiquent que de la connaissance précieuse peut se cacher dans ces règles. Ainsi, en médecine, il peut être intéressant de connaître les caractéristiques qui empêchent une maladie de se déclarer. Une telle règle serait représentée par $X \Rightarrow \overline{Y}$ avec la conclusion Y correspondant à la maladie que les caractéristiques X empêchent de déclarer. Les règles du type $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow Y$ vont également permettre de trouver des substituts. Par exemple, les personnes qui ne boivent pas de vin boivent des sodas $\overline{Vin} \Rightarrow Soda$ et celle qui boivent du vin ne boivent pas de sodas $Vin \Rightarrow \overline{Sodas}$. On peut donc considérer que les sodas et le vin sont mutuellement substituables. De la même façon, les règles $\overline{X} \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ peuvent également être très précieuses en médecine. Par exemple, la règle $\overline{Fer} \Rightarrow Anémie$ indique qu'une carence en fer est une des causes de l'anémie. Les règles $Hepcidine \Rightarrow Anémie$ et $\overline{Hepcidine} \Rightarrow \overline{Anémie}$ sont aussi riches d'enseignement. Les deux règles impliquant l'hormone *hepcidine* dans l'apparition ou la disparition de l'*anémie* montrent leur lien étroit avec cette maladie. Les chercheurs ont ainsi découvert que cette hormone régulait l'absorption du fer. Le fer nécessaire à notre organisme provient de l'alimentation et pénètre au niveau intestinal. Cependant, il existe un phénomène de régulation : lorsque nous manquons de fer, il est absorbé par

la paroi intestinale ; lorsque nous en avons trop, il ne peut plus franchir la barrière intestinale. Or, cette hormone libérée par le foie agit directement sur la paroi intestinale en bloquant l'entrée de fer : en son absence, le fer entre librement dans l'intestin d'où la règle $\overline{Hepcidine} \Rightarrow \overline{Anémie}$; et en sa présence excessive le fer est arrêté d'où, la seconde règle $Hepcidine \Rightarrow Anémie$. Les règles dont la prémisse et la conclusion sont composées de motifs positifs et négatifs peuvent également s'avérer utiles. Prenons par exemple la règle valide suivante $Bière, \overline{Pizza} \Rightarrow Chips$ accompagnée de la règle $Bière \Rightarrow Chips$ non valide à cause de la confiance. Ici, l'ajout de la négation va permettre d'extraire une règle plus spécifique non détectable par les règles positives. En conclusion, si l'extraction se limitait aux règles positives, toutes ces informations ne seraient pas mises en évidence. Les utilisateurs métiers risquent donc de passer à côté de connaissances importantes en limitant leur analyse aux règles positives, puisque une absence de variable n'est pas tout le temps une absence de connaissance. En effet, il se peut que l'absence de la variable révèle un comportement comme nous avons pu le voir avec l'absence des pizzas lors de l'achat de bières et de chips.

1.4.3 Algorithme naïf

Afin d'extraire ces nouvelles règles, une approche naïve consiste à traiter l'absence d'un item comme un nouvel item puis à lancer un algorithme classique d'extraction de règles d'association positives. Cependant la complexité de l'extraction des règles d'association dépend essentiellement du nombre de motifs et est exponentielle. Par conséquent, cette approche naïve serait dramatique en raison essentiellement des coûts de calculs, mais également en raison du nombre prohibitif de règles redondantes et inintéressantes extraites. Un exemple d'une telle approche se trouve dans [Kouris et al., 2007] qui considèrent les items négatifs comme d'autres items et va utiliser une variante de l'algorithme d'extraction de règles positives `MSApriori` [Liu et al., 1999a] en utilisant deux seuils distincts pour le support des motifs positifs et le support des motifs négatifs.

Cependant, il existe une relation entre les supports des motifs positifs et négatifs. Cette relation nous évite d'effectuer un nouveau balayage de la base de données pour obtenir le support des motifs négatifs puisqu'ils peuvent être déduits à partir des supports des motifs positifs.

Propriété 2 - Relation entre les supports des motifs positifs et négatifs :
Les supports de $\overline{X}$, de $X\overline{Y}$ et de $\overline{X}\,\overline{Y}$ peuvent être obtenus avec les formules suivantes :

- $sup(\overline{X}) = 1 - sup(X)$,
- $sup(X\overline{Y}) = sup(X) - sup(XY)$,
- $sup(\overline{X}\,\overline{Y}) = 1 - sup(X) - sup(Y) + sup(XY)$.

Passons maintenant aux principales méthodes existantes dans la littérature pour extraire les règles négatives.

1.4.4 Extraction à l'aide d'une taxonomie

Certaines techniques essayent de restreindre l'espace de recherche des règles. Pour réduire l'espace de recherche, [Savasere et al., 1998] utilisent des taxonomies également parfois appelées hiérarchie de concepts représentant les connaissances du domaine afin de générer les règles négatives notées $X \not\Rightarrow Y$ à partir des positives déjà extraites par d'autres méthodes. Avec cette notation, on ne sait donc pas si la négation se situe sur la prémisse et/ou sur la conclusion. Ces règles extraites à l'aide de taxonomies vont porter le nom de règles générales. Une taxonomie est un regroupement d'objets similaires classés en catégories et sous-catégories. Un exemple de taxonomie est donné dans la figure 1.4.

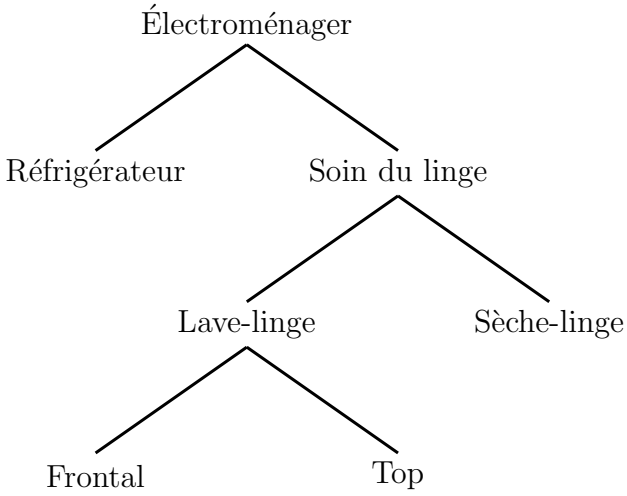

FIGURE 1.4 – Exemple de taxonomie

Dans cet exemple, le rayon électroménager est composé de réfrigérateurs et d'appareils pour le soin du linge. Dans les appareils pour le soin du linge, on retrouve les lave-linge ainsi que les sèche-linge. Enfin, il existe deux catégories de lave-linge, ceux qui s'ouvrent par le haut (*top*) et ceux qui s'ouvrent par le devant (*frontal*). Lors de l'extraction, les auteurs se basent sur une hypothèse d'uniformité : les objets appartenant à la même catégorie (*par exemple top et frontal qui sont deux types de lave-linge*) sont supposés intervenir dans les mêmes associations. Ainsi, si une marque M de lessive est fréquemment achetée avec un lave-linge frontal ($Frontal \Rightarrow Marque\ M$), on peut supposer que la même marque M de lessive est également fréquemment achetée avec un lave-linge top ($Top \Rightarrow Marque\ M$). Si ce n'est pas le cas, une règle négative est découverte ($Top \not\Rightarrow Marque\ M$). [Savasere et al., 1998] vont chercher à extraire uniquement les règles négatives $X \not\Rightarrow Y$ où le support de X et le support de Y sont supérieurs au support minimum et où l'intérêt de la règle est supérieur à l'intérêt minimum. Le support et l'intérêt minimum seront fournis par l'utilisateur. La mesure d'intérêt est défini comme suit : $intérêt = \frac{\varepsilon[sup(XY)] - sup(XY)}{sup(X)}$ où $\varepsilon[sup(XY)]$ correspond au support attendu $sup(X) \times sup(Y) \times n$.

Une approche similaire est présentée dans [Yuan et al., 2002], mais cette fois-ci les auteurs se focalisent sur les règles de substitution $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow Y$ (*nommées ainsi par* [Teng et al., 2002]). Dans l'exemple précédent, des règles de substitution seraient par exemple $Top \Rightarrow \overline{Frontal}$ ou $\overline{Top} \Rightarrow Frontal$, puisque en général les clients achètent une seule machine soit Top soit $Frontal$.

On trouve également une autre méthode dans [Subramanian et al., 2003] qui introduit une taxonomie « *et-ou* ». La particularité de cette taxonomie est la prise en compte

des relations du type « *est une partie de* », en plus de la relation « *est une* » présente dans les taxonomies « *ou* » déjà existantes dans la littérature. Dans l'exemple précédent, $Frontal$ est un $Lave - linge$. Les auteurs vont donc considérer $Frontal$ comme un item et $Lave - linge$ comme un concept. La hiérarchie « *est une* » implique que si un nœud fils a un support suffisant alors celui du nœud père l'est également. C'est-à-dire que l'on va considérer le concept $Lave - linge$ fréquent si l'item $Frontal$ est fréquent. Du point de vue des auteurs, cette propriété n'est pas toujours pertinente. Prenons par exemple la figure 1.5 qui représente un rayon sur les livres informatiques.

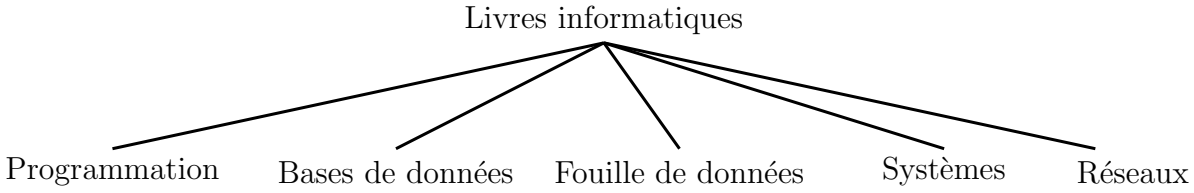

FIGURE 1.5 – Taxonomie « *est une* » non pertinente

Parmi ces livres informatiques on retrouve notamment des livres sur la programmation, sur les bases de données, sur la fouille de données, sur les systèmes et sur les réseaux. Dans cet exemple, on ne doit pas considérer que le concept $Livres\ informatiques$ est fréquent uniquement parce qu'il existe beaucoup de livres sur la $Programmation$. En effet, si les autres catégories de livres sont vides, le support de $Livres\ informatiques$ est vérifié uniquement grâce au support de $Programmation$. Les auteurs considèrent que les livres sur la programmation ne sont qu'une petite partie des livres informatiques et introduisent donc la notion « *est une partie de* ». [Subramanian et al., 2003] vont extraire les règles positives et négatives simultanément. Les règles contenant un concept en prémisse ou en conclusion seront extraites uniquement si l'ensemble des items ou concepts fils seront fréquents.

Quelques années plus tard, [Tsai et al., 2010] vont transformer la base de données en un fichier de transactions verticales. Ils vont ensuite utiliser la taxonomie pour filtrer les transactions qui n'appartiennent pas au domaine ou qui ne comportent pas d'item requis par l'utilisateur. Ils vont donc appliquer une phase de sélection des données avant d'extraire les règles.

Une autre approche utilise pour chaque attribut une taxonomie. On retrouve ce type d'approche dans [Wu et al., 2011] qui va extraire des associations négatives en proposant une extension de l'algorithme `Attribute-Oriented Induction` [Cai et al., 1990] qui extrait des connaissances générales en utilisant les données et les connaissances de l'utilisateur. Une connaisssance générale est négative si son support observé est sensiblement inférieur à son support attendu. Cette approche nous semble très semblable à celle de [Savasere et al., 1998].

[Taniar et al., 2012] utilisent une taxonomie afin d'extraire les règles positives les plus spécifiques et vont réduire le nombre de règles négatives en cherchant uniquement les règles négatives les plus générales.

En utilisant une taxonomie de bonne qualité et de bonne granularité, les utilisateurs vont pouvoir découvrir des règles négatives plus intéressantes. Cependant, il existe un inconvénient majeur à ces méthodes puisqu'elles nécessitent de fournir une taxonomie prédéfinie. Ces approches sont donc difficiles à généraliser puisqu'elles dépendent du domaine d'étude. Passons maintenant à des techniques plus génériques.

1.4.5 Extraction à l'aide de mesures de qualité

Afin d'être plus générique, d'autres approches décident de restreindre l'espace de recherche en se basant sur l'utilisation de mesures de qualité.

[Brin et al., 1997a] utilisent le test du χ^2 [Pearson, 1900] pour déterminer la dépendance entre deux motifs puis une mesure de corrélation pour déterminer si la dépendance est positive ou négative.

[Boulicaut et al., 2001] vont extraire les règles d'association négatives $XY \Rightarrow \overline{Z}$ et $\overline{V}W \Rightarrow Z$ de la base de données, en utilisant les informations concernant les attributs positifs pour induire les négations. Leur algorithme est générique et peut utiliser un ensemble de contraintes anti-monotones (*cf. définition* 9) et monotones (*cf. définition* 10). Les contraintes sont pour les auteurs des mesures. Comme exemple de mesure anti-monotone nous pouvons citer le support (*cf. sous-section* 1.3.3). Nous verrons par la suite une mesure monotone (*cf. section* 4.3). L'efficacité peut donc être décuplée puisqu'une conjonction de contraintes anti-monotones est anti-monotone et qu'une conjonction de contraintes monotones est monotone [Boulicaut et al., 2001].

➤ Extraction sans utiliser le support

Certaines méthodes n'utilisent pas la mesure du support lors de l'extraction. Afin d'extraire les règles négatives, [Thiruvady and Webb, 2004] proposent une extension de l'algorithme `Generalized Rules Discovery` [Webb, 2000] qui extrait les k règles les plus intéressantes. Cet algorithme n'utilise donc pas de support minimum mais utilise une mesure d'intérêt que l'utilisateur est libre de choisir, ainsi qu'une contrainte sur le nombre de règles à générer. Pour simplifier la recherche, les auteurs se limitent aux règles possédant un seul item en conclusion.

[Hämäläinen, 2012] propose un algorithme nommé `Kingfisher`, qui recherche des règles possédant un ensemble d'items en prémisse et un seul item en conséquence, sans utiliser la contrainte du support. L'approche peut utiliser une mesure de qualité générique mais elle s'est focalisée sur le test exact de Fisher [Fisher, 1925] et sur le test du χ^2 [Pearson, 1900]. Cet algorithme va pouvoir rechercher les k meilleures règles non redondantes ou l'ensemble des règles suffisamment intéressantes.

➤ Extraction dans un format condensé

Certaines approches préfèrent extraire les règles d'association dans un format plus condensé. Pour se faire, les approches étudiées ci-dessous utilisent la notion de motifs fermés. Un motif fermé est un ensemble maximal d'items communs à un ensemble de transactions. [Gasmi et al., 2007] proposent d'extraire un sous-ensemble des règles d'association avec négations représentant un condensé, à partir duquel ils arrivent à retrouver l'ensemble des règles d'association avec négations. Cette représentation condensée est donc basée sur les motifs fréquents fermés.

La majorité des approches pour l'extraction de règles d'association négatives utilise la notion de conjonction de motifs. [Hamrouni et al., 2010] vont proposer une approche autorisant également l'utilisation de disjonctions de motifs. Pour calculer le support d'un motif disjonctif ABC notée $\vee ABC$, il faut compter le nombre de transactions contenant au moins un des trois items. L'algorithme utilise également la notion de motifs fermés

et se déroule en trois étapes. La première étape va être l'extraction d'une représentation concise des motifs fréquents basés sur les motifs disjonctifs. Une structure partiellement ordonnée sera ensuite construite pour permettre la sélection des sous-ensembles de règles d'association représentatives, qui va donc permettre de diminuer le nombre de règles. La dernière étape va être de dériver les règles d'association générales à partir de la structure construite.

➤ Extractions basées sur Apriori

Plusieurs techniques reposant sur l'algorithme `Apriori` ont été trouvées dans la littérature. Ces algorithmes vont extraire en même temps les règles positives et négatives en se basant notamment sur les mesures du support et de la confiance.

☛ Règles usuelles :

Dans cette partie, les méthodes vont chercher à extraire les règles positives de la forme $X \Rightarrow Y$ ainsi que les règles négatives de la forme $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$.

[Wu et al., 2004] utilisent deux mesures supplémentaires pour extraire les règles valides : la valeur absolue de la nouveauté [Lavrac et al., 1999] qui permet de sélectionner les motifs qui sont inattendus au sens probabiliste et donc surprenants, ainsi que le facteur de certitude [Shortliffe, 1976] qui va permettre de connaître le type de règles à générer.

[Antonie and Zaïane, 2004] utilisent la mesure de corrélation de Pearson [Pearson, 1896] afin de n'étudier que les règles dont les motifs de la prémisse et de la conclusion sont bien corrélés. En fonction de la corrélation de la prémisse et de la conclusion, les auteurs vont étudier certains types de règles. Un processus de seuil automatique progressif est également présenté. Si lors de la première itération de l'algorithme aucune règle n'est extraite, le seuil minimum de corrélation est abaissé progressivement jusqu'à ce qu'un ensemble de règles non vides soit trouvé.

[Cornelis et al., 2006] se limitent à l'utilisation du support et de la confiance qu'ils jugent facilement paramétrables, afin que l'approche soit plus intuitive pour les utilisateurs.

[Dong et al., 2006] décident d'utiliser un seuil différent pour la confiance de chaque type de règles et se retrouvent ainsi avec quatre seuils pour les quatre formes de règles. Leur algorithme repose donc sur l'utilisation de ces quatre seuils pour la confiance mais également sur l'utilisation du test du χ^2 pour déterminer quels types de règles sont à étudier en fonction de la corrélation. Si le lift [Brin et al., 1997b] de la règle positive est supérieur à 1, alors les auteurs vont vérifier la confiance des règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$. Tandis que si le lift est inférieur à 1, alors ils se focalisent sur les règles $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow Y$. Les contraintes sur les différentes valeurs de la confiance leur permettent d'affiner plus facilement le nombre de règles extraites. [Dong et al., 2007a] reprennent leur méthode `MLMS` [Dong et al., 2007b] qui recherche simultanément les motifs fréquents et non fréquents en utilisant un seuil différent pour le support minimum pour chaque taille de motif. Ainsi un 3-motif nécessitera d'apparaître dans moins de transactions qu'un 2-motif pour être considéré comme fréquent. Par la suite, ils vont générer les règles positives à partir des motifs fréquents et les règles négatives à partir des motifs fréquents et des motifs non fréquents. L'utilisation d'une mesure combinant le coefficient de corrélation et la confiance va permettre d'élaguer les règles inintéressantes.

Tout comme [Cornelis et al., 2006], [Wang et al., 2008] vont uniquement utiliser les propriétés du support et de la confiance pour générer l'ensemble des règles positives et négatives. Pour les règles positives, une approche basée sur `Apriori` va être utilisée. Pour les règles négatives, l'algorithme utilise deux propriétés :

1. si $Y' \subset Y$ et $\overline{X} \Rightarrow Y$ est valide, alors $\overline{X} \Rightarrow Y'$ est valide,
2. si $Y \subset Y'$, $C_1 \in \{X, \overline{X}\}$ et $C_1 \Rightarrow \overline{Y}'$ est valide alors $C_1 \Rightarrow \overline{Y}$ est non valide.

La première propriété permet donc de générer les règles valides de la forme $\overline{X} \Rightarrow Y$ en étendant la conclusion des règles valides déjà obtenues, et d'élaguer les règles candidates en examinant les sous-motifs de taille $(k-1)$ dans les conclusions des règles valides. La deuxième propriété permet d'utiliser les règles potentielles (*non valides mais pouvant être valides en étendant leurs conclusions*) pour générer les règles valides en étendant leurs conclusions, et d'élaguer les règles candidates en examinant les sous-motifs de taille $(k-1)$ dans les conclusions des règles potentielles. La différence avec `Apriori` est que ce dernier utilise les motifs fréquents pour générer les règles et élaguer les motifs candidats, alors que leur approche utilise les règles positives valides pour générer les règles $\overline{X} \Rightarrow Y$ et les règles potentiellement valides pour générer les autres règles négatives.

☛ **Règles plus générales :**

[Gan et al., 2006] sont les premiers à s'intéresser à l'extraction des règles négatives sous sa forme la plus générale, c'est-à-dire des règles possédant des prémisses et des conclusions pouvant se composer d'items positifs x, d'items négatifs $\overline{x}$ et des négations de motifs positifs $\overline{X}$ qui représentent la présence non simultanée des items composant le motif X dans les transactions. Leur algorithme utilise un `arbre Patricia` [Pietracaprina and Zandolin, 2003], également connu sous le nom d'arbre radix qui est une amélioration de la structure `FP-tree` utilisée par l'algorithme `FP-Growth` [Han et al., 2000]. La première étape est donc de construire l'arbre Patricia. Ensuite ils recherchent en utilisant `FP-Growth`, l'ensemble des motifs fréquents, puis vont générer par niveau l'ensemble des motifs composés d'items positifs, d'items négatifs et des négations de motifs positifs. La dernière étape sera de générer l'ensemble des règles valides à partir des motifs précédemment générés.

[Kadir et al., 2011] recherchent les 1-motifs x fréquents ainsi que les 1-motifs négatifs $\overline{x}$ fréquents. L'algorithme va ensuite générer l'ensemble des motifs fréquents à partir de ces items. Les auteurs se retrouvent donc avec des motifs positifs, des motifs négatifs et des motifs de la forme $X\overline{Y}$. Lors de la recherche des règles, ils vont utiliser l'algorithme d'`Apriori` mais mettre de côté les règles $\overline{X} \Rightarrow \overline{Y}$ car ils considèrent que ce type de règles n'est pas intéressant dans l'analyse du panier de la ménagère. De plus, leur algorithme n'extrait pas de motifs du type $\overline{XY}$ mais des motifs du type $\overline{X}\,\overline{Y}$. Les règles associées ne seront donc pas des règles utilisant des négations de motifs mais des règles utilisant des conjonctions de motifs positifs et/ou négatifs.

Nous venons de découvrir quelques mesures utilisées dans l'extraction de règles d'association positives et négatives. Cependant on peut extraire des règles d'association grâce à d'autres indices et notamment des indices performants en présence de données volumineuses [Lerman and Azé, 2007] et [Lerman and Guillaume, 2013].

Les principaux travaux existants dans l'extraction de règles d'association négatives viennent d'être présentés. D'autres approches existent mais se focalisent sur certains

types de règles négatives que les auteurs ont appelées règles de substitution [Teng et al., 2002], règles d'exclusion [Amir et al., 1997], règles d'inférence [Gan et al., 2006] et règles d'exception [Suzuki and Shimura, 1996] [Gras et al., 2007]. Nous allons découvrir ces différents types de règles dans les sections suivantes.

1.4.6 Extraction de règles de substitution

Les règles de substitution, (*parfois appelées règles mixtes* [Missaoui et al., 2008]), ont pour but d'apporter des connaissances sur les motifs substituables. Par exemple, les personnes qui boivent du coca cola ne boivent pas de pepsi : *coca cola* $\Rightarrow \overline{pepsi}$. Cette règle signifie que les motifs *coca cola* et *pepsi* sont négativement corrélés et que *coca cola* est un substitut au *pepsi*. Dans l'exemple précédent, le substitut possède les mêmes propriétés que le substituable, cependant ce n'est pas toujours le cas. Par exemple, une boîte de chocolats peut très bien être un substitut à un bouquet de fleurs. Les règles de substitution ne sont pas toujours évidentes et peuvent donc renfermer des informations précieuses. [Teng et al., 2002] et [Teng et al., 2005] proposent l'algorithme `SRM` afin de découvrir uniquement les règles de substitution. Cet algorithme se déroule en deux phases. La première phase consiste à générer les motifs concrets parmi les motifs fréquents. Un motif concret est un motif fréquent dont les items sont statistiquement dépendants. La seconde phase va être la génération des règles de substitution à partir de ces motifs concrets.

1.4.7 Extraction de règles d'exclusion

[Amir et al., 1997] recherchent les règles d'exclusion de la forme $XY\overline{Z} \Rightarrow T$ avec $XY \Rightarrow T$ non valide à cause de la confiance. Leur proposition est de transformer la base de données en un arbre, puis d'extraire les règles positives et les règles d'exclusion directement à partir de l'arbre précédemment généré.

1.4.8 Extraction de règles d'inférence

Tout comme [Gan et al., 2006], [Missaoui et al., 2008] s'intéressent à l'extraction des règles les plus générales. Ils génèrent les règles d'association négatives à partir des règles positives mais dans un contexte bien particulier : celui des implications logiques.

Définition 12 - Implication logique, règle d'inférence, règle logique :
Une implication logique, également appelée règle d'inférence ou règle logique, est une règle dont la confiance est égale à 1. Cet état est représenté dans la figure 1.6.

Dans cette figure, $\mathcal{D}$ représente l'ensemble des transactions contenues dans la base de données. $\{t \in \mathcal{D} \mid X \subseteq t\}$ et $\{t \in \mathcal{D} \mid Y \subseteq t\}$ sont l'ensemble des transactions qui vérifient respectivement X et Y. Ici, $\{t \in \mathcal{D} \mid X \subseteq t\} \subseteq \{t \in \mathcal{D} \mid Y \subseteq t\}$ et par conséquent $sup(XY) = sup(X)$. Il existe donc bien une implication logique entre les motifs X et Y puisque la confiance de la règle $conf(X \Rightarrow Y) = \frac{sup(XY)}{sup(X)} = \frac{sup(X)}{sup(X)}$ est bien égale à 1.

Cette restriction aux implications logique permet d'utiliser des propriétés intéressantes et exploitables et notamment les axiomes d'Armstrong. Les trois principaux axiomes sont la transitivité, l'augmentation et la réflexivité. La transitivité définit que si $X \Rightarrow Y$

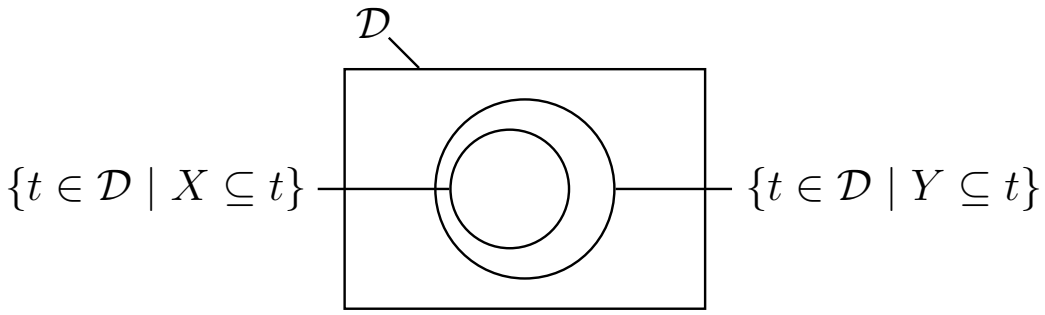

FIGURE 1.6 – Cas représentant l'implication logique entre les motifs X et Y

et $Y \Rightarrow Z$ sont valides alors $X \Rightarrow Z$ le sera également. L'augmentation relate le fait que si $X \Rightarrow Y$ est valide alors $XZ \Rightarrow Y$ le sera aussi pour tout Z. Et enfin, la réflexivité implique que si X contient Y alors $X \Rightarrow Y$ est vraie. Les axiomes complémentaires qui sont déduits des trois premiers peuvent également être utilisés. Ainsi l'union va permettre de dire que si $X \Rightarrow Y$ et $X \Rightarrow Z$ sont valides alors $X \Rightarrow YZ$ l'est également. La pseudo-transitivité permet de déduire $WX \Rightarrow Z$ si $X \Rightarrow Y$ et $WY \Rightarrow Z$ sont valides. Et enfin la décomposition dit que si $X \Rightarrow Y$ valide et Y contient Z alors $X \Rightarrow Z$ est bonne. Des extensions de ces travaux sont présents dans [Missaoui et al., 2010] et [Missaoui et al., 2012] et complètent notamment le système d'inférence précédent.

1.4.9 Extraction de règles d'exception

La notion de règles d'exception diverge en fonction du domaine dans lequel elle est utilisée. La première définition est celle introduite dans [Suzuki and Shimura, 1996] tandis que la deuxième est celle introduite par [Gras et al., 2007] et repose sur l'analyse statistique implicative.

➤ Exceptions selon Suzuki

La complexité du problème d'extraction des règles d'association augmente quand le nombre de variables de la base de données augmente. Pour une grande base de données, il faut donc utiliser une méthode afin de réduire l'espace de recherche. La principale méthode consiste à prendre en compte uniquement les données présentes dans un assez grand nombre de transactions. Cependant, l'objectif est d'extraire des règles intéressantes, c'est-à-dire, des règles qui apportent de l'information inattendue et surprenante à l'utilisateur. Malheureusement, il parait probable que les connaissances apportées par les règles valides possédant un support élevé soient déjà connues des experts. Il faut donc arriver à extraire des règles représentant des pépites de connaissances. Les règles d'exception appartiennent à cette catégorie.

Au sens littéral, une exception est une dérogation à une règle générale. Une règle d'exception est donc une règle qui va décrire les contre-exemples de la règle générale. Par définition, une exception est satisfaite par un nombre assez réduit de transactions, ce qui implique que le support ne va pas permettre de juger de la qualité de ces règles. Les règles d'exception sont donc des règles inattendues qui révèlent des connaissances intéressantes.

Afin d'illustrer la différence entre règle générale et règle d'exception, prenons un exemple dans le règne animal. On peut considérer que tous les animaux qui possèdent

des ailes peuvent voler. Pour représenter cette règle générale on va utiliser $ailes \Rightarrow voler$. Néanmoins, il existe certains animaux qui possèdent bien des ailes mais qui ne peuvent pas voler : par exemple l'autruche. Afin de tenir compte de ce cas particulier, une règle d'exception est créée : $ailes, autruche \Rightarrow \overline{voler}$. Cette règle va s'appliquer dans une situation plus spécifique que la règle générale. En résumé, on se retrouve avec deux règles qui possèdent des conclusions contradictoires et dont les prémisses sont identiques à un motif près.

Plus généralement, la découverte de règles d'exception consiste à extraire des couples de règles sous la forme :

— $X \Rightarrow Y$ qui correspond à une règle générale,
— et $XZ \Rightarrow \overline{Y}$ qui correspond à une règle d'exception.

Dans la littérature, on trouve deux approches afin d'extraire les règles d'exception. L'approche objective, également appelée approche indirecte est une approche dirigée par les données. Elle consiste à utiliser un algorithme qui va extraire en même temps les règles générales et les règles d'exception. Ces algorithmes vont utiliser différentes mesures de qualité objectives afin de juger de l'intérêt des règles extraites. Une mesure objective mesure l'intérêt d'une règle en fonction de sa structure et des données utilisées dans le processus de découverte. Puisque ces méthodes sont assez gourmandes en temps de calcul, l'espace de recherche est souvent restreint en imposant le nombre d'items dans les règles, la présence de certains motifs, ou encore le nombre de règles à extraire.

☛ Approches objectives

La première méthode objective fut `MEPRO` introduite dans [Suzuki and Shimura, 1996]. Elle utilise un algorithme par séparation et évaluation (*branch and bound*) et se base sur les propriétés de la J-mesure [Smyth and Goodman, 1992] pour accélérer l'extraction. La mesure permet de quantifier le contenu de l'information d'une règle et va être appliquée au couple règle générale / règle d'exception. Un classement des règles selon leurs intérêts va donc être effectué et l'algorithme retournera les k règles les plus intéressantes. Cependant, cette méthode génère certaines règles d'exception qui sont dues au hasard. Pour résoudre ce problème, [Suzuki, 1997] utilise cinq contraintes reposant sur le support et la confiance pour évaluer l'intérêt d'un couple de règles. Parmi ces contraintes, il faut que le support de la prémisse de la règle générale $sup(X)$ et le support de la prémisse de la règle d'exception $sup(XZ)$ soient supérieurs ou égaux à deux seuils définis par l'utilisateur. Il faut également que la confiance de la règle générale $conf(X \Rightarrow Y)$ et la confiance de la règle d'exception $conf(XZ \Rightarrow \overline{Y})$ soient supérieures ou égales à deux autres seuils définis par l'utilisateur. Et enfin, il faut vérifier que la règle $Z \Rightarrow \overline{Y}$, appelée règle de référence, ne soit pas valide et donc que sa confiance $conf(Z \Rightarrow \overline{Y})$ soit inférieure ou égale à un dernier seuil renseigné par l'utilisateur. En effet, sans la dernière condition, une règle d'exception pourrait être inférée à partir de la règle de référence et ne serait donc pas inattendue. Le problème de cette méthode, est qu'elle demande à l'utilisateur de renseigner cinq seuils qui sont assez difficiles à fixer car nécessitant une bonne connaissance des données mais également du processus d'extraction. En effet, des seuils trop restrictifs peuvent éliminer des règles intéressantes et au contraire des seuils trop permissifs vont produire des règles inintéressantes. Pour gérer ces problèmes, [Suzuki, 1999] propose une méthode qui met à jour les seuils à chaque itération pour s'adapter aux données. Une étude comparative de ces trois méthodes

est disponible dans [Suzuki, 2004]. Afin d'améliorer la qualité des règles extraites pour découvrir les règles les plus inattendues et surprenantes, [Suzuki and Kodratoff, 1998] utilisent une version modifiée de l'intensité d'implication [Gras, 1979]. L'intensité d'implication évalue si le nombre de contre-exemples (*ceux qui vérifient la prémisse mais qui ne vérifient pas la conclusion*) est significativement faible, mais elle n'arrive pas à classer les règles en fonction de leur degré de surprise, ce qui explique la modification. [Suzuki and Zytkow, 2000] définissent la recherche de règles d'exception intéressantes comme la recherche d'un triplet de règles ($X \Rightarrow Y$, $XZ \Rightarrow \overline{Y}$ et $Z \not\Rightarrow \overline{Y}$) puis classent ces triplets en onze types. Un algorithme est ensuite proposé pour l'extraction simultanée des onze types de triplets. [Hussain et al., 2000] vont utiliser l'algorithme `Apriori` pour extraire l'ensemble des règles générales, puis ils vont vérifier pour chaque paire de règles générales $X \Rightarrow Y$ et $Z \Rightarrow Y$ le support de XZ. Si XZ n'est pas fréquent alors $XZ \Rightarrow \overline{Y}$ est considérée comme une règle d'exception candidate. Un balayage de la base de données permet de vérifier le support et la confiance des règles candidates puis la divergence de Kullback–Leibler (*mesure de dissimilarité définie dans* [Kullback and Leibler, 1951]) est utilisée afin de vérifier de façon objective l'intérêt d'une règle d'exception.

☛ Approches subjectives

L'approche subjective ou directe est une approche dirigée par l'utilisateur puisque les règles générales sont obtenues grâce à un expert métier qui fournit les connaissances du domaine, également appelées croyances. Pour évaluer l'intérêt de ces règles d'exception, les méthodes vont utiliser principalement des mesures subjectives. Une mesure subjective mesure l'intérêt d'une règle en fonction de sa structure et des données utilisées dans le processus de découverte, mais dépend également de l'utilisateur qui étudie les données. Il se peut donc qu'une règle intéressante pour un utilisateur ne le soit pas pour un autre. Ainsi [Piatetsky-Shapiro and Matheus, 1994] définissent qu'un motif intéressant est actionnable, c'est-à-dire s'il permet la prise de décision sur des actions futures. Il faut que la connaissance apportée permette à l'utilisateur d'en tirer un avantage (*augmenter un bénéfice, diminuer un coût, améliorer l'efficacité...*). [Silberschatz and Tuzhilin, 1996] utilisent un système de croyances et proposent d'évaluer l'intérêt avec l'inattendu, qui stipule qu'une règle est intéressante si elle surprend l'expert. Chaque croyance va être classée en croyance faible ou en croyance forte en fonction du degré de certitude qu'ils ont par rapport à celle-ci. Au cours de l'analyse, le degré des croyances faibles va pouvoir évoluer tandis que celui des croyances fortes ne changera pas. Par la suite [Padmanabhan and Tuzhilin, 1998] et [Padmanabhan and Tuzhilin, 1999] proposent des algorithmes pour l'extraction des règles inattendues. Pour chaque croyance $X \Rightarrow Y$, ils vont utiliser l'algorithme `ZoomUR` qui commence par découvrir toutes les règles d'exception qui contredisent le système de croyances $XZ \Rightarrow \overline{Y}$, en utilisant un algorithme basé sur `Apriori` et qui utilise donc le support et la confiance (*mesures objectives*). Par la suite, ils cherchent des règles plus générales $X'Z \Rightarrow \overline{Y}$ avec $X' \subset X$ à partir des exceptions de la première phase. Par exemple, l'expert fournit comme croyance que les animaux qui possèdent des ailes volent : $ailes \Rightarrow voler$. L'algorithme va commencer par trouver une règle plus élaborée $ailes, autruche \Rightarrow \overline{voler}$ puis une règle plus générale $autruche \Rightarrow \overline{voler}$ qui est totalement différente de la croyance de départ. Par la suite, [Padmanabhan and Tuzhilin, 2000] poursuivent ce travail avec `MinZoomUR` en réduisant le nombre de règles d'exception extraites en se basant notamment sur un ensemble minimal de motifs inattendus, afin d'éliminer une redondance dans les règles provenant du traitement individuel de chaque

croyance. [Liu et al., 1999b] proposent une approche qui utilise l'analyse de la déviation pour identifier les règles d'exception intéressantes. Cette approche est également flexible puisqu'elle fonctionne sur des règles générales pouvant être apportées par l'expert ou extraites à partir des données.

Deux articles état de l'art sur l'extraction de règles d'exception sont disponibles dans [Suzuki, 2006] et [Duval et al., 2007].

➤ **Exceptions selon Gras**

En analyse statistique implicative, la notion de règles d'exception est légèrement différente par rapport à celle présentée précédemment. En effet, si $X \Rightarrow Z$ et $Y \Rightarrow Z$, une règle d'exception est $XY \Rightarrow \overline{Z}$. Ainsi, si deux variables en impliquent une troisième, et que leur conjonction implique la négation, on considère que l'on est en présence d'une règle d'exception. A notre connaissance, cette définition de règles d'exception n'a été utilisée que dans [Gras et al., 2007] et [Gras et al., 2013]. Les auteurs proposent deux approches pour mettre en évidence ces règles. La première approche se base sur l'utilisation de l'intensité d'implication [Gras, 1979] qui va permettre de rejeter les règles $XY \Rightarrow Z$ qui ont une faible valeur pour cette mesure, et à contrario va valider les règles $XY \Rightarrow \overline{Z}$ possédant une forte valeur. La seconde approche est une extension des règles en R-règles (*règles de règles*) de type $R \Rightarrow R'$ [Gras and Kuntz, 2006] qui permet d'extraire de façon graphique les règles d'exception. Cette seconde approche est basée sur une représentation hiérarchique des règles et utilise un indice de cohésion pour mettre en avant une relation entre motifs ou entre règles.

1.5 Conclusion

La première partie de ce chapitre présente le processus d'extraction des connaissances à partir des données. Nous avons ensuite étudié le problème d'extraction des règles d'association positives et présenté notamment l'algorithme de référence `Apriori`. Par la suite, nous avons exposé le problème d'extraction des règles négatives ainsi qu'un état de l'art sur les différentes méthodes existantes.

Nous recensons donc plusieurs types de règles d'association négatives : les règles de substitution qui permettent de trouver des motifs substituables, les règles d'exclusion qui ajoutent une négation d'item dans la prémisse d'une règle afin de la rendre valide, les règles d'inférence qui représentent des implications logiques entre motifs et les règles d'exception qui définissent des cas particuliers aux règles générales.

Certains algorithmes se concentrent sur un unique type de règles négatives. Cette restriction à un certain type de règles ne leur permet pas d'extraire l'ensemble des règles d'association négatives intéressantes. D'autres approches vont choisir d'utiliser une taxonomie afin de guider leur méthode d'extraction. Cependant, ces méthodes ne sont pas généralisables puisqu'elles dépendent du domaine d'étude et nécessitent des connaissances que seul un expert connaissant parfaitement les données peut fournir. Les approches les plus générales sont finalement celles qui sont basées sur *Apriori*. Nous allons cependant exclure de l'étude l'approche de [Dong et al., 2006] qui utilise un seuil différent pour la confiance de chaque type de règles, ainsi que celle de [Dong et al., 2007a] qui utilise un seuil différent du support pour chaque taille de motifs. Ces deux approches, bien que basées sur Apriori, semblent difficiles à évaluer puisque les auteurs ne renseignent pas comment choisir les valeurs des différents seuils à définir. Quant à l'approche de [Wang et al., 2008],

elle semble être un dérivé de celle de [Cornelis et al., 2006] dont l'amélioration ne semble pas évidente. Nous allons donc nous focaliser sur les méthodes qui nous semblent les plus prometteuses, à savoir celles de [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006].

Chapitre 2

Extraction de règles d'association positives et négatives

Sommaire

2.1 Introduction

Dans ce chapitre, nous allons présenter plus en détail les trois propositions majeures existantes pour extraire les règles d'association positives et négatives. Ces trois méthodes, [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006], reposent sur l'algorithme `Apriori` [Agrawal and Srikant, 1994] et utilisent les mesures classiquement utilisées (*support et confiance*) afin de sélectionner les règles valides. Ces trois algorithmes vont donc extraire en plus des règles positives $X \Rightarrow Y$, les règles négatives suivantes $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$. Chaque algorithme va proposer des critères différents pour définir ce qu'est une règle intéressante et ne va donc pas extraire le même ensemble de règles valides. Toutefois, pour chaque algorithme, le processus d'extraction se décompose en deux phases distinctes. La première phase est la recherche des motifs candidats, et la seconde phase consiste à générer les règles valides à partir de ces motifs candidats. En présentant les méthodes utilisées par ces auteurs, nous mettons en évidence

leurs avantages et leurs inconvénients. Et enfin, nous reprenons la base de données utilisée pour présenter `Apriori`(*cf. section* 1.2) et déroulons chaque algorithme sur ce petit exemple fil-rouge.

La présentation des méthodes se fera par ordre chronologique. Nous commençons donc par la méthode de [Wu et al., 2004].

2.2 Algorithme proposé par Wu *et al.*

Dans ce qui suit, nous présentons tout d'abord les différentes contraintes que doivent respecter les règles pour être considérées comme valides et être extraites par la méthode de [Wu et al., 2004]. Nous présentons ensuite les différents algorithmes utilisés au cours de leur processus d'extraction. Le processus d'extraction se divise en deux parties, à savoir la recherche des motifs fréquents et non fréquents potentiellement intéressants et l'extraction des règles à partir des motifs potentiellement intéressants précédemment trouvés. Nous déroulons ensuite l'algorithme sur notre exemple fil-rouge.

2.2.1 Critères de validité d'une règle

Il existe certaines ambiguïtés et contradictions entre le texte et les algorithmes mais nous allons expliquer comment nous avons compris cet article. La première ambiguïté provient des traitements effectués par la fonction *nfip* que nous allons présenter par la suite. La seconde ambiguïté survient dans la méthode générant les règles. Dans les deux cas, cela provient de l'ambivalence de la définition d'une règle négative. En effet, dans le texte les auteurs simplifient/généralisent la notation d'une règle négative en utilisant la forme $X \Rightarrow \overline{Y}$. Par la suite, la fonction *nfip* semble utiliser cette généralisation alors que ce n'est plus le cas lors de la génération des règles.

[Piatetsky-Shapiro, 1991] explique qu'une règle $X \Rightarrow Y$ n'est pas intéressante si l'inégalité suivante est vérifiée : $P(XY) \simeq P(X) \times P(Y)$. Plus explicitement, une règle n'est pas intéressante si sa prémisse et sa conclusion sont approximativement indépendantes. De manière générale, l'indépendance désigne l'absence d'influence entre deux variables ou événements. En fouille de données, deux motifs sont dit indépendants lorsque l'apparition d'un des motifs n'affecte pas l'apparition de l'autre, c'est-à-dire lorsque $sup(XY) = sup(X) \times sup(Y)$. Si la réalisation de la prémisse ne fournit aucune information sur la réalisation de la conclusion, il est inutile d'étudier des règles issues de motifs indépendants (*ou trop proches de cet état*). Un exemple de cet état d'indépendance est représenté dans la figure 2.1.

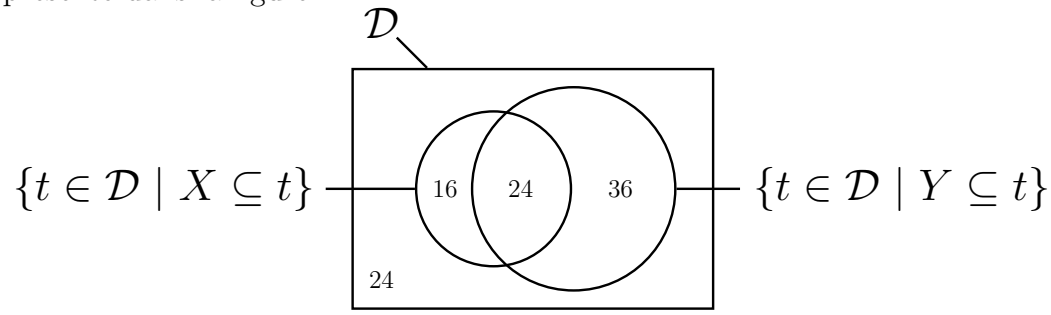

FIGURE 2.1 – Exemple d'indépendance entre les motifs X et Y

Dans cette figure, $\mathcal{D}$ représente l'ensemble des 100 transactions contenues dans la base de données. $\{t \in \mathcal{D} \mid X \subseteq t\}$ et $\{t \in \mathcal{D} \mid Y \subseteq t\}$ sont l'ensemble des transactions qui vérifient respectivement X et Y. Ici, $sup(X) = \frac{16+24}{100} = 0,40$ et $sup(Y) = \frac{24+36}{100} = 0,60$. Il existe donc bien une indépendance entre les motifs X et Y puisque $sup(XY) = sup(X) \times sup(Y)$ puisqu'en effet $sup(XY) = \frac{24}{100} = 0,24$.

[Wu et al., 2004] vont s'inspirer de cette définition et proposer la mesure d'intérêt suivante : $int(X,Y) = |sup(XY) - sup(X) \times sup(Y)|$ ainsi qu'un seuil minimum d'intérêt min_{int}. Cette mesure correspond en réalité à la valeur absolue de la nouveauté [Lavrac et al., 1999] et va permettre de vérifier que l'écart de la règle à l'indépendance est assez important.

Les règles positives sont générées à partir des motifs fréquents d'intérêt potentiel. L'intérêt potentiel va être défini à l'aide d'une mesure composite puisqu'elle combine à la fois le support, la confiance et l'intérêt. Un motif sera considéré comme d'intérêt potentiel si les valeurs du support, de la confiance et de l'intérêt sont supérieures ou égales aux seuils fixés par l'utilisateur. L'intérêt potentiel d'un motif fréquent M sera vérifié par la fonction *fip* :

$$fip(M) = (sup(M) \geqslant min_{sup}) \wedge (\exists X, Y \text{ tel que } X \cup Y = M) \wedge fips(X,Y)$$

avec *fips* définie comme suit :

$$fips(X,Y) = (X \cap Y = \varnothing) \wedge (f(X,Y,min_{sup},min_{conf},min_{int}) = 1)$$

et la fonction f :

$$f(X,Y,min_{sup},min_{conf},min_{int}) = \frac{sup(XY)+conf(X \Rightarrow Y)+int(X,Y)-(min_{sup}+min_{conf}+min_{int})+1}{|sup(XY)-min_{sup}|+|conf(X \Rightarrow Y)-min_{conf}|+|int(X,Y)-min_{int}|+1}$$

L'inconvénient majeur de cette mesure provient du fait qu'elle vérifie la supériorité par rapport aux seuils minimaux des trois mesures en même temps. C'est-à-dire qu'elle va vérifier en une seule fois, si le support du motif est supérieur au seuil de support minimum, si la confiance de la règle est supérieure au seuil de confiance minimum et si l'intérêt de la règle est supérieur au seuil d'intérêt minimum. Ainsi, si le support ne vérifie pas le seuil minimal, la confiance et l'intérêt vont quand même être calculés, même si nous savons par avance que le motif étudié ne sera pas un motif fréquent d'intérêt potentiel. De plus, la vérification de la confiance ($conf(X \Rightarrow Y) \geqslant min_{conf}$) dans cette mesure composite va empêcher l'utilisation de la propriété de la confiance (*cf. propriété* 1). L'utilisation d'une conjonction de contraintes est intéressante lorsque l'ensemble des contraintes possède une propriété anti-monotone ou monotone [Boulicaut et al., 2001]. Or, ce n'est pas le cas ici. En effet, le support est la seule mesure anti-monotone et on est en présence d'une mesure composite et non d'une conjonction de contraintes ($C_1 \wedge C_2 \wedge C_3$).

Les règles négatives, quant à elles, sont générées à partir des motifs non fréquents d'intérêt potentiel. Il faut tout d'abord vérifier que le motif positif M ne soit pas fréquent. M est un motif non fréquent d'intérêt potentiel (*vérifié par la fonction nfip*) si :

$$nfip(M) = (sup(M) < min_{sup}) \wedge (\exists M_1, M_2 \text{ tel que } M_1 \cup M_2 = M) \wedge nfips(M_1, M_2)$$

avec *nfips* définie comme suit :

$$nfips(M_1, M_2) = (M_1 \cap M_2 = \varnothing) \wedge (g(M_1, M_2, min_{sup}, min_{conf}, min_{int}) = 2)$$

et la fonction g :

$$g(M_1, M_2, min_{sup}, min_{conf}, min_{int}) = f(M_1, M_2, min_{sup}, min_{conf}, min_{int}) + \frac{sup(M_1)+sup(M_2)-2\times min_{sup}+1}{|sup(M_1)-min_{sup}|+|sup(M_2)-min_{sup}|+1}$$

avec $M_1 \in \{X, \overline{X}\}$, $M_2 \in \{Y, \overline{Y}\}$ et $X \cup Y \neq M$.

Nous retrouvons dans la fonction *nfip* les mêmes inconvénients que dans la fonction *fip* puisque cette fonction va calculer la confiance et l'intérêt du motif, alors que les supports des motifs M_1 et M_2 risquent d'être insuffisants. Il aurait été plus pertinent pour ces deux méthodes de commencer par vérifier les différents supports, puis la confiance et enfin l'intérêt.

Le facteur de certitude [Shortliffe, 1976] (*noté fc*) va ensuite être utilisé afin de connaître le type de règles à générer. Le facteur de certitude est défini en fonction de la zone où se situe la règle :

Zone où conf$(\mathbf{X \Rightarrow Y}) \geqslant$ sup$(\mathbf{Y})$:
$$fc(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - sup(Y)}{1 - sup(Y)}$$

Zone où conf$(\mathbf{X \Rightarrow Y}) <$ sup$(\mathbf{Y})$:
$$fc(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - sup(Y)}{sup(Y)}$$

Lorsque la réalisation de la prémisse X augmente les chances d'apparition de la conclusion Y, c'est-à-dire lorsque $conf(X \Rightarrow Y) \geqslant sup(Y)$, le facteur de certitude détermine la distance de la règle $X \Rightarrow Y$ entre l'indépendance et l'implication logique. Et lorsque la réalisation de X diminue les chances d'apparition de Y, c'est-à-dire lorsque $conf(X \Rightarrow Y) < sup(Y)$, cette mesure évalue la distance de la règle entre l'incompatibilité et l'indépendance. L'état d'incompatibilité intervient quand la confiance de la règle est nulle. Cet état est représenté dans la figure 2.2.

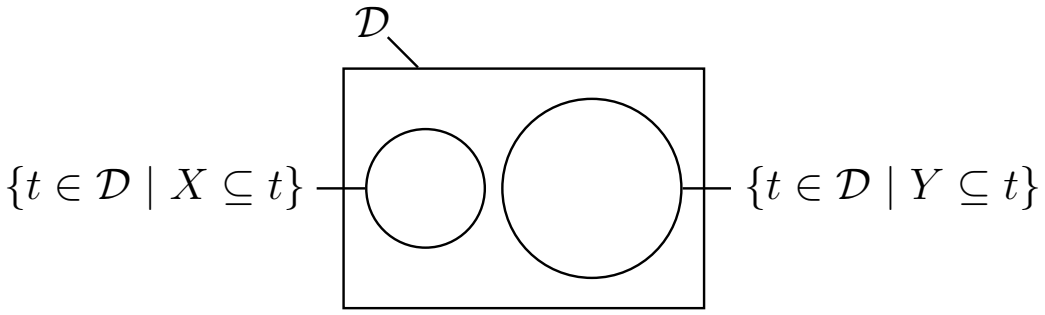

FIGURE 2.2 – Cas représentant l'incompatibilité entre les motifs X et Y

Il existe donc bien une incompatibilité entre les motifs X et Y puisque $conf(X \Rightarrow Y) = \frac{sup(XY)}{sup(X)} = 0$ puisqu'en effet $sup(XY) = 0$.

Cette mesure va donc permettre de garder uniquement les règles dont les motifs sont bien corrélés. Une règle est générée si la valeur du facteur de certitude est supérieure ou égale à min_{conf}. Cette contrainte est très forte car la valeur de min_{conf} est souvent proche de 1, ce qui peut potentiellement entraîner une perte de règles intéressantes. L'intérêt d'utiliser simultanément la mesure d'intérêt et le facteur de certitude est limité puisque ces deux mesures permettent d'éliminer les règles possédant des motifs indépendants.

Le tableau 2.1 récapitule les contraintes pour qu'une règle soit considérée comme valide selon leurs critères :

$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$
$sup(XY) \geqslant min_{sup}$	$sup(XY) < min_{sup}$
	$sup(\overline{X}Y) \geqslant min_{sup}$
$conf(X \Rightarrow Y) \geqslant min_{conf}$	$conf(\overline{X} \Rightarrow Y) \geqslant min_{conf}$
$int(X, Y) \geqslant min_{int}$	$int(\overline{X}, Y) \geqslant min_{int}$
$fc(Y\|X) \geqslant min_{conf}$	$fc(Y\|\overline{X}) \geqslant min_{conf}$
	$sup(X) \geqslant min_{sup}$
	$sup(Y) \geqslant min_{sup}$
$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$
$sup(XY) < min_{sup}$	$sup(XY) < min_{sup}$
$sup(X\overline{Y}) \geqslant min_{sup}$	$sup(\overline{X}\,\overline{Y}) \geqslant min_{sup}$
$conf(X \Rightarrow \overline{Y}) \geqslant min_{conf}$	$conf(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{conf}$
$int(X, \overline{Y}) \geqslant min_{int}$	$int(\overline{X}, \overline{Y}) \geqslant min_{int}$
$fc(\overline{Y}\|X) \geqslant min_{conf}$	$fc(\overline{Y}\|\overline{X}) \geqslant min_{conf}$
$sup(X) \geqslant min_{sup}$	$sup(X) \geqslant min_{sup}$
$sup(Y) \geqslant min_{sup}$	$sup(Y) \geqslant min_{sup}$

TABLEAU 2.1 – Règles valides

En conclusion, l'ensemble des règles $X \Rightarrow Y$, $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$ doivent respecter les différents seuils minimums pour le support, la confiance, l'intérêt ainsi que le facteur de certitude. Les règles négatives doivent en plus vérifier que le support des deux motifs positifs composant la règle est fréquent (*i.e.* $sup(X) \geqslant min_{sup}$ *et* $sup(Y) \geqslant min_{sup}$) mais que le support de leur combinaison XY est non fréquent.

Nous allons maintenant présenter la méthode utilisée par [Wu et al., 2004] pour extraire les règles d'association positives et négatives valides.

2.2.2 Méthode d'extraction

La méthode d'extraction est divisée en deux étapes distinctes. La première étape génère tous les motifs fréquents et non fréquents potentiellement intéressants à l'aide de la fonction *MIP* (*cf. algorithme* 5).

Algorithme 5 : *MIP* - Recherche des Motifs d'Intérêt Potentiel

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, confiance minimum min_{conf}, intérêt minimum min_{int}
Sorties : ensemble des Motifs Fréquents d'Intérêt Potentiel $MFIP$ et ensemble des Motifs Non Fréquents d'Intérêt Potentiel $MNFIP$

```
1  MFIP = ∅
2  MNFIP = ∅
3  F_1 = {i ∈ I tel que sup(i) ⩾ min_sup}
4  MFIP = MFIP ∪ F_1
5  pour (k = 1; F_k ≠ ∅; k++) faire
6      C_{k+1} = F_k ⋈ F_k
7      pour tout motif M ∈ C_{k+1} faire
8          s = support(D, M)
9          si s ⩾ min_sup alors
10             F_{k+1} = F_{k+1} ∪ {M}
11         sinon
12             NF_{k+1} = NF_{k+1} ∪ {M}
13     pour tout motif M ∈ F_{k+1} faire
14         si !fip(M) alors
15             F_{k+1} = F_{k+1} − M
16         MFIP = MFIP ∪ F_{k+1}
17     pour tout motif M ∈ NF_{k+1} faire
18         si !nfip(M) alors
19             NF_{k+1} = NF_{k+1} − M
20         MNFIP = MNFIP ∪ NF_{k+1}
21 retourner MFIP et MNFIP
```

L'algorithme commence par initialiser l'ensemble des motifs fréquents d'intérêt potentiel $MFIP$ et l'ensemble des motifs non fréquents d'intérêt potentiel $MNFIP$ aux ensembles vides (*lignes 1 et 2*). Les auteurs récupèrent ensuite dans F_1 les items fréquents, c'est-à-dire les items dont le support est supérieur au seuil minimum du support fixé par l'utilisateur (*ligne 3*). F_1 est ensuite stocké dans l'ensemble $MFIP$ des motifs fréquents d'intérêt potentiel (*ligne 4*). Cette étape est inutile puisqu'il faut au moins deux items pour générer une règle. Le processus suivant (*lignes 5 à 20*) va être réitéré jusqu'à ce que l'on n'obtienne plus de motif fréquent ($F_k \neq \varnothing$). Tout d'abord, les *(k+1)*-motifs candidats vont être générés à partir des k-motifs fréquents (*ligne 6*). Cette phase n'est pas optimisée car elle n'utilise pas la fonction *candidats* standard qu'utilise `Apriori` (*cf. algorithme* 2) pour générer les motifs candidats. Leur méthode consiste simplement à combiner deux k-motifs fréquents pour obtenir un motif candidat de taille *k+1*. Contrairement à `Apriori`,

deux k-motifs peuvent être combinés pour créer un nouveau *(k+1)*-motif candidat même s'ils n'ont pas en commun les *(k-1)* premiers items. Ainsi AC et BC peuvent se combiner pour créer le motif ABC puisqu'ils ont l'item C en commun. Cette manière de procéder va engendrer deux problèmes. Premièrement, un k-motif va pouvoir être généré de k manières différentes. En effet, le motif ABC peut être combiné à partir de (AB et AC), (AB et BC) et également (AC et BC). Deuxièmement, les auteurs ne vont pas vérifier que la totalité des sous-ensembles sont fréquents comme le font les auteurs d'`Apriori`. Par exemple, si AB est non fréquent, le motif ABC va pouvoir être généré à partir de la combinaison de AC et BC. Cette phase va donc entraîner la création d'une multitude de motifs non fréquents.

Ensuite, pour tous les motifs M de C_{k+1} (*ligne 7*), on calcule le support avec la fonction *support* (*ligne 8*) qui interroge la base de données. Puis, si le motif M est fréquent (*ligne 9*), alors on le stocke dans l'ensemble F_{k+1} qui servira à générer les candidats de niveau supérieur à la prochaine itération. Sinon (*ligne 11*), le motif M est ajouté à l'ensemble NF_{k+1} des motifs non fréquents. Les deux dernières étapes (*lignes 13 à 16*) et (*lignes 17 à 20*) vont ajouter les motifs fréquents et non fréquents d'intérêt potentiel aux ensembles $MFIP$ et $MNFIP$. Pour l'ensemble $MFIP$, les auteurs parcourent l'ensemble des motifs fréquents contenus dans F_{k+1} (*ligne 13*) et vont tester à l'aide de la fonction *fip* l'intérêt potentiel de chaque motif. Si le motif M n'est pas un motif fréquent d'intérêt potentiel (*ligne 14*) alors M est retiré de l'ensemble F_{k+1} (*ligne 15*). Nous constatons également deux problèmes à ce niveau. Tout d'abord, la mesure *fip* ne possède pas de propriété anti-monotone. Nous ne pouvons donc pas déduire l'intérêt potentiel d'un sur-ensemble à partir d'un motif. Par conséquent, le fait de supprimer le motif X de l'ensemble F_{k+1} lorsque l'intérêt potentiel n'est pas vérifié, va empêcher de générer un motif XY à la prochaine itération qui aurait pu respecter l'intérêt potentiel. Ensuite, lorsque l'un des couples de motifs $X \cup Y = M$ ne respecte pas le support, la confiance et l'intérêt, le motif M est supprimé de l'ensemble F_{k+1} puis F_{k+1} est ajouté à l'ensemble $MFIP$ des motifs fréquents d'intérêt potentiel (*ligne 16*). Le fait de supprimer le motif à cause d'une seule combinaison non valide, va éliminer un grand nombre de combinaisons valides. Le même processus d'élimination est appliqué à l'ensemble NF_{k+1} des motifs non fréquents en utilisant la fonction *nfip*. Une fois les motifs inintéressants éliminés, l'ensemble NF_{k+1} est ajouté à l'ensemble $MNFIP$ des motifs non fréquents d'intérêt potentiel (*ligne 20*). Afin d'éliminer uniquement les combinaisons non valides et non tout le motif comme le font les auteurs, nous pensons qu'il est plus approprié que la phase de recherche des motifs d'intérêt potentiel (*lignes 13 à 20*) se fasse uniquement dans la deuxième partie de l'algorithme et que la première phase consiste simplement à chercher les motifs fréquents et non fréquents.

La deuxième étape de la méthode d'extraction génère les règles positives à partir des motifs fréquents d'intérêt potentiel et les règles négatives à partir des motifs non fréquents d'intérêt potentiel. Cette étape est réalisée par l'*algorithme* 6.

Algorithme 6 : Extraction des règles d'association positives et négatives

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, confiance minimum min_{conf}, intérêt minimum min_{int}
Sortie : ensemble des règles valides R

```
1  R = ∅
2  (MFIP, MNFIP) = MIP(D, min_sup, min_conf, min_int)
3  pour tout motif M ∈ MFIP faire
4      pour toute combinaison X ∪ Y = M faire
5          si fips(X, Y) alors
6              si fc(Y|X) ⩾ min_conf alors
7                  R = R ∪ {X ⇒ Y}
8              si fc(X|Y) ⩾ min_conf alors
9                  R = R ∪ {Y ⇒ X}
10 pour tout motif M ∈ MNFIP faire
11     pour toute combinaison X ∪ Y = M faire
12         si nfips(X, Y) alors
13             si fc(Y|X̄) ⩾ min_conf alors
14                 R = R ∪ {X̄ ⇒ Y}
15             si fc(X̄|Y) ⩾ min_conf alors
16                 R = R ∪ {Y ⇒ X̄}
17             si fc(Ȳ|X) ⩾ min_conf alors
18                 R = R ∪ {X ⇒ Ȳ}
19             si fc(X|Ȳ) ⩾ min_conf alors
20                 R = R ∪ {Ȳ ⇒ X}
21             si fc(Ȳ|X̄) ⩾ min_conf alors
22                 R = R ∪ {X̄ ⇒ Ȳ}
23             si fc(X̄|Ȳ) ⩾ min_conf alors
24                 R = R ∪ {Ȳ ⇒ X̄}
25 retourner R
```

Cet algorithme commence par initialiser l'ensemble R des règles valides à l'ensemble vide (*ligne 1*). Puis, les auteurs font appel à la fonction *MIP* (*cf. algorithme* 5) pour récupérer l'ensemble des motifs fréquents d'intérêt potentiel $MFIP$ et l'ensemble des motifs non fréquents d'intérêt potentiel $MNFIP$ (*ligne 2*). Le processus continue par l'extraction des règles positives. Pour tous les motifs M contenus dans l'ensemble $MFIP$ (*ligne 3*) et pour toutes les combinaisons $X \cup Y = M$ (*ligne 4*), les auteurs vérifient la fonction *fips* pour le motif XY.

Cette étape aurait été redondante si nous n'avions pas modifié la première phase, car tous les motifs contenus dans $MFIP$ vérifient déjà la contrainte *fips* puisque *fips* est une fonction appelée par *fip* lors de l'extraction des motifs fréquents d'intérêt potentiel. Si *fips* est vérifiée (*ligne 5*) alors les auteurs vérifient ensuite que les valeurs du facteur de certitude pour la règle $X \Rightarrow Y$ (*ligne 6*) et pour la règle $Y \Rightarrow X$ (*ligne 8*) soient

supérieures au seuil minimum de la confiance min_{conf}. Si la valeur du facteur de certitude vérifie la contrainte pour la règle $X \Rightarrow Y$ alors cette règle est ajoutée à l'ensemble R des règles valides (*ligne 7*). De plus, si la valeur du facteur de certitude vérifie la contrainte pour la règle $Y \Rightarrow X$ alors cette règle est également ajoutée à l'ensemble des règles valides (*ligne 9*). Nous constatons également un problème dans cette partie. En effet, la génération des règles $X \Rightarrow Y$ et $Y \Rightarrow X$ pour **toutes** les combinaisons de motifs $X \cup Y$ est redondante, puisqu'au moment où X et Y vont échanger leur valeur (*initialement si* $X = A$ *et* $Y = BC$, X *et* Y *auront échangé leur valeur lorsque* $X = BC$ *et* $Y = A$), les règles seront doublement générées. Il aurait donc été plus pertinent de parcourir toutes les combinaisons $X \cup Y$ mais uniquement pour la règle $X \Rightarrow Y$.

Le processus pour l'extraction des règles négatives est sensiblement similaire. En effet, pour tous les motifs M contenus dans l'ensemble $MNFIP$ (*ligne 10*) et pour toutes les combinaisons $X \cup Y = M$ (*ligne 11*), les auteurs vérifient la fonction *nfips* pour le motif XY. Ici encore, la vérification de la fonction *nfips* aurait été redondante si nous n'avions pas modifié la première phase, puisque lors de la construction de l'ensemble des motifs contenus dans $MNFIP$ nous utilisons la fonction *nfip* faisant elle-même appelle à *nfips*. Si *nfips* est vérifiée (*ligne 12*), alors les auteurs vérifient ensuite que la valeur du facteur de certitude pour l'ensemble des règles négatives est supérieure au seuil minimum de la confiance min_{conf} (*lignes 13 - 15 - 17 - 19 - 21 et 23*). Si la valeur du facteur de certitude pour les différentes règles dépasse le seuil, alors la règle concernée est ajoutée à l'ensemble R des règles valides (*lignes 14 - 16 - 18 - 20 - 22 et 24*). Le problème de redondance des règles est également présent dans ce processus puisque les auteurs parcourent **toutes** les combinaisons $X \cup Y$. De plus, il n'existe aucun moyen de savoir quel(s) est(sont) le(s) motif(s) négatif(s) dans la combinaison $X \cup Y = M$. Par conséquent, la fonction *nfips(X,Y)* et les différents calculs pour le facteur de certitude sont un peu ambigus pour le lecteur. Il aurait été plus clair de remplacer le parcours des motifs non fréquent d'intérêt potentiel (*lignes 10 à 24*) par ce que nous proposons dans l'*algorithme 7*.

Algorithme 7 : Proposition d'une version simplifiée pour la seconde partie de l'extraction des règles

```
1 pour tout motif M ∈ MNFIP faire
2     pour toute combinaison M1 ∪ M2 = M avec M1 ∈ {X, X̄}, M2 ∈ {Y, Ȳ} et
      X ∪ Y ≠ M faire
3         si nfips(M1, M2) alors
4             si fc(M2|M1) ⩾ min_conf alors
5                 R = R ∪ {M1 ⇒ M2}
```

Nous proposons également de parcourir l'ensemble des motifs non fréquents d'intérêt potentiel (*ligne 1*), mais en décomposant le motif M comme deux motifs M_1 et M_2 (*ligne 2*). Ainsi, il ne peut plus y avoir d'ambiguïté lors du calcul de la fonction *nfips* (*ligne 3*) et lors du calcul du facteur de certitude (*ligne 4*).

La dernière étape de leur méthode consiste à retourner l'ensemble des règles valides R (*ligne 25*).

2.2.3 Exemple

Nous allons maintenant dérouler l'algorithme de [Wu et al., 2004] sur l'exemple fil-rouge que nous avons déjà utilisé pour dérouler `Apriori`. Nous rappelons cet exemple dans le tableau 2.2 afin de simplifier la lecture. Dans un but de comparaison avec `Apriori`, nous reprenons les valeurs que nous avions pour les mesures du support et de la confiance. Nous allons donc prendre les paramètres suivants : 0,25 pour le support minimum, 0,80 pour la confiance minimum et 0,10 pour la mesure d'intérêt.

A	B	C	D	E
1	0	1	1	0
0	1	1	0	1
1	1	1	0	0
0	1	0	0	1

Tableau 2.2 – Exemple fil-rouge

1) Extraction des motifs

1-Motifs :

La première étape consiste à rechercher les motifs d'intérêt potentiel. Pour rechercher les motifs d'intérêt potentiel, la première étape consiste à récupérer les motifs fréquents de taille 1. Pour un support de 0,25, un motif est fréquent s'il apparaît au moins une fois. Les items A, B, C, D et E sont donc fréquents (*cf. tableau* 2.3). Nous retrouvons donc le tableau 1.3 puisque nous avons pris la même valeur pour le seuil min_{sup}.

Item	Support	Item	Support	Item	Support
A	0,50	B	0,75	C	0,75
D	0,25	E	0,50		

Tableau 2.3 – Items fréquents de taille 1 accompagnés de leur support

2-Motifs :

Les 1-motifs sont ensuite combinés afin de créer les 2-motifs et les supports pour ces différents motifs sont calculés. Le résultat est visible dans le tableau 2.4.

2-Motif	Support	2-Motif	Support	2-Motif	Support	2-Motif	Support
AB	0,25	AC	0,50	AD	0,25	AE	0
BC	0,50	BD	0	BE	0,50	CD	0,25
CE	0,25	DE	0				

Tableau 2.4 – Motifs de taille 2 accompagnés de leur support

Les motifs sont ensuite ajoutés dans deux ensembles en fonction de leur support. F contient les motifs fréquents tandis que NF contient les motifs non fréquents :
— $F = \{AB, AC, AD, BC, BE, CD, CE\}$.
— $NF = \{AE, BD, DE\}$.

3-Motifs :

Les motifs fréquents de taille 2 sont par la suite réutilisés afin de générer les candidats de taille 3. La génération du prochain niveau s'obtient en associant tout simplement deux k-motifs pour obtenir un *(k+1)*-motif. Par exemple, si on combine AB et AC, le motif ABC est créé même si le motif BC n'est pas fréquent. Par contre, comme pour `Apriori`, si on combine AB et CD, aucun motif n'est créé car le motif obtenu $ABCD$ n'a pas la bonne taille. On obtient donc le tableau 2.5 pour le prochain niveau.

3-Motif	Support	3-Motif	Support	3-Motif	Support	3-Motif	Support
ABC	0,25	ABD	0	ABE	0	ACD	0,25
ACE	0	BCE	0,25	BCD	0	CDE	0

Tableau 2.5 – Motifs de taille 3 accompagnés de leur support

On ajoute les motifs de taille 3 aux motifs fréquents et non fréquents. Les ensembles F et NF sont donc les suivants :
— $F = \{AB, AC, AD, BC, BE, CD, CE, ABC, ACD, BCE\}$.
— $NF = \{AE, BD, DE, ABD, ABE, ACE, BCD, CDE\}$.

4-Motifs :

L'ensemble des motifs fréquents de taille 3 n'étant toujours pas vide, nous pouvons continuer la génération des candidats. Les motifs de taille 4 sont donnés dans le tableau 2.6.

4-Motif	Support	4-Motif	Support
$ABCE$	0	$ABCD$	0

Tableau 2.6 – Motifs de taille 4 accompagnés de leur support

Comme précédemment, on ajoute les motifs de taille 4 aux motifs fréquents et non fréquents, et on obtient les ensembles suivants :
— $F = \{AB, AC, AD, BC, BE, CD, CE, ABC, ACD, BCE\}$.
— $NF = \{AE, BD, DE, ABD, ABE, ACE, BCD, CDE, ABCE, ABCD\}$.

Aucun motif fréquent de taille 4 n'a pu être généré. Par conséquent, la génération des motifs s'arrête là.

2) Motifs d'intérêt potentiel

Motifs fréquents d'intérêt potentiel :

Nous venons de trouver l'ensemble des motifs, passons maintenant à la recherche des règles. Les motifs fréquents doivent être des motifs d'intérêt potentiel et par conséquent respecter la mesure fip. Les deux premières conditions de la mesure fip sont déjà respectées pour l'ensemble des motifs de F puisque cet ensemble comporte uniquement des motifs fréquents et il existe bien X et Y tel que $XY \in F$. Il suffit donc de vérifier la dernière condition : $fips(X, Y)$ pour chaque combinaison XY.

Nous allons maintenant expliquer cette méthode sur le motif AC. Il faut donc vérifier la combinaison AC et la combinaison CA. Commençons par étudier la combinaison AC :

$$fips(A, C) = (A \cap C = \varnothing) \wedge (f(A, C, min_{sup}, min_{conf}, min_{int}) = 1)$$

avec la fonction f valant :

$$f(A, C, 0,25, 0,80, 0,10) = \frac{0{,}5+10+0{,}125-(0{,}25+0{,}80+0{,}1)+1}{|0{,}50-0{,}25|+|1-0{,}80|+|0{,}125-0{,}1|+1} = \frac{1{,}625-1{,}15+1}{|0{,}25|+|0{,}20|+|0{,}025|+1}$$

$$= \frac{1{,}475}{1{,}475} = 1$$

puisque $conf(A \Rightarrow C) = \frac{sup(AC)}{sup(A)} = \frac{0{,}50}{0{,}50} = 1$ et
$int(A, C) = |sup(AC) - sup(A) \times sup(C)| = 0,50 - 0,50 \times 0,75 = 0,125.$

La combinaison AC est donc bien une combinaison fréquente d'intérêt potentiel puisque la fonction $f(A, C, min_{sup}, min_{conf}, min_{int})$ vaut 1 et par conséquent $fips(A, C)$ est aussi égale à 1. Passons maintenant à la combinaison CA :

$$fips(C, A) = (C \cap A = \varnothing) \wedge (f(C, A, min_{sup}, min_{conf}, min_{int}) = 1)$$

avec la fonction f valant :

$$f(C, A, 0,25, 0,80, 0,10) \simeq \frac{0{,}50+0{,}66+0{,}125-(0{,}25+0{,}80+0{,}1)+1}{|0{,}50-0{,}25|+|0{,}66-0{,}80|+|0{,}125-0{,}10|+1} \simeq \frac{1{,}285-1{,}15+1}{|0{,}25|+|-0{,}14|+|0{,}025|+1}$$

$$\simeq \frac{1{,}135}{1{,}455} \simeq 0,80$$

puisque $conf(C \Rightarrow A) = \frac{sup(AC)}{sup(C)} = \frac{0{,}50}{0{,}75} \simeq 0.66$ et
$int(A, C) = |sup(AC) - sup(A) \times sup(C)| = 0,50 - 0,50 \times 0,75 = 0,125.$

La fonction $f(C, A, min_{sup}, min_{conf}, min_{int})$ ne vaut pas 1 et par conséquent $fips(C, A)$ n'est pas égale à 1. La combinaison CA n'est donc pas d'intérêt potentiel.

Pour les autres motifs fréquents, le processus est identique. Pour un motif de taille k il faut donc tester $2^k - 2$ combinaisons. Par exemple pour le motif fréquent ACD il faut tester six combinaisons et comme nous l'indiquons dans le tableau 2.7 seules les combinaisons $CD \cup A$ et $D \cup AC$ sont valides (*ici on garde la version non compactée pour éviter les ambiguïtés*). Les combinaisons fréquentes d'intérêt potentiel obtenues sont

renseignées dans le tableau 2.7.

Prémisse	Conclusion	Support	Confiance	Intérêt
A	C	0,50	1	0,125
CD	A	0,25	1	0,125
D	A	0,25	1	0,125
D	AC	0,25	1	0,125
E	B	0,50	1	0,125

TABLEAU 2.7 – Combinaisons fréquentes d'intérêt potentiel

Motifs non fréquents d'intérêt potentiel :

Le processus est sensiblement identique pour les motifs non fréquents. Les motifs non fréquents doivent être des motifs d'intérêt potentiel et donc respecter la mesure $nfip$. Les deux premières conditions de la mesure $nfip$ sont déjà respectées pour l'ensemble des motifs de NF puisque cet ensemble comporte uniquement des motifs non fréquents et il existe bien M_1 et M_2 tel que $M_1M_2 \in NF$. Il suffit donc de vérifier la dernière condition : $nfips(M_1, M_2)$ pour chaque combinaison M_1M_2.

Nous allons maintenant expliquer cette méthode sur le motif AE. Pour le motif non fréquent AE, il faut vérifier six combinaisons. En effet, les combinaisons $\overline{A}E$, $A\overline{E}$, $\overline{A}\,\overline{E}$, $\overline{E}A$, $E\overline{A}$ et $\overline{E}\,\overline{A}$ sont à étudier. Le processus étant assez répétitif, nous allons uniquement dérouler l'algorithme sur la combinaison $\overline{A}E$:

$$nfips(\overline{A}, E) = (\overline{A} \cap E = \varnothing) \wedge (g(\overline{A}, E, min_{sup}, min_{conf}, min_{int}) = 2)$$

avec la fonction g valant :

$$g(\overline{A}, E, 0,25, 0,80, 0, 1) = f(\overline{A}, E, 0,25, 0,80, 0, 1) + \frac{sup(\overline{A})+sup(E)-2\times 0,25+1}{|sup(\overline{A})-0,25|+|sup(E)-0,25|+1}$$

$$= \frac{0,50+1+0,25-(0,25+0,80+0,1)+1}{|0,50-0,25|+|1-0,80|+|0,25-0,10|+1} + \frac{0,50+0,50-2\times 0,25+1}{|0,50-0,25|+|0,50-0,25|+1}$$

$$= \frac{1,6}{1,6} + \frac{1,5}{1,5} = 1 + 1 = 2$$

Comme la fonction $g(\overline{A}, E, 0,25, 0,80, 0, 10)$ retourne 2, la combinaison $\overline{A}E$ est donc d'intérêt potentiel.

Pour les autres motifs et les autres combinaisons, la méthode de calcul reste la même. Les combinaisons non fréquentes d'intérêt potentiel sont synthétisées dans le tableau 2.8.

Prémisse	Conclusion	Support	Confiance	Intérêt	Sup(Prémisse)	Sup(Conclusion)
$\overline{A}$	BE	0,50	1	0,25	0,50	0,50
$\overline{A}$	E	0,50	1	0,25	0,50	0,50
$\overline{AC}$	BE	0,50	1	0,25	0,50	0,50
$\overline{AC}$	E	0,50	1	0,25	0,50	0,50
$\overline{ACD}$	B	0,75	1	0,1875	0,75	0,75
$\overline{AD}$	B	0,75	1	0,1875	0,75	0,75
$\overline{B}$	ACD	0,25	1	0,1875	0,25	0,25
$\overline{B}$	AD	0,25	1	0,1875	0,25	0,25
$\overline{B}$	CD	0,25	1	0,1875	0,25	0,25
$\overline{B}$	D	0,25	1	0,1875	0,25	0,25
$\overline{BE}$	A	0,50	1	0,25	0,50	0,50
$\overline{BE}$	AC	0,50	1	0,25	0,50	0,50
$\overline{CD}$	B	0,75	1	0,1875	0,75	0,75
$\overline{D}$	B	0,75	1	0,1875	0,75	0,75
$\overline{E}$	A	0,50	1	0,25	0,50	0,50
$\overline{E}$	AC	0,50	1	0,25	0,50	0,50
A	$\overline{BCE}$	0,50	1	0,125	0,50	0,75
A	$\overline{BE}$	0,50	1	0,25	0,50	0,50
A	$\overline{CE}$	0,50	1	0,125	0,50	0,75
A	$\overline{E}$	0,50	1	0,25	0,50	0,50
AB	$\overline{E}$	0,25	1	0,125	0,25	0,50
ABC	$\overline{E}$	0,25	1	0,125	0,25	0,50
AC	$\overline{BE}$	0,50	1	0,25	0,50	0,50
AC	$\overline{E}$	0,50	1	0,25	0,50	0,50
ACD	$\overline{B}$	0,25	1	0,1875	0,25	0,25
AD	$\overline{B}$	0,25	1	0,1875	0,25	0,25
AD	$\overline{BC}$	0,20	1	0,125	0,25	0,50
B	$\overline{ACD}$	0,75	1	0,1875	0,75	0,75
B	$\overline{AD}$	0,75	1	0,1875	0,75	0,75
B	$\overline{CD}$	0,75	1	0,1875	0,75	0,75
B	$\overline{D}$	0,75	1	0,1875	0,75	0,75
BC	$\overline{AD}$	0,50	1	0,125	0,50	0,75
BC	$\overline{D}$	0,50	1	0,125	0,50	0,75
BCE	$\overline{A}$	0,25	1	0,125	0,25	0,50
BE	$\overline{A}$	0,50	1	0,25	0,50	0,50
BE	$\overline{AC}$	0,50	1	0,25	0,50	0,50
CD	$\overline{B}$	0,25	1	0,1875	0,25	0,25
CD	$\overline{E}$	0,25	1	0,125	0,25	0,50
CE	$\overline{A}$	0,25	1	0,125	0,25	0,50
D	$\overline{B}$	0,25	1	0,1875	0,25	0,25
D	$\overline{BC}$	0,25	1	0,125	0,25	0,50
D	$\overline{E}$	0,25	1	0,125	0,25	0,50
E	$\overline{A}$	0,50	1	0,25	0,50	0,50
E	$\overline{AB}$	0,50	1	0,125	0,50	0,75
E	$\overline{ABC}$	0,50	1	0,125	0,50	0,75
E	$\overline{AC}$	0,50	1	0,25	0,50	0,50
E	$\overline{CD}$	0,50	1	0,125	0,50	0,75
E	$\overline{D}$	0,50	1	0,125	0,50	0,75

TABLEAU 2.8 – Combinaisons non fréquentes d'intérêt potentiel

Ce tableau contient 48 combinaisons non fréquentes d'intérêt potentiel : 16 combinaisons pouvant mener à des règles du type $\overline{X} \Rightarrow Y$ et 32 combinaisons pouvant mener à des règles du type $X \Rightarrow \overline{Y}$.

3) Extraction des règles

La dernière étape avant de générer les règles est de vérifier pour chaque combinaison la valeur du facteur de certitude. Le facteur de certitude est défini en fonction de la zone dans laquelle la règle se situe. Si la règle $X \Rightarrow Y$ est dans la zone attractive, c'est-à-dire si $conf(X \Rightarrow Y) > sup(Y)$ alors le facteur de certitude est défini comme $fc(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - sup(Y)}{1 - sup(Y)}$. Sinon, si la règle $X \Rightarrow Y$ est dans la zone répulsive, c'est-à-dire si $conf(X \Rightarrow Y) < sup(Y)$ alors le facteur de certitude est défini comme $fc(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - sup(Y)}{sup(Y)}$. Dans cet exemple, au vu de la valeur de la confiance, toutes les combinaisons se situent dans la zone attractive. Calculons maintenant le facteur de certitude pour deux combinaisons : AC qui est une combinaison fréquente d'intérêt potentiel et $\overline{AD}B$ qui est une combinaison non fréquente d'intérêt potentiel.

Pour AC, comme la confiance de la règle $A \Rightarrow C$ est supérieure au support de C (*puisque* $1 > 0,75$) le facteur de certitude se calcule avec la formule :

$$fc(A \Rightarrow C) = \frac{conf(A \Rightarrow C) - sup(C)}{1 - sup(C)} = \frac{1 - 0,75}{1 - 0,75} = 1$$

Comme la valeur du facteur de certitude est supérieure au seuil de la confiance $(0,80)$, la règle $A \Rightarrow C$ va être extraite.

Passons maintenant à la combinaison $\overline{AD}B$. Ici encore, la confiance de la règle $\overline{AD} \Rightarrow B$ est supérieure au support de B (*puisque* $1 > 0,75$). Le facteur de certitude se calcule donc avec la même formule :

$$fc(\overline{AD} \Rightarrow B) = \frac{conf(\overline{AD} \Rightarrow B) - sup(B)}{1 - sup(B)} = \frac{1 - 0,75}{1 - 0,75} = 1$$

Le facteur de certitude est également supérieur au seuil de la confiance pour la règle $\overline{AD} \Rightarrow B$, par conséquent cette règle sera également extraite. La procédure est la même pour l'ensemble des combinaisons. Nous remarquons que les 21 combinaisons non fréquentes d'intérêt potentiel vont être validées après la vérification du facteur de certitude.

Le tableau 2.9 récapitule l'ensemble des règles extraites.

Règle	Facteur de certitude	Règle	Facteur de certitude
$A \Rightarrow C$	1	$CD \Rightarrow A$	1
$D \Rightarrow A$	1	$D \Rightarrow AC$	1
$E \Rightarrow B$	1		
$\overline{A} \Rightarrow BE$	1	$\overline{A} \Rightarrow E$	1
$\overline{AC} \Rightarrow BE$	1	$\overline{AC} \Rightarrow E$	1
$\overline{ACD} \Rightarrow B$	1	$\overline{AD} \Rightarrow B$	1
$\overline{B} \Rightarrow ACD$	1	$\overline{B} \Rightarrow AD$	1
$\overline{B} \Rightarrow CD$	1	$\overline{B} \Rightarrow D$	1
$\overline{BE} \Rightarrow A$	1	$\overline{BE} \Rightarrow AC$	1
$\overline{CD} \Rightarrow B$	1	$\overline{D} \Rightarrow B$	1
$\overline{E} \Rightarrow A$	1	$\overline{E} \Rightarrow AC$	1
$A \Rightarrow \overline{BCE}$	1	$A \Rightarrow \overline{BE}$	1
$A \Rightarrow \overline{CE}$	1	$A \Rightarrow \overline{E}$	1
$AB \Rightarrow \overline{E}$	1	$ABC \Rightarrow \overline{E}$	1
$AC \Rightarrow \overline{BE}$	1	$AC \Rightarrow \overline{E}$	1
$ACD \Rightarrow \overline{B}$	1	$AD \Rightarrow \overline{B}$	1
$AD \Rightarrow \overline{BC}$	1	$B \Rightarrow \overline{ACD}$	1
$B \Rightarrow \overline{AD}$	1	$B \Rightarrow \overline{CD}$	1
$B \Rightarrow \overline{D}$	1	$BC \Rightarrow \overline{AD}$	1
$BC \Rightarrow \overline{D}$	1	$BCE \Rightarrow \overline{A}$	1
$BE \Rightarrow \overline{A}$	1	$BE \Rightarrow \overline{AC}$	1
$CD \Rightarrow \overline{B}$	1	$CD \Rightarrow \overline{E}$	1
$CE \Rightarrow \overline{A}$	1	$D \Rightarrow \overline{B}$	1
$D \Rightarrow \overline{BC}$	1	$D \Rightarrow \overline{E}$	1
$E \Rightarrow \overline{A}$	1	$E \Rightarrow \overline{AB}$	1
$E \Rightarrow \overline{ABC}$	1	$E \Rightarrow \overline{AC}$	1
$E \Rightarrow \overline{CD}$	1	$E \Rightarrow \overline{D}$	1

TABLEAU 2.9 – Règles extraites sur la base d'exemple classées par type de règles

En conclusion, l'algorithme de [Wu et al., 2004] génère 53 règles au total : 5 règles positives $X \Rightarrow Y$, 16 règles négatives du type $\overline{X} \Rightarrow Y$, 32 règles négatives du type $X \Rightarrow \overline{Y}$ et 0 règle négative du type $\overline{X} \Rightarrow \overline{Y}$. Une analyse comparative sur les règles extraites de cet algorithme avec `Apriori` et les autres algorithmes que nous allons voir par la suite est disponible dans le *chapitre* 6.

2.3 Algorithme proposé par Antonie et Zaïane

La seconde approche étudiée est celle de [Antonie and Zaïane, 2004]. Nous commençons par présenter les critères de validité d'une règle positive et négative. Nous expliquons ensuite leur méthode d'extraction pour les règles d'association. Cette méthode se compose d'un seul algorithme qui se divise en deux parties. La première partie consiste à rechercher les motifs candidats XY où X et Y sont fortement corrélés. La seconde partie concerne la génération des règles et s'opère à partir des motifs candidats préalablement extraits. Un petit exemple est ensuite utilisé afin de dérouler l'algorithme.

2.3.1 Critères de validité d'une règle

[Antonie and Zaïane, 2004] considèrent qu'une règle ne peut être valide que si sa prémisse et sa conclusion sont fortement corrélées. Ils proposent donc d'ajouter le coefficient de corrélation ρ [Pearson, 1896] à l'approche support/confiance. La formule du coefficient de corrélation est la suivante :

$$\rho(X,Y) = \frac{sup(XY)\times sup(\overline{X}\,\overline{Y}) - sup(\overline{X}Y)\times sup(X\overline{Y})}{\sqrt{sup(X)\times sup(Y)\times sup(\overline{X})\times sup(\overline{Y})}}$$

Un seuil minimum de corrélation min_ρ va donc être introduit afin de déterminer les motifs relativement bien corrélés. Pour fixer ce seuil de corrélation, les auteurs suivent la théorie de [Cohen, 1988] qui met en place un « barème » afin d'interpréter l'intensité ou la force du coefficient de corrélation : ainsi un seuil de 0,5 implique une grande corrélation, de 0,3 une corrélation raisonnable et de 0,1 une petite corrélation. Les auteurs vont donc fixer par défaut leur seuil à 0,5, mais si le seuil est trop grand et qu'ils n'obtiennent pas de règles, alors ils vont réitérer leur algorithme en diminuant progressivement ce seuil jusqu'à obtenir un ensemble de règles non vide. Les utilisateurs sont libres de changer ce seuil en fonction des données à analyser.

Le coefficient de corrélation va leur permettre, d'une part, d'éliminer les motifs faiblement corrélés, et d'autre part, de déterminer les règles à étudier. Les auteurs vont calculer la corrélation pour chaque combinaison XY. En fonction de la valeur de la corrélation, les règles étudiées ne seront pas les mêmes :

- si $-min_\rho < \rho(X,Y) < min_\rho$ alors le motif XY ne servira pas à générer de règles. En effet, même si nous pouvons dire quelque chose sur la liaison qui existe entre les motifs X et Y (*lien positif ou négatif*), ce lien n'est pas assez fort pour être intéressant.
- si $\rho(X,Y) \geqslant min_\rho$ alors les motifs X et Y sont corrélés positivement et cette corrélation est jugée significative. Dans ce cas, les auteurs vont étudier les règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$. La valeur du support de XY va ensuite déterminer quelle règle étudier. Pour les deux règles, la confiance sera vérifiée. Si les auteurs étudient la règle $\overline{X} \Rightarrow \overline{Y}$ alors son support sera également vérifié.
- si $\rho(X,Y) \leqslant -min_\rho$ alors les motifs X et Y sont corrélés négativement et cette corrélation est jugée significative.. Ils vont générer les règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ et calculer leur confiance. Mais ils ne vérifient à aucun moment le support du motif générant la règle. Nous corrigerons l'algorithme en vérifiant les supports des motifs générés.

Le tableau 2.10 récapitule les contraintes pour qu'une règle soit considérée valide selon leurs critères. Nous renseignons également les contraintes ajoutées à notre implémentation.

$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$
$\rho(X,Y) \geqslant min_\rho$ $sup(XY) \geqslant min_{sup}$ $conf(X \Rightarrow Y) \geqslant min_{conf}$	$\rho(X,Y) \leqslant -min_\rho$ ajout : $sup(\overline{X}Y) \geqslant min_{sup}$ $conf(\overline{X} \Rightarrow Y) \geqslant min_{conf}$
$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$
$\rho(X,Y) \leqslant -min_\rho$ ajout : $sup(X\overline{Y}) \geqslant min_{sup}$ $conf(X \Rightarrow \overline{Y}) \geqslant min_{conf}$	$\rho(X,Y) \geqslant min_\rho$ $sup(XY) < min_{sup}$ $conf(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{conf}$ $sup(\overline{X}\,\overline{Y}) \geqslant min_{sup}$

TABLEAU 2.10 – Règles valides

En conclusion, l'ensemble des règles $X \Rightarrow Y$, $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$ doivent respecter les différents seuils des mesures suivantes : support, confiance et coefficient de corrélation.

Nous allons maintenant présenter la méthode utilisée par [Antonie and Zaïane, 2004] pour extraire les règles d'association positives et négatives valides.

2.3.2 Méthode d'extraction

La méthode proposée par [Antonie and Zaïane, 2004] est présentée dans l'*algorithme 8*.

Si l'utilisateur ne fournit pas de seuil pour la corrélation, l'algorithme utilise la valeur par défaut (*lignes 1 et 2*). Le processus suivant (*lignes 3 à 28*) se répète tant que l'ensemble R des règles valides extraites est vide (*ligne 28*). Le processus commence par initialiser l'ensemble R des règles valides à l'ensemble vide (*ligne 4*). L'algorithme va ensuite chercher les items fréquents et les stocker dans l'ensemble F_1 (*ligne 5*). Cet ensemble sera utilisé par la suite pour générer les motifs candidats. Le processus suivant (*lignes 6 à 24*) va être réitéré jusqu'à ce que l'ensemble F_{k-1} des motifs fréquents soit vide. La construction des candidats est différente par rapport à celle d'`Apriori`. En effet, les candidats C_k de taille k sont construits en utilisant le produit cartésien entre l'ensemble F_{k-1} des *(k-1)*-motifs fréquents de niveau inférieur et l'ensemble F_1 des items fréquents (*ligne 7*). Ceci permet de générer des motifs XY non fréquents mais où la corrélation entre X et Y est assez forte et où le motif $X\overline{Y}$ ou $\overline{X}Y$ peut être fréquent. Ensuite, pour tous les motifs M de C_k (*ligne 8*), les auteurs calculent le support (*ligne 9*). Si le support du motif M est valide (*ligne 10*), alors M est ajouté à l'ensemble F_k des motifs fréquents qui va ensuite être utilisé pour générer les motifs candidats à la prochaine itération (*ligne 7*). Après, pour toutes les combinaisons $XY = M$ (*ligne 12*), les auteurs vont calculer le coefficient de corrélation de Pearson (*ligne 13*) et le comparer au seuil de corrélation minimum min_ρ.

Si le coefficient de corrélation est supérieur ou égal à min_ρ (*ligne 14*), ils vont vérifier le support du motif XY. Si XY est fréquent (*ligne 15*), alors la confiance de la règle $X \Rightarrow Y$ est analysée (*ligne 16*), et si la règle est valide pour cette mesure alors elle est

Algorithme 8 : Extraction des règles d'association positives et négatives

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, confiance minimum min_{conf}, corrélation minimum min_ρ
Sortie : ensemble des règles valides R

si min_ρ *est non défini* **alors**
  $min_\rho = 0,5$
faire
  $R = \varnothing$
  $F_1 = \{i \in \mathcal{I}$ tel que $sup(i) \geqslant min_{sup}\}$
  pour $(k = 2; F_{k-1} \neq \varnothing; k++)$ **faire**
    $C_k = F_{k-1} \bowtie F_1$;
    pour tout *motif* $M \in C_k$ **faire**
      $s = support(\mathcal{D}, M)$;
      si $s \geq min_{supp}$ **alors**
        $F_k = F_k \cup \{M\}$;
      pour toute *combinaison* $XY = M$ **faire**
        $\rho = \rho(X, Y)$;
        si $\rho \geqslant min_\rho$ **alors**
          si $sup(XY) \geqslant min_{supp}$ **alors**
            si $conf(X \Rightarrow Y) \geqslant min_{conf}$ **alors**
              $R = R \cup \{X \Rightarrow Y\}$;
          sinon si $conf(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{conf}$ **et** $sup(\overline{X}\,\overline{Y}) \geqslant min_{supp}$ **alors**
            $R = R \cup \{\overline{X} \Rightarrow \overline{Y}\}$;
        sinon si $\rho \leqslant -min_\rho$ **alors**
          si $conf(\overline{X} \Rightarrow Y) \geqslant min_{conf}$ **alors**
            $R = R \cup \{\overline{X} \Rightarrow Y\}$;
          si $conf(X \Rightarrow \overline{Y}) \geqslant min_{conf}$ **alors**
            $R = R \cup \{X \Rightarrow \overline{Y}\}$;
  $min_\rho = min_\rho - 0,1$
  si $min_\rho < 0$ **alors**
    Quitter le programme ;
tant que $R = \varnothing$;
retourner R

ajoutée à l'ensemble R des règles valides (*ligne 17*). Si le motif XY n'est pas fréquent, les auteurs vont analyser la confiance de la règle $\overline{X} \Rightarrow \overline{Y}$ et le support du motif $\overline{X}\,\overline{Y}$ (*ligne 18*). Si les valeurs de ces deux mesures pour la règle $\overline{X} \Rightarrow \overline{Y}$ sont valides, alors la règle est ajoutée à l'ensemble R (*ligne 19*).

Si le coefficient de corrélation est inférieur ou égal à $-min_\rho$ (*ligne 20*), ils étudient les règles mixtes $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ en analysant uniquement la confiance (*lignes 21 et 23*) mais ils oublient de vérifier le support. Si les valeurs des confiances des règles sont valides alors elles sont ajoutées à l'ensemble R (*lignes 22 et 24*).

La partie suivante de l'algorithme (*lignes 25 à 27*) permet de réitérer l'algorithme automatiquement si aucune règle n'a pu être extraite lors de la première itération. En effet, si l'ensemble de sortie des règles est vide (*ligne 28*), alors le seuil de corrélation min_ρ est diminué de $0,1$ (*ligne 25*) et le processus reprend (*ligne 4*). Dans la pratique, la configuration automatique de ce paramètre facilite son utilisation mais risque d'augmenter les temps de calculs. En effet, il faut calculer le coefficient de corrélation pour l'ensemble des combinaisons possibles (*lignes 12 et 13*) puis vérifier si des règles peuvent être obtenues à chaque fois que l'on décrémente ce seuil. La dernière étape de l'algorithme (*ligne 29*) consiste à retourner l'ensemble des règles d'association positives et négatives extraites par le processus.

Le parcours de toutes les combinaisons $XY = M$ va engendrer de la redondance. En effet, le coefficient de corrélation est une mesure symétrique. Par conséquent, on a $\rho(X,Y) = \rho(Y,X)$. Ainsi la moitié des calculs du coefficient de corrélation vont être inutiles puisque ils ont déjà été effectués. Il en est de même pour le support du motif YX qu'ils sont obligés de revérifier (*ligne 15*) alors que la propriété $sup(XY) = sup(YX)$ existe. Il aurait été plus judicieux de traiter directement les règles symétriques : et par conséquent de calculer la confiance des règles $X \Rightarrow Y$, $Y \Rightarrow X$, $\overline{X} \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow \overline{X}$ après avoir vérifié le support dans le cas où $\rho \geqslant min_\rho$ et la confiance des règles $\overline{X} \Rightarrow Y$, $Y \Rightarrow \overline{X}$, $X \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow X$ dans le cas où $\rho \leqslant -min_\rho$. Par ailleurs, tout comme [Wu et al., 2004], les auteurs n'utilisent pas la propriété de la confiance alors qu'elle aurait pu être utilisée pour les règles positives.

2.3.3 Exemple

Nous allons maintenant dérouler l'algorithme de [Antonie and Zaïane, 2004] sur notre exemple fil-rouge que nous redonnons dans le *tableau* 2.11. Nous prenons les mêmes valeurs pour le support et la confiance que celles utilisées dans les exemples précédents. Les paramètres sont donc les suivants : 0,25 pour le support minimum, 0,80 pour la confiance minimum et 0,50 pour le coefficient de corrélation.

A	B	C	D	E
1	0	1	1	0
0	1	1	0	1
1	1	1	0	0
0	1	0	0	1

Tableau 2.11 – Exemple fil-rouge

Nous allons dérouler la version corrigée de l'algorithme afin que l'ensemble des règles respecte le support minimum. Nous mesurons également l'impact de cette modification sur les résultats dans la conclusion de cet exemple.

1) Extraction des motifs fréquents

1-Motifs fréquents :

La première étape consiste à rechercher les motifs fréquents de taille 1. Pour un support de 0,25, les motifs A, B, C, D et E sont fréquents (*cf. tableau* 2.12). Nous retrouvons le tableau 1.3 puisque nous avons pris la même valeur pour le seuil min_{sup}. Par la suite, les autres tableaux contenant les motifs fréquents seront également similaires à ceux d'`Apriori`.

Item	Support	Item	Support	Item	Support
A	0,50	B	0,75	C	0,75
D	0,25	E	0,50		

Tableau 2.12 – Items fréquents de taille 1 accompagnés de leur support

2-Motifs candidats :

Les candidats C_k de taille k sont construits en utilisant le produit cartésien entre l'ensemble des *(k-1)*-motifs fréquents de niveau inférieur et l'ensemble F_1 des items fréquents. Les candidats de taille 2 sont visibles dans le tableau 2.13. Les candidats de taille 2 sont les mêmes que ceux pour `Apriori` (*cf. tableau* 1.4) puisque pour $k = 2$, $F_{k-1} \bowtie F_1$ revient à la même chose que $F_{k-1} \bowtie F_{k-1}$.

2-Motif					
AB	AC	AD	AE	BC	BD
BE	CD	CE	DE		

Tableau 2.13 – Motifs candidats de taille 2

2-Motifs fréquents :

Le support est ensuite calculé pour chaque motif candidat puis comparé au support minimum. Le tableau 2.14 restitue les motifs fréquents parmi les candidats du tableau 2.13.

2-Motif	Support	2-Motif	Support	2-Motif	Support
AB	0,25	AC	0,50	AD	0,25
BC	0,50	BE	0,50	CD	0,25
CE	0,25				

TABLEAU 2.14 – Motifs fréquents de taille 2 accompagnés de leur support

3-Motifs candidats :

Les motifs candidats de taille 3 sont ensuite créés à partir de la combinaison des motifs fréquents de taille 2 et les motifs fréquents de taille 1. Dix 3-motifs candidats sont donc générés (*cf. tableau* 2.15). Ces candidats de taille 3 ne sont plus les mêmes que ceux trouvés par `Apriori` (*cf. tableau* 1.7) puisque pour $k > 2$, $F_{k-1} \bowtie F_1$ n'est plus similaire à $F_{k-1} \bowtie F_{k-1}$.

3-Motif					
ABC	ABD	ABE	ACD	ACE	ADE
BCD	BCE	BDE	CDE		

TABLEAU 2.15 – Motifs candidats de taille 3

3-Motifs fréquents :

Les fréquents de taille 3 sont ensuite déterminés (*cf. tableau* 2.16).

3-Motif	Support	3-Motif	Support	3-Motif	Support
ABC	0,25	ACD	0,25	BCE	0,25

TABLEAU 2.16 – Motifs fréquents de taille 3 accompagnés de leur support

4-Motifs candidats :

L'ensemble des fréquents n'étant toujours pas vide, la génération des candidats se poursuit. Le prochain niveau de candidats est répertorié dans le tableau 2.17.

4-Motif			
$ABCD$	$ABCE$	$ACDE$	$BCDE$

TABLEAU 2.17 – Motifs candidats de taille 4

4-Motifs fréquents :

Nous déterminons les fréquents de taille 4 : aucun motif candidat présent dans le tableau 2.17 n'est fréquent. L'ensemble des fréquents de taille 4 étant vide, la recherche

des motifs candidats s'arrête à cette étape.

2) Corrélation des motifs

La prochaine étape consiste à calculer le coefficient de corrélation pour toutes les combinaisons possibles $XY = M$ avec $M \in C_k$. Calculons la corrélation pour les différentes combinaisons composant le motif candidat BDE. Le coefficient de corrélation étant une mesure symétrique, nous allons le calculer, dans cet exemple, uniquement pour les combinaisons $DE \cup B$, $BE \cup D$ et $BD \cup E$. Nous réintroduisons ici la notation initiale pour mettre en évidence les combinaisons étudiées. Cependant l'algorithme original calcule également le coefficient de corrélation pour les combinaisons $B \cup DE$, $D \cup BE$ et $E \cup BD$.

$$\rho(DE, B) = \frac{sup(BDE) \times sup(\overline{DE}\,\overline{B}) - sup(\overline{DE}B) \times sup(DE\overline{B})}{\sqrt{sup(DE) \times sup(B) \times sup(\overline{DE}) \times sup(\overline{B})}} = \frac{0 \times 0{,}25 - 0{,}75 \times 0}{\sqrt{0 \times 0{,}75 \times 1 \times 0{,}25}} = 0$$

$$\rho(BE, D) = \frac{sup(BDE) \times sup(\overline{BE}\,\overline{D}) - sup(\overline{BE}D) \times sup(BE\overline{D})}{\sqrt{sup(BE) \times sup(D) \times sup(\overline{BE}) \times sup(\overline{D})}} = \frac{0 \times 0{,}25 - 0{,}25 \times 0{,}50}{\sqrt{0{,}50 \times 0{,}25 \times 0{,}50 \times 0{,}75}}$$

$$\simeq \frac{-0{,}125}{0{,}217} \simeq -0,58$$

$$\rho(BD, E) = \frac{sup(BDE) \times sup(\overline{BD}\,\overline{E}) - sup(\overline{BD}E) \times sup(BD\overline{E})}{\sqrt{sup(BD) \times sup(E) \times sup(\overline{BD}) \times sup(\overline{E})}} = \frac{0 \times 0{,}50 - 0{,}50 \times 0}{\sqrt{0 \times 0{,}50 \times 1 \times 0{,}50}} = 0$$

Motif 1	Motif 2	Corrélation	Motif 1	Motif 2	Corrélation
A	B	-0,58	A	BCE	-0,58
A	BE	-1	A	C	0,58
A	CD	0,58	A	CE	-0,58
A	D	0,58	A	E	-1
AB	E	-0,58	ABC	E	-0,58
AC	B	-0,58	AC	BE	-1
AC	D	0,58	AC	E	-1
ACD	B	-1	ACD	E	-0,58
AD	B	-1	AD	BC	-0,58
AD	E	-0,58	B	CD	-1
B	D	-1	B	E	0,58
BC	D	-0,58	BE	C	-0,58
BE	CD	-0,58	BE	D	-0,58
C	E	-0,58	CD	E	-0,58
D	E	-0,58			

Tableau 2.18 – Motifs suffisamment corrélés

Après avoir calculé le coefficient de corrélation pour ces trois combinaisons, seule la combinaison $BE \cup D$ est valide au regard de la corrélation. Le coefficient de corrélation

étant une mesure symétrique, la combinaison $D \cup BE$ est également valide. Le tableau 2.18 énumère les différentes combinaisons valides ainsi que leur valeur pour le coefficient de corrélation. La combinaison XY ayant la même corrélation que la combinaison YX, le tableau ne renseigne qu'une seule des deux combinaisons.

3) Génération des règles valides

Les combinaisons corrélées positivement vont permettre de générer les règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ tandis que les combinaisons corrélées négativement vont permettre de générer les règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. Nous allons maintenant rechercher les règles valides pour les combinaisons AC et $AC \cup B$. Commençons par la combinaison AC.

La corrélation pour la combinaison AC est égale à 0,58 et donc bien supérieure ou égale à la corrélation minimum fournie par l'utilisateur $(0, 50)$. Par conséquent nous allons étudier les règles $A \Rightarrow C$ et $\overline{A} \Rightarrow \overline{C}$. La première vérification concerne le support du motif AC. AC est un motif fréquent puisqu'il possède un support de 0,50. Nous allons donc étudier uniquement la règle positive puisque ce motif est fréquent. La confiance de $A \Rightarrow C$ est égale à 1. La règle est donc valide. Passons maintenant à la combinaison $AC \cup B$.

La corrélation pour la combinaison $AC \cup B$ est égale à -0,58 et donc inférieure ou égale à $-0, 50$. L'étude va donc concerner les règles $\overline{AC} \Rightarrow B$ et $AC \Rightarrow \overline{B}$ en analysant uniquement la confiance. La confiance de la règle $\overline{AC} \Rightarrow B$ est égale à $\frac{sup(\overline{AC}B)}{sup(\overline{AC})} = \frac{0,50}{0,50} = 1$. La confiance de la règle $AC \Rightarrow \overline{B}$ est égale à $\frac{sup(AC\overline{B})}{sup(AC)} = \frac{0,25}{0,50} = 0, 50$. La règle $\overline{AC} \Rightarrow B$ possède une confiance supérieure au seuil de la confiance donnée en paramètre, par conséquent uniquement cette règle sera conservée.

[Antonie and Zaïane, 2004] ne vérifient à aucun moment le support des règles mixtes $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. Afin d'obtenir des résultats comparables aux autres méthodes, nous considérons que la vérification du support est un oubli non délibéré de leur part et nous vérifions le support avant de générer les règles mixtes. Le tableau 2.19 récapitule l'ensemble des règles extraites avec l'ajout de la contrainte du support minimum sur l'ensemble des règles.

En conclusion, l'algorithme de [Antonie and Zaïane, 2004] génère 69 règles au total : 5 règles positives $X \Rightarrow Y$, 24 règles négatives du type $\overline{X} \Rightarrow Y$, 40 règles négatives du type $X \Rightarrow \overline{Y}$ et 0 règle négative du type $\overline{X} \Rightarrow \overline{Y}$. Une analyse comparative sur les règles extraites de cet algorithme par rapport aux autres algorithmes est disponible dans le *chapitre 6*.

Nous avons également regardé quel est l'impact de notre correction sur les résultats. Dans cet exemple, l'ajout des contraintes $sup(\overline{X}Y) \geqslant min_{sup}$ et $sup(X\overline{Y}) \geqslant min_{sup}$ n'entraînent aucune modification sur les règles extraites. Par conséquent, notre modification n'a aucun impact.

Règle	Support	Confiance	Règle	Support	Confiance
$A \Rightarrow C$	0,50	1	$CD \Rightarrow A$	0,75	1
$D \Rightarrow A$	0,75	1	$D \Rightarrow AC$	0,75	1
$E \Rightarrow B$	0,50	1			
$\overline{A} \Rightarrow B$	0,50	1	$\overline{A} \Rightarrow BE$	0,50	1
$\overline{A} \Rightarrow E$	0,50	1	$\overline{AC} \Rightarrow B$	0,50	1
$\overline{AC} \Rightarrow BE$	0,50	1	$\overline{AC} \Rightarrow E$	0,50	1
$\overline{ACD} \Rightarrow B$	0,75	1	$\overline{AD} \Rightarrow B$	0,75	1
$\overline{B} \Rightarrow A$	0,75	1	$\overline{B} \Rightarrow AC$	0,75	1
$\overline{B} \Rightarrow ACD$	0,75	1	$\overline{B} \Rightarrow AD$	0,75	1
$\overline{B} \Rightarrow CD$	0,75	1	$\overline{B} \Rightarrow D$	0,75	1
$\overline{BE} \Rightarrow A$	0,50	1	$\overline{BE} \Rightarrow AC$	0,50	1
$\overline{BE} \Rightarrow C$	0,50	1	$\overline{C} \Rightarrow BE$	0,75	1
$\overline{C} \Rightarrow E$	0,75	1	$\overline{CD} \Rightarrow B$	0,75	1
$\overline{D} \Rightarrow B$	0,75	1	$\overline{E} \Rightarrow A$	0,50	1
$\overline{E} \Rightarrow AC$	0,50	1	$\overline{E} \Rightarrow C$	0,50	1
$A \Rightarrow \overline{BCE}$	0,50	1	$A \Rightarrow \overline{BE}$	0,50	1
$A \Rightarrow \overline{CE}$	0,50	1	$A \Rightarrow \overline{E}$	0,50	1
$AB \Rightarrow \overline{E}$	0,75	1	$ABC \Rightarrow \overline{E}$	0,75	1
$AC \Rightarrow \overline{BE}$	0,50	1	$AC \Rightarrow \overline{E}$	0,50	1
$ACD \Rightarrow \overline{B}$	0,75	1	$ACD \Rightarrow \overline{E}$	0,75	1
$AD \Rightarrow \overline{B}$	0,75	1	$AD \Rightarrow \overline{BC}$	0,75	1
$AD \Rightarrow \overline{E}$	0,75	1	$B \Rightarrow \overline{ACD}$	0,75	1
$B \Rightarrow \overline{AD}$	0,75	1	$B \Rightarrow \overline{CD}$	0,75	1
$B \Rightarrow \overline{D}$	0,75	1	$BC \Rightarrow \overline{AD}$	0,50	1
$BC \Rightarrow \overline{D}$	0,50	1	$BCE \Rightarrow \overline{A}$	0,75	1
$BE \Rightarrow \overline{A}$	0,50	1	$BE \Rightarrow \overline{AC}$	0,50	1
$BE \Rightarrow \overline{CD}$	0,50	1	$BE \Rightarrow \overline{D}$	0,50	1
$CD \Rightarrow \overline{B}$	0,75	1	$CD \Rightarrow \overline{BE}$	0,75	1
$CD \Rightarrow \overline{E}$	0,75	1	$CE \Rightarrow \overline{A}$	0,75	1
$D \Rightarrow \overline{B}$	0,75	1	$D \Rightarrow \overline{BC}$	0,75	1
$D \Rightarrow \overline{BE}$	0,75	1	$D \Rightarrow \overline{E}$	0,75	1
$E \Rightarrow \overline{A}$	0,50	1	$E \Rightarrow \overline{AB}$	0,50	1
$E \Rightarrow \overline{ABC}$	0,50	1	$E \Rightarrow \overline{AC}$	0,50	1
$E \Rightarrow \overline{ACD}$	0,50	1	$E \Rightarrow \overline{AD}$	0,50	1
$E \Rightarrow \overline{CD}$	0,50	1	$E \Rightarrow \overline{D}$	0,50	1

Tableau 2.19 – Règles extraites sur la base d'exemple classées par type de règles

2.4 Algorithme proposé par Cornelis *et al.*

Dans ce qui suit, nous présentons tout d'abord les différentes contraintes que doivent respecter les règles pour être considérées comme valides et être extraites par la méthode de [Cornelis et al., 2006]. Nous présentons ensuite les différents algorithmes utilisés au cours du processus d'extraction. Le processus d'extraction se divise en cinq parties. En effet, les règles positives, mixtes et entièrement négatives (*avec la prémisse et la conclusion négatives*) sont recherchées à partir d'ensembles différents de motifs. L'exemple fil-rouge est ensuite repris afin de dérouler l'algorithme.

2.4.1 Critères de validité d'une règle

[Cornelis et al., 2006] basent leur méthode uniquement sur le support et la confiance afin d'extraire les règles d'association positives et négatives. Leur choix, de ne pas utiliser de mesure additionnelle, se justifie par la volonté de rendre la méthode plus intuitive pour les utilisateurs.

La génération des règles valides est effectuée à partir de trois ensembles de motifs : les fréquents positifs XY, les fréquents mixtes $\overline{X}Y$ et les fréquents négatifs $\overline{X}\,\overline{Y}$. Par conséquent, chaque type de règle va devoir respecter un support minimum. Une deuxième contrainte est ajoutée pour les règles négatives afin d'éliminer certaines règles inintéressantes : la minimalité de la négation. Le motif $\overline{Xi}\,\overline{Y}$ est considéré comme minimal s'il n'existe pas de motif $\overline{X'}\,\overline{Y'}$ fréquent tel que $X' \subseteq Xi$ et $Y' \subseteq Y$. Chaque ensemble de motifs fréquents va permettre de générer uniquement un seul type de règles : les règles positives $X \Rightarrow Y$ à partir des motifs positifs XY, les règles mixtes $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ à partir des motifs mixtes $\overline{X}Y$, et les règles négatives $\overline{X} \Rightarrow \overline{Y}$ à partir des motifs négatifs $\overline{X}\,\overline{Y}$. En ce qui concerne la génération des règles mixtes $X \Rightarrow \overline{Y}$, nous avons détecté une erreur car ils génèrent cette règle à partir du motif $\overline{X}Y$ et ne vérifient donc ni le support du motif $X\overline{Y}$ ni la minimalité de la négation du motif $\overline{Y}$. Afin de résoudre ces deux problèmes, nous avons modifié l'algorithme en extrayant à partir des motifs mixtes $\overline{X}Y$ les règles $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$. La dernière étape consiste à vérifier la validité de la confiance de ces règles.

$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$
$sup(XY) \geqslant min_{sup}$ $conf(X \Rightarrow Y) \geqslant min_{conf}$	$sup(\overline{X}Y) \geqslant min_{sup}$ $conf(\overline{X} \Rightarrow Y) \geqslant min_{conf}$ $\overline{X}$ *minimal*
$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$
retrait : $sup(\overline{X}Y) \geqslant min_{sup}$ ajout : $sup(X\overline{Y}) \geqslant min_{sup}$ $conf(X \Rightarrow \overline{Y}) \geqslant min_{conf}$ retrait : $\overline{X}$ *minimal* ajout : $\overline{Y}$ *minimal*	$sup(\overline{X}\,\overline{Y}) \geqslant min_{sup}$ $conf(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{conf}$ $sup(X) \geqslant min_{sup}$ $sup(Y) \geqslant min_{sup}$ $\overline{X}$ et $\overline{Y}$ *minimaux*

Tableau 2.20 – Règles valides

Le tableau 2.20 récapitule les contraintes pour qu'une règle soit considérée valide par [Cornelis et al., 2006]. Nous renseignons également les contraintes ajoutées à notre implémentation : nous remplaçons les contraintes $sup(\overline{X}Y) \geqslant min_{sup}$ et $\overline{X}$ *minimal* par les contraintes $sup(X\overline{Y}) \geqslant min_{sup}$ et $\overline{Y}$ *minimal* pour la génération des règles $X \Rightarrow \overline{Y}$.

En conclusion, l'ensemble des règles $X \Rightarrow Y$, $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$ doivent respecter la contrainte d'un support et d'une confiance minimum. De plus, les règles mixtes ainsi que les règles négatives doivent être extraites à partir de motifs fréquents qui respectent la contrainte de minimalité de la négation. Une dernière condition est ajoutée pour les règles $\overline{X} \Rightarrow \overline{Y}$ où il faut vérifier que les motifs X et Y sont fréquents.

2.4.2 Méthode d'extraction

[Cornelis et al., 2006] vont procéder en cinq étapes pour extraire les règles d'association positives et négatives valides :

1. rechercher l'ensemble $L(P_1)$ des motifs fréquents X,
2. rechercher l'ensemble $L(P_2)$ des motifs négatifs fréquents $\overline{X}$,
3. rechercher l'ensemble $L(P_3)$ des motifs négatifs fréquents $\overline{X}\,\overline{Y}$,
4. rechercher l'ensemble $L(P_4)$ des motifs mixtes fréquents $\overline{X}Y$,
5. générer les règles positives et négatives valides.

Pour l'**étape (1)**, les auteurs laissent le choix de l'algorithme pour rechercher les motifs fréquents X. Par conséquent, cette recherche peut être effectuée par l'algorithme `Apriori`. Les motifs fréquents X extraits seront stockés dans l'ensemble $L(P_1)$.

Pour l'**étape (2)**, ils génèrent les motifs négatifs fréquents $\overline{X}$ à partir des motifs fréquents précédemment trouvés lors de l'étape (1). Cette recherche est immédiate grâce à la formule suivante : $sup(\overline{X}) = 1 - sup(X)$. Cela revient donc à récupérer les motifs $X \in L(P_1)$ tel que $1 - sup(X) \geqslant min_{sup}$. Cette étape pose la question de son intérêt puisque l'ensemble $L(P_2)$ des motifs négatifs fréquents $\overline{X}$ n'est pas utilisé dans la suite de l'algorithme.

Dans l'**étape (3)**, les auteurs recherchent l'ensemble des motifs négatifs fréquents $\overline{X}\,\overline{Y}$ où $\overline{X}$ et $\overline{Y}$ sont des motifs négatifs fréquents minimaux. Cette recherche est effectuée par la fonction *fréquentsNégatifs* (*cf. algorithme 9*).

[Cornelis et al., 2006] commencent par initialiser l'ensemble $L(P_3)$ des motifs négatifs fréquents $\overline{X}\,\overline{Y}$ et l'ensemble des motifs candidats $C(P_3)$ aux ensembles vides (*lignes 1 et 2*). Les *2*-motifs candidats $\overline{i_1}\,\overline{i_2}$ sont ensuite générés à partir de deux items fréquents i_1 et i_2 trouvés à l'étape (1) (*ligne 3*). Le processus suivant est réitéré (*lignes 4 à 12*) jusqu'à ce que l'ensemble des candidats soit vide, c'est-à-dire lorsque $C(P_3)_k = \varnothing$. Puis, pour toutes les combinaisons XY de l'ensemble $C(P_3)_k$, ils vont tout d'abord calculer le support du motif XY avec la fonction *support* (*ligne 6*). Si ce motif est fréquent (*ligne 7*), alors il est ajouté à l'ensemble résultat $L(P_3)_k$ (*ligne 8*). Sinon, les auteurs vont générer le prochain niveau de candidats en ajoutant à la partie négative ($\overline{X}$ *ou* $\overline{Y}$) un item fréquent i non présent dans XY (*ligne 10*). Pour retenir les nouveaux candidats $\overline{Xi}\,\overline{Y}$

Algorithme 9 : *fréquentsNégatifs* - Recherche des motifs négatifs fréquents $\overline{X}\,\overline{Y}$

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, ensemble $L(P_1)$ des motifs fréquents X
Sortie : ensemble $L(P_3)$ des motifs négatifs fréquents $\overline{X}\,\overline{Y}$

1 $L(P_3) = \varnothing$
2 $C(P_3) = \varnothing$
3 $C(P_3)_2 = \{\overline{i_1}\,\overline{i_2}$ tel que $i_1, i_2 \in L(P_1)_1$ et $i_1 \neq i_2\}$
4 **pour** $(k = 2; C(P_3)_k \neq \varnothing; k++)$ **faire**
5 **pour toute** *combinaison* $M = XY \in C(P_3)_k$ **faire**
6 $s = support(\mathcal{D}, M)$;
7 **si** $s \geqslant min_{sup}$ **alors**
8 $L(P_3)_k = L(P_3)_k \cup \{M\}$
9 **sinon**
10 **pour tout** $i \in L(P_1)_1$ *tel que* $i \notin XY$ **faire**
11 $C(P_3)_{k+1} = C(P_3)_{k+1} \cup \{\overline{Xi}\,\overline{Y}$ tel que Xi fréquent et $\overline{Xi}\,\overline{Y}$ minimal$\}$
12 $C(P_3)_{k+1} = C(P_3)_{k+1} \cup \{\overline{X}\,\overline{Yi}$ tel que Yi fréquent et $\overline{X}\,\overline{Yi}$ minimal$\}$
13 **retourner** $L(P_3)$

ou $\overline{X}\,\overline{Yi}$ (*lignes 11 et 12*), il faut tout d'abord vérifier que les motifs Xi ou Yi soient fréquents (*i.e.* $Xi \in L(P_1)$ *et* $Yi \in L(P_1)$) mais également que les nouveaux candidats soient des motifs négatifs minimaux. Le motif $\overline{Xi}\,\overline{Y}$ est considéré comme minimal s'il n'existe pas $\overline{X'}\,\overline{Y'} \in L(P_3)$ tel que $X' \subseteq Xi$ et $Y' \subseteq Y$. Lorsque tous les candidats sont parcourus, l'ensemble $L(P_3)$ des motifs négatifs fréquents $\overline{X}\,\overline{Y}$ est retourné (*ligne 13*).

Cette étape est quelque peu redondante puisqu'à partir d'un motif candidat non fréquent $\overline{X}\,\overline{Y}$, les auteurs vont générer tous les candidats possibles puis vérifier la minimalité du nouveau motif. Par conséquent, des candidats similaires peuvent être générés de différentes manières. Par exemple, $\overline{a}\,\overline{bc}$ peut être obtenu à partir du candidat non fréquent $\overline{a}\,\overline{b}$ en ajoutant l'item c mais également à partir du candidat non fréquent $\overline{a}\,\overline{c}$ en ajoutant l'item b. Cette étape aurait pu être optimisée. En effet, si l'étape (2) est utilisée afin de rechercher les motifs négatifs fréquents minimaux, l'étape (3) peut ensuite combiner directement les motifs minimaux. Cela permettrait également de veiller à ce que le motif soit fréquent.

Pour l'**étape (4)**, les auteurs vont utiliser la fonction *fréquentsMixtes* (*cf. algorithme* 10) afin de rechercher l'ensemble des motifs mixtes fréquents $\overline{X}Y$. Avant d'expliquer cette étape, commençons par introduire les notations $C(P_4)_{k,p}$ et $L(P_4)_{k,p}$ qui correspondent respectivement aux motifs candidats et aux motifs fréquents possédant k items positifs et p items négatifs.

Algorithme 10 : *fréquentsMixtes* - Recherche des motifs mixtes fréquents $\overline{X}Y$

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, ensemble $L(P_1)$ des motifs fréquents X
Sortie : ensemble $L(P_4)$ des motifs mixtes fréquents $\overline{X}Y$

1 $L(P_4) = \varnothing$
2 $C(P_4) = \varnothing$
3 $C(P_4)_{1,1} = \{\overline{i_1}i_2$ tel que $i_1, i_2 \in L(P_1)_1$ et $i_1 \neq i_2\}$
4 **pour** $(k = 1; C(P_4)_{k,1} \neq \varnothing; k++)$ **faire**
5 **pour** $(p = 1; C(P_4)_{k,p} \neq \varnothing; p++)$ **faire**
6 **pour tout** *motif* $M \in C(P_4)_{k,p}$ **faire**
7 $s = support(\mathcal{D}, M)$;
8 **si** $s \geqslant min_{sup}$ **alors**
9 $L(P_4)_{k,p} = L(P_4)_{k,p} \cup \{M\}$
10 **pour tout** *motif* $(M_1 = \overline{X}Yi_1) \in L(P_4)_{k,p}$ *et* $(M_2 = \overline{X}Yi_2) \in L(P_4)_{k,p}$ **faire**
11 $M = \overline{X}Yi_1i_2$
12 **si** $\nexists X' \subset X$ *tel que* $sup(\overline{X'}Yi_1i_2) \geqslant min_{sup}$ *et* $\nexists Y' \subset Yi_1i_2$ *tel que* $sup(\overline{X}Y') < min_{sup}$ **alors**
13 $C(P_4)_{k,p+1} = C(P_4)_{k,p+1} \cup \{M\}$
14 **pour tout** *motif* $X \in L(P_1)_{k+1}$ *et* $i \in L(P_1)_1$ **faire**
15 $M = \overline{X}i$
16 **si** $\nexists X' \subset X$ *tel que* $\overline{X'}i \in L(P_4)$ **alors**
17 $C(P_4)_{k+1,1} = C(P_4)_{k+1,1} \cup \{\overline{X}i\}$
18 **retourner** $L(P_4)$

[Cornelis et al., 2006] commencent par initialiser l'ensemble $L(P_4)$ des motifs mixtes fréquents $\overline{X}Y$ et l'ensemble candidat $C(P_4)$ aux ensembles vides (*lignes 1 et 2*). Les 2-motifs candidats $\overline{i_1}i_2$ sont ensuite générés à partir de deux items fréquents i_1 et i_2 trouvés à l'étape (1) (*ligne 3*). Le processus suivant est réitéré (*lignes 4 à 17*) jusqu'à ce que l'ensemble des candidats soit vide, c'est-à-dire lorsque $C(P_4)_{k,1} = \varnothing$. La génération des candidats s'effectue en augmentant la taille du motif positif Y (*ligne 4*) puis en augmentant celle du motif négatif X (*ligne 5*), c'est-à-dire en commençant par augmenter la partie positive du motif puis en augmentant sa partie négative. Pour chaque motif M contenu dans $C(P_4)_{k,p}$ (*ligne 6*), les auteurs commencent par calculer le support avec la fonction *support* (*ligne 7*). Si le motif M est fréquent (*ligne 8*), alors M est ajouté à l'ensemble résultat $L(P_4)_{k,p}$ (*ligne 8*).

Les auteurs vont construire itérativement les prochains candidats en augmentant d'abord la partie positive Y (*lignes 10 à 13*). Ainsi, pour tout motif $(M_1 = \overline{X}Yi_1) \in L(P_4)_{k,p}$ et $(M_2 = \overline{X}Yi_2) \in L(P_4)_{k,p}$ (*ligne 10*), ils vont générer le motif M potentiellement candidat $\overline{X}Yi_1i_2$ (*ligne 11*). Ils vérifient ensuite la minimalité du motif M (*ligne 12*). Pour se faire, deux conditions doivent être vérifiées. La première est qu'il ne doit pas y avoir de sous-ensemble X' de X tel que le motif $\overline{X'}Yi_1i_2$ soit fréquent. Sans cette condition, si à l'itération précédente nous avions le motif $\overline{X'}Yi_1i_2$ fréquent et si $X' \subset X$ alors le motif $\overline{X}Yi_1i_2$ sera forcément fréquent car le fait d'augmenter le motif négatif va potentiellement augmenter le support. Et la seconde condition est qu'il ne doit pas y avoir

de sous-ensemble Y' de Y tel que le motif $\overline{X}Y'$ soit non fréquent. Si le motif $\overline{X}Y'$ est non fréquent et si $Y' \subset Yi_1i_2$, alors $\overline{X'}Yi_1i_2$ sera forcément non fréquent car le fait d'augmenter le motif positif va potentiellement diminuer le support. Ces deux conditions vont donc permettre de générer des motifs négatifs minimaux et d'optimiser le processus afin d'éviter de calculer à la prochaine itération le support (*ligne 7*) d'un motif que l'on sait déjà non fréquent. Si le motif M est minimal alors il est ajouté à l'ensemble $C(P_4)_{k,p+1}$ des candidats pour la prochaine itération (*ligne 13*).

Les dernières étapes de cet algorithme consistent à augmenter le motif négatif X (*lignes 14 à 17*) en récupérant les négations des motifs fréquents de plus grande taille. Les auteurs vont également récupérer les items i de l'ensemble des motifs fréquents $L(P_1)$ et vont l'ajouter au motif X en vérifiant que le nouveau motif $\overline{X}i$ est minimal. Si le motif $\overline{X}i$ est minimal (*ligne 16*) alors il est ajouté à l'ensemble des motifs candidats $C(P_4)_{k+1,1}$ (*ligne 17*).

La dernière étape qui est l'**étape (5)** génère toutes les règles positives et négatives valides à partir des différents motifs précédemment extraits à l'aide de l'*algorithme* 11.

Algorithme 11 : Extraction des règles d'association positives et négatives

Entrées : base de données D, ensemble $L(P_1)$ des motifs fréquents X, ensemble $L(P_3)$ des motifs négatifs fréquents $\overline{X}\,\overline{Y}$, ensemble $L(P_4)$ des motifs mixtes fréquents $\overline{X}Y$, confiance minimum min_{conf}

Sortie : ensemble des règles valides R

```
1  R = ∅
2  pour tout motif XY ∈ L(P1) faire
3      R = R ∪ règles(LP1, min_conf)
4  pour tout motif X̄ Ȳ ∈ L(P3) faire
5      si conf(X̄ ⇒ Ȳ) ⩾ min_conf alors
6          R = R ∪ {X̄ ⇒ Ȳ}
7  pour tout motif X̄Y ∈ L(P4) faire
8      si conf(X̄ ⇒ Y) ⩾ min_conf alors
9          R = R ∪ {X̄ ⇒ Y}
10     si conf(X ⇒ Ȳ) ⩾ min_conf alors
11         R = R ∪ {X ⇒ Ȳ}
12 retourner R
```

Les auteurs commencent par initialiser l'ensemble R des règles valides à l'ensemble vide (*ligne 1*). Pour chaque motif XY de $L(P_1)$ (*ligne 2*), les auteurs vont utiliser la fonction *règles* d'`Apriori` (*cf. algorithme* 3) pour générer les règles valides (*ligne 3*). Puis, ils vont rechercher les règles négatives valides. Pour chaque motif $\overline{X}\,\overline{Y}$ de $L(P_3)$ (*ligne 4*), ils vérifient la validité de la confiance de la règle $\overline{X} \Rightarrow \overline{Y}$ (*ligne 5*) et ajoutent cette dernière à l'ensemble R des règles valides (*ligne 6*). Et enfin, ils cherchent les règles mixtes. Pour chaque motif $\overline{X}Y$ de $L(P_4)$ (*ligne 7*), ils vérifient tout d'abord la validité de la confiance de la règle $\overline{X} \Rightarrow Y$ (*ligne 8*) puis celle de la règle $X \Rightarrow \overline{Y}$ (*ligne 10*). Si le seuil de la confiance est vérifié, les règles sont ajoutées à l'ensemble R (*lignes 9 et 11*). L'ensemble R des règles valides est ensuite retourné (*ligne 12*).

2.4.3 Exemple

Nous allons maintenant dérouler l'algorithme de [Cornelis et al., 2006] sur notre exemple fil-rouge (*cf. tableau* 2.21). Nous conservons les mêmes seuils que précédemment pour le support et la confiance, à savoir respectivement 0,25 et 0,80.

A	B	C	D	E
1	0	1	1	0
0	1	1	0	1
1	1	1	0	0
0	1	0	0	1

TABLEAU 2.21 – Exemple fil-rouge

Nous allons dérouler la version corrigée de l'algorithme afin que l'ensemble des règles respecte le support minimum et que les règles négatives possèdent bien un motif négatif minimal. Nous mesurons l'impact de notre modification sur les résultats dans la conclusion de cet exemple.

La méthode de [Cornelis et al., 2006] se déroule en cinq étapes.

1) Extraction des motifs fréquents X

Pour chercher les motifs fréquents, les auteurs laissent le choix de l'algorithme. Par conséquent, nous utilisons `Apriori`. Nous rappelons l'ensemble des motifs fréquents trouvés par l'algorithme `Apriori` dans le tableau 2.22.

Motif	Support	Motif	Support	Motif	Support
A	0,50	AB	0,25	ABC	0,25
AC	0,50	ACD	0,25	AD	0,25
B	0,75	BC	0,50	BCE	0,25
BE	0,50	C	0,75	CD	0,25
CE	0,25	D	0,25	E	0,50

TABLEAU 2.22 – Motifs fréquents accompagnés de leur support

2) Extraction des motifs négatifs fréquents $\overline{X}$

Les motifs négatifs fréquents $\overline{X}$ sont ensuite générés à partir des motifs positifs. Le tableau 2.23 fournit les motifs négatifs fréquents générés.

Motif	Support	Motif	Support	Motif	Support
$\overline{A}$	0,50	$\overline{AB}$	0,75	$\overline{ABC}$	0,75
$\overline{AC}$	0,50	$\overline{ACD}$	0,75	$\overline{AD}$	0,75
$\overline{B}$	0,25	$\overline{BC}$	0,50	$\overline{BCE}$	0,75
$\overline{BE}$	0,50	$\overline{C}$	0,25	$\overline{CD}$	0,75
$\overline{CE}$	0,75	$\overline{D}$	0,75	$\overline{E}$	0,50

TABLEAU 2.23 – Motifs négatifs fréquents $\overline{X}$ accompagnés de leur support

3) Extraction des motifs négatifs fréquents $\overline{X}\,\overline{Y}$

Nous cherchons ensuite l'ensemble des motifs négatifs fréquents $\overline{X}\,\overline{Y}$. La création des candidats $\overline{X}\,\overline{Y}$ de taille 2 s'opère en combinant deux négations de motifs fréquents de taille 1. Le tableau 2.24 présente les candidats $\overline{X}\,\overline{Y}$ de taille 2. La combinaison $\overline{X}\,\overline{Y}$ étant similaire à la combinaison $\overline{Y}\,\overline{X}$, seule la première sera renseignée dans le tableau.

Item 1	Item 2	Item 1	Item 2	Item 1	Item 2
$\overline{A}$	$\overline{B}$	$\overline{A}$	$\overline{C}$	$\overline{A}$	$\overline{D}$
$\overline{A}$	$\overline{E}$	$\overline{B}$	$\overline{C}$	$\overline{B}$	$\overline{D}$
$\overline{B}$	$\overline{E}$	$\overline{C}$	$\overline{D}$	$\overline{C}$	$\overline{E}$
$\overline{D}$	$\overline{E}$				

TABLEAU 2.24 – Candidats $\overline{X}\,\overline{Y}$ de taille 2

La prochaine étape consiste à tester la valeur du support. Si le support vérifie le seuil minimal alors les combinaisons sont ajoutées à l'ensemble des motifs négatifs fréquents $\overline{X}\,\overline{Y}$, sinon elles sont utilisées pour créer le prochain niveau de candidats. Le tableau 2.25 répertorie les motifs négatifs fréquents.

Item 1	Item 2	Support	Item 1	Item 2	Support
$\overline{A}$	$\overline{C}$	0,25	$\overline{A}$	$\overline{D}$	0,50
$\overline{B}$	$\overline{E}$	0,25	$\overline{C}$	$\overline{D}$	0,25
$\overline{D}$	$\overline{E}$	0,25			

TABLEAU 2.25 – Motifs négatifs $\overline{X}\,\overline{Y}$ fréquents de taille 2 accompagnés de leur support

Les autres combinaisons non fréquentes vont être utilisées pour générer le prochain niveau de candidats. Nous rappelons que pour se faire il faut ajouter à la partie négative ($\overline{X}$ *ou* $\overline{Y}$) un item fréquent i non présent dans XY. Le nouveau candidat $\overline{X}i\overline{Y}$ ou $\overline{X}\,\overline{Y}i$ sera retenu s'il respecte les deux conditions suivantes : Xi ou Yi fréquent et $\overline{Xi}\,\overline{Y}$ ou

$\overline{X}\,\overline{Yi}$ minimal. Déroulons la méthode sur la combinaison $\overline{A}\,\overline{B}$ non fréquente.

Pour $\overline{A}\,\overline{B}$: les items C, D et E ne sont pas présents. On ajoute donc les items aux parties négatives :
— $\overline{AC}\,\overline{B}$: ok.
— $\overline{AD}\,\overline{B}$: ok.
— $\overline{AE}\,\overline{B}$: éliminé car AE n'est pas fréquent.
— $\overline{A}\,\overline{BC}$: éliminé car $\overline{A}\,\overline{C}$ existe déjà.
— $\overline{A}\,\overline{BD}$: éliminé car BD n'est pas fréquent.
— $\overline{A}\,\overline{BE}$: ok.

On teste ensuite le support de ces combinaisons. Aucune des combinaisons n'est valide et par conséquent elles serviront à créer le prochain niveau de candidats.

Pour $\overline{AC}\,\overline{B}$: les items D et E ne sont pas présents. On ajoute donc les items aux parties négatives :
— $\overline{ACD}\,\overline{B}$: ok.
— $\overline{ACE}\,\overline{B}$: éliminé car ACE n'est pas fréquent.
— $\overline{AC}\,\overline{BD}$: éliminé car BD n'est pas fréquent.
— $\overline{AC}\,\overline{BE}$: ok.

Pour $\overline{AD}\,\overline{B}$: les items C et E ne sont pas présents. On ajoute donc les items aux parties négatives :
— $\overline{ACD}\,\overline{B}$: ok.
— $\overline{ADE}\,\overline{B}$: éliminé car ADE n'est pas fréquent.
— $\overline{AD}\,\overline{BC}$: éliminé car $\overline{A}\,\overline{C}$ existe déjà.
— $\overline{AD}\,\overline{BE}$: éliminé car $\overline{D}\,\overline{E}$ existe déjà.

Pour $\overline{A}\,\overline{BE}$: les items C et D ne sont pas présents. On ajoute donc les items aux parties négatives :
— $\overline{AC}\,\overline{BE}$: ok.
— $\overline{AD}\,\overline{BE}$: éliminé car $\overline{D}\,\overline{E}$ existe déjà.
— $\overline{A}\,\overline{BCE}$: éliminé car $\overline{A}\,\overline{C}$ existe déjà.
— $\overline{A}\,\overline{BDE}$: éliminé car BDE n'est pas fréquent.

On teste ensuite le support des combinaisons non éliminées. Aucune des combinaisons n'est valide et par conséquent elles serviront à créer le prochain niveau de candidats.

Pour $\overline{ACD}\,\overline{B}$: l'item E n'est pas présent. On ajoute donc cet item aux parties négatives :
— $\overline{ACDE}\,\overline{B}$: éliminé car $\overline{B}\,\overline{E}$ existe déjà.
— $\overline{ACD}\,\overline{BE}$: éliminé car $\overline{D}\,\overline{E}$ existe déjà.

Pour $\overline{AC}\,\overline{BE}$: l'item D n'est pas présent. On ajoute donc cet item aux parties négatives :
— $\overline{ACD}\,\overline{BE}$: éliminé car $\overline{D}\,\overline{E}$ existe déjà.
— $\overline{AC}\,\overline{BDE}$: éliminé car $\overline{A}\,\overline{D}$ et $\overline{C}\,\overline{D}$ existent.

La génération des candidats pour cet exemple s'arrête à cette étape car plus aucun candidat ne peut être généré.

Le processus de génération des candidats est identique pour l'ensemble des combinaisons $\overline{A}\,\overline{B}$, $\overline{A}\,\overline{E}$, $\overline{B}\,\overline{C}$, $\overline{B}\,\overline{D}$ et $\overline{C}\,\overline{E}$. Cependant aucun motif négatif fréquent ne sera généré par cette procédure. L'ensemble des motifs négatifs $\overline{X}\,\overline{Y}$ fréquents se limite donc aux motifs de taille 2 présents dans le tableau 2.25.

4) Extraction des motifs mixtes fréquents $\overline{X}Y$

Nous poursuivons ensuite avec la recherche des motifs mixtes fréquents $\overline{X}Y$. La création des candidats $\overline{X}Y$ de taille 2 s'opère en combinant une négation de motif fréquent de taille 1 avec un autre motif fréquent de taille 1. Le tableau 2.26 présente les candidats $\overline{X}Y$ de taille 2.

Item négatif	Item positif	Item négatif	Item positif	Item négatif	Item positif
$\overline{A}$	B	$\overline{A}$	C	$\overline{A}$	D
$\overline{A}$	E	$\overline{B}$	A	$\overline{B}$	C
$\overline{B}$	D	$\overline{B}$	E	$\overline{C}$	A
$\overline{C}$	B	$\overline{C}$	D	$\overline{C}$	E
$\overline{D}$	A	$\overline{D}$	B	$\overline{D}$	C
$\overline{D}$	E	$\overline{E}$	A	$\overline{E}$	B
$\overline{E}$	C	$\overline{E}$	D		

TABLEAU 2.26 – Candidats $\overline{X}Y$ de taille 2

La prochaine étape consiste à tester la valeur du support. Si le support est valide alors les combinaisons sont ajoutées à l'ensemble des motifs mixtes fréquents $\overline{X}Y$. Le tableau 2.27 répertorie les motifs mixtes fréquents.

La génération des motifs mixtes candidats $\overline{X}Y$ se poursuit tout d'abord en augmentant le motif positif Y. Nous rappelons d'abord la procédure. Pour tous les motifs M_1 et M_2 tels que $M_1 \neq M_2$ où la partie négative est identique (*i.e.* $\overline{X} \in M_1 = \overline{X} \in M_2$) et où la partie positive (*i.e.* $Yi_1 \in M_1$ et $Yi_2 \in M_2$) partagent les mêmes $(k-1)$ items (*c'est-à dire le motif* Y. M_1 *contiendra en plus l'item* i_1 *qui ne sera pas dans le motif* M_2, *et* M_2 *contiendra l'item* i_2 *qui ne sera pas dans le motif* M_1), ils vont générer le motif M potentiellement candidat $\overline{X}Yi_1i_2$. Ce nouveau candidat potentiel $\overline{X}Yi_1i_2$ est ensuite analysé pour vérifier sa minimalité. La première vérification consiste à confirmer qu'il n'y a pas de sous-ensemble X' de X tel que le motif $\overline{X'}Yi_1i_2$ soit fréquent. La seconde vérification consiste à confirmer qu'il n'y a pas de sous-ensemble Y' de Y tel que le motif $\overline{X}Y'$ soit non fréquent. Si le motif est minimal alors il est ajouté à l'ensemble des motifs candidats de taille 3. Déroulons la procédure sur les motifs $\overline{D}A$, $\overline{D}B$ $\overline{D}C$ et $\overline{D}E$.

En combinant ces motifs, on obtient les motifs suivants : $\overline{D}AB$, $\overline{D}AC$, $\overline{D}AE$, $\overline{D}BC$, $\overline{D}BE$ et $\overline{D}CE$. On vérifie ensuite la contrainte de minimalité. Pour l'ensemble de ces

Item négatif	Item positif	Support	Item négatif	Item positif	Support
$\overline{A}$	B	0,50	$\overline{A}$	C	0,25
$\overline{A}$	E	0,50	$\overline{B}$	A	0,25
$\overline{B}$	C	0,25	$\overline{B}$	D	0,25
$\overline{C}$	B	0,25	$\overline{C}$	E	0,25
$\overline{D}$	A	0,25	$\overline{D}$	B	0,75
$\overline{D}$	C	0,50	$\overline{D}$	E	0,50
$\overline{E}$	A	0,25	$\overline{E}$	B	0,25
$\overline{E}$	C	0,25	$\overline{E}$	D	0,25

TABLEAU 2.27 – Motifs mixtes $\overline{X}Y$ fréquents de taille 2 accompagnés de leur support

motifs, il n'existe pas de sous-ensemble X' de X tel que le motif $\overline{X'}Yi_1i_2$ soit fréquent. La première condition est donc vérifiée. Pour la seconde condition concernant le motif $\overline{D}AE$, on retrouve le sous-ensemble A de AE tel que le motif $\overline{D}A$ est non fréquent. Par conséquent le motif $\overline{D}AE$ ne respecte pas la contrainte de minimalité et va donc être supprimé. À la prochaine itération les motifs $\overline{D}ABC$ et $\overline{D}BCE$ seront également retenu comme motifs fréquents.

La dernière étape de la procédure que nous rappelons ici, consiste à augmenter le motif négatif X en récupérant les négations des motifs fréquents de plus grande taille et en le combinant aux items i de l'ensemble des motifs fréquents. Il faut ensuite vérifier que le nouveau motif $\overline{X}i$ soit minimal. Déroulons la méthode sur le motif fréquent AC.

Pour le motif AC : les items fréquents B, D et E ne sont pas présents. On combine donc le motif AC avec ces items :
— $\overline{AC}B$: éliminé car $\overline{A}B$ est fréquent.
— $\overline{AC}D$: ok.
— $\overline{AC}E$: éliminé car $\overline{C}E$ est fréquent.

Le motif $\overline{AC}D$ est donc un motif candidat, cependant à la prochaine itération il ne vérifiera pas la contrainte du support et par conséquent il ne sera pas ajouté à l'ensemble des motifs fréquents. Le motif étant le seul candidat, nous ne pourrons pas le combiner pour créer le prochain niveau de candidats. La génération des candidats s'arrêtera donc là. L'ensemble des motifs mixtes fréquents de taille 3 et 4 est renseigné dans le tableau 2.28.

5) Génération des règles positives et négatives valides

La dernière étape de l'algorithme de [Cornelis et al., 2006] consiste à générer les règles. Les règles positives vont être générées à l'aide de la fonction *règles* d'`Apriori`. Cette fonction nous permet donc d'obtenir les mêmes règles que dans le chapitre 1 (*cf. tableau* 1.11). Pour les règles négatives $\overline{X} \Rightarrow \overline{Y}$, elles vont être générées à partir des motifs $\overline{X}\,\overline{Y}$ fréquents et retenues si leur confiance vérifie le seuil. Les motifs mixtes $\overline{X}Y$

Item négatif	Item positif	Support	Item négatif	Item positif	Support
$\overline{A}$	BC	0,25	$\overline{A}$	BCE	0,25
$\overline{A}$	BE	0,50	$\overline{A}$	CE	0,25
$\overline{B}$	AC	0,25	$\overline{B}$	ACD	0,25
$\overline{B}$	AD	0,25	$\overline{B}$	CD	0,25
$\overline{C}$	BE	0,25	$\overline{D}$	AB	0,25
$\overline{D}$	ABC	0,25	$\overline{D}$	AC	0,25
$\overline{D}$	BC	0,50	$\overline{D}$	BCE	0,25
$\overline{D}$	BE	0,50	$\overline{D}$	CE	0,25
$\overline{E}$	AB	0,25	$\overline{E}$	ABC	0,25
$\overline{E}$	AC	0,50	$\overline{E}$	ACD	0,25
$\overline{E}$	AD	0,25	$\overline{E}$	BC	0,25
$\overline{E}$	CD	0,25			

TABLEAU 2.28 – Autres motifs mixtes $\overline{X}Y$ fréquents accompagnés de leur support

fréquents permettront de générer les règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ après vérification de la confiance. Seulement, comme nous l'avons dit précédemment, le support du motif $X\overline{Y}$ n'est pas vérifié. Nous avons donc modifié l'algorithme de [Cornelis et al., 2006] et nous avons extrait les règles $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$ pour chaque motif $\overline{X}Y$. Les résultats sont visibles dans le tableau (*cf. tableau* 2.29).

Règle	Support	Confiance	Règle	Support	Confiance
$\overline{A} \Rightarrow B$	0,50	1	$\overline{A} \Rightarrow BE$	0,50	1
$\overline{A} \Rightarrow E$	0,50	1	$\overline{B} \Rightarrow A$	0,25	1
$\overline{B} \Rightarrow AC$	0,25	1	$\overline{B} \Rightarrow ACD$	0,25	1
$\overline{B} \Rightarrow AD$	0,25	1	$\overline{B} \Rightarrow C$	0,25	1
$\overline{B} \Rightarrow CD$	0,25	1	$\overline{B} \Rightarrow D$	0,25	1
$\overline{C} \Rightarrow B$	0,25	1	$\overline{C} \Rightarrow BE$	0,25	1
$\overline{C} \Rightarrow E$	0,25	1	$\overline{D} \Rightarrow B$	0,75	1
$\overline{E} \Rightarrow A$	0,50	1	$\overline{E} \Rightarrow AC$	0,50	1
$\overline{E} \Rightarrow C$	0,50	1			
$A \Rightarrow \overline{E}$	0,50	1	$AB \Rightarrow \overline{D}$	0,25	1
$AB \Rightarrow \overline{E}$	0,25	1	$ABC \Rightarrow \overline{D}$	0,25	1
$ABC \Rightarrow \overline{E}$	0,25	1	$AC \Rightarrow \overline{E}$	0,50	1
$ACD \Rightarrow \overline{B}$	0,25	1	$ACD \Rightarrow \overline{E}$	0,25	1
$AD \Rightarrow \overline{B}$	0,25	1	$AD \Rightarrow \overline{E}$	0,25	1
$B \Rightarrow \overline{D}$	0,75	1	$BC \Rightarrow \overline{D}$	0,50	1
$BCE \Rightarrow \overline{A}$	0,25	1	$BCE \Rightarrow \overline{D}$	0,25	1
$BE \Rightarrow \overline{A}$	0,50	1	$BE \Rightarrow \overline{D}$	0,50	1
$CD \Rightarrow \overline{B}$	0,25	1	$CD \Rightarrow \overline{E}$	0,25	1
$CE \Rightarrow \overline{A}$	0,25	1	$CE \Rightarrow \overline{D}$	0,25	1
$D \Rightarrow \overline{B}$	0,25	1	$D \Rightarrow \overline{E}$	0,25	1
$E \Rightarrow \overline{A}$	0,50	1	$E \Rightarrow \overline{D}$	0,50	1
$\overline{A} \Rightarrow \overline{D}$	0,50	1	$\overline{B} \Rightarrow \overline{E}$	0,25	1
$\overline{C} \Rightarrow \overline{A}$	0,25	1	$\overline{C} \Rightarrow \overline{D}$	0,25	1

TABLEAU 2.29 – Règles négatives extraites sur la base d'exemple classées par type de règles

En conclusion, l'algorithme de [Cornelis et al., 2006] génère 54 règles au total : 9 règles positives $X \Rightarrow Y$, 17 règles négatives du type $\overline{X} \Rightarrow Y$, 24 règles négatives du type $X \Rightarrow \overline{Y}$ et 4 règles négatives du type $\overline{X} \Rightarrow \overline{Y}$. Une analyse comparative sur les règles extraites de cet algorithme par rapport aux autres algorithmes est disponible dans le *chapitre 6*.

Nous avons également regardé quel est l'impact de notre correction sur les résultats. Dans cet exemple, le fait de remplacer les contraintes $sup(\overline{X}Y) \geqslant min_{sup}$ et $\overline{X}$ *minimal* par les contraintes $sup(X\overline{Y}) \geqslant min_{sup}$ et $\overline{Y}$ *minimal* lors de la génération des règles $X \Rightarrow \overline{Y}$ a un impact important sur les règles $X \Rightarrow \overline{Y}$ générées. Les autres types de règles ne sont pas impactés. Le tableau 2.30 restitue les règles $X \Rightarrow \overline{Y}$ extraites sans notre modification. Comme nous pouvons le voir, seules 6 règles sont communes aux deux versions : $A \Rightarrow \overline{E}$, $B \Rightarrow \overline{D}$, $D \Rightarrow \overline{B}$, $D \Rightarrow \overline{E}$, $E \Rightarrow \overline{A}$ et $E \Rightarrow \overline{D}$. Par ailleurs, nous constatons que la contrainte de minimalité du motif négatif pour ces règles n'est pas respectée. En effet, il n'y a que les 6 règles communes aux deux versions qui la

respectent. Cependant toutes les règles respectent bien la contrainte du support minimum.

Règle	Support	Confiance	Règle	Support	Confiance
$A \Rightarrow \overline{BCE}$	0,50	1	$A \Rightarrow \overline{BE}$	0,50	1
$A \Rightarrow \overline{CE}$	0,50	1	$A \Rightarrow \overline{E}$	0,50	1
$B \Rightarrow \overline{ACD}$	0,75	1	$B \Rightarrow \overline{AD}$	0,75	1
$B \Rightarrow \overline{CD}$	0,75	1	$B \Rightarrow \overline{D}$	0,75	1
$D \Rightarrow \overline{AB}$	0,25	1	$D \Rightarrow \overline{ABC}$	0,25	1
$D \Rightarrow \overline{B}$	0,25	1	$D \Rightarrow \overline{BC}$	0,25	1
$D \Rightarrow \overline{BCE}$	0,25	1	$D \Rightarrow \overline{BE}$	0,25	1
$D \Rightarrow \overline{CE}$	0,25	1	$D \Rightarrow \overline{E}$	0,25	1
$E \Rightarrow \overline{A}$	0,50	1	$E \Rightarrow \overline{AB}$	0,50	1
$E \Rightarrow \overline{ABC}$	0,50	1	$E \Rightarrow \overline{AC}$	0,50	1
$E \Rightarrow \overline{ACD}$	0,50	1	$E \Rightarrow \overline{AD}$	0,50	1
$E \Rightarrow \overline{CD}$	0,50	1	$E \Rightarrow \overline{D}$	0,50	1

TABLEAU 2.30 – Règles négatives $X \Rightarrow \overline{Y}$ originalement extraites

2.5 Conclusion

Dans ce chapitre, nous avons étudié trois approches d'extraction des règles d'association positives et négatives basées sur `Apriori`. Nous nous sommes focalisés sur les travaux de [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006]. [Wu et al., 2004] et [Antonie and Zaïane, 2004] proposent d'utiliser une ou plusieurs mesures supplémentaires afin d'éliminer certaines règles non intéressantes. L'utilisation des mesures supplémentaires (*valeur absolue de la nouveauté et facteur de certitude pour Wu et al., coefficient de corrélation pour Antonie et Zaïane*) va permettre de ne garder que les motifs relativement bien corrélés. Par conséquent cela supprimera les règles trop proches de l'indépendance. Cornelis *et al.* choisissent de garder une approche intelligible et ne vont donc pas utiliser de mesure supplémentaire.

Après avoir effectué une étude approfondie de ces trois méthodes, nous avons mis en avant essentiellement deux failles :

1. un nombre encore important de règles inintéressantes,
2. et un parcours non optimisé de recherche des règles.

L'objectif du prochain chapitre va être de combler partiellement ces deux failles et de présenter les optimisations que nous proposons d'intégrer à notre méthode d'extraction.

Chapitre 3

Optimisations de l'extraction des règles

Sommaire

3.1 Introduction

Dans ce chapitre, nous présentons les solutions que nous proposons pour résoudre les **deux failles** mises en avant dans le chapitre précédent.

Pour résoudre partiellement la **première faille** énoncée dans le chapitre 2 et qui est d'élaguer les règles inintéressantes nous faisons trois propositions. La **première proposition** est de baser l'extraction des règles d'association positives et négatives sur les motifs raisonnablement fréquents, et non plus sur les motifs fréquents comme le font [Agrawal and Srikant, 1994] dans l'algorithme `Apriori`. La **deuxième proposition** est d'utiliser une mesure de qualité pour sélectionner les règles valides : la mesure M_G [Guillaume, 2010]. Cette mesure va permettre d'éliminer une nouvelle catégorie de règles inintéressantes que les mesures classiquement utilisées (*support et confiance*) ne peuvent pas éliminer. La **dernière proposition** présente une nouvelle contrainte qui renforce notre souhait d'extraire les règles les plus intéressantes et qui permet d'extraire un nouveau type de règles : $\overline{x_1}..\overline{x_p} \Rightarrow \overline{y_1}..\overline{y_q}$.

Pour la résolution de la **seconde faille**, à savoir le parcours de recherche des règles non optimisé, nous allons démontrer que seulement la moitié des règles sont à étudier. De plus, nous réintégrons la propriété de la confiance, abandonnée par [Wu et al., 2004] et [Antonie and Zaïane, 2004], qui permet d'optimiser le parcours de recherche des règles. Et enfin, nous ajoutons également des propriétés d'élagage sous la forme de méta-règles en se basant sur la mesure M_G.

3.2 Élaguer les règles inintéressantes

Pour améliorer la qualité des règles extraites ainsi que les temps d'extraction de l'algorithme (*conséquence directe de la diminution du nombre de règles extraites*), nous proposons d'une part d'extraire les règles d'association positives et négatives à partir des motifs raisonnablement fréquents et d'autre part d'ajouter la mesure M_G à l'approche support/confiance pour élaguer certaines règles inintéressantes. Nous présentons ensuite une nouvelle contrainte éliminant d'autres règles inintéressantes et permettant d'extraire un nouveau type de règles négatives : $\overline{X}_1\overline{X}_p \Rightarrow \overline{Y}_1\overline{Y}_q$. Nous commençons donc par présenter le concept de motifs raisonnablement fréquents.

3.2.1 Motifs Raisonnablement Fréquents

Notre algorithme va se baser sur les motifs raisonnablement fréquents et non plus sur les motifs fréquents comme le fait `Apriori`. Un motif raisonnablement fréquent est un motif fréquent qui possède également un support inférieur ou égal à un seuil maximal que nous nommons max_{sup}. Ce nouveau seuil sera utilisé sur l'ensemble des motifs, c'est-à-dire sur les motifs X, $\overline{X}$, $\overline{X}Y$ et $\overline{X}\,\overline{Y}$.

Définition 13 - Motif raisonnablement fréquent :
Un motif $\mathcal{M} \in \{X, \overline{X}, \overline{X}Y, \overline{X}\,\overline{Y}\}$ *est un motif raisonnablement fréquent si* $min_{sup} \leqslant sup(\mathcal{M}) \leqslant max_{sup}$.

La valeur de max_{sup} est fixée par défaut à $1 - min_{sup}$ mais les utilisateurs sont libres de modifier le seuil à leur convenance en fonction des données à analyser.

En plus des motifs raisonnablement fréquents et des motifs non fréquents, on trouve les motifs omniprésents. Un motif omniprésent est un motif fréquent qui possède un support très élevé et en général proche de 1.

Définition 14 - Motif omniprésent :
Un motif $\mathcal{M} \in \{X, \overline{X}, \overline{X}Y, \overline{X}\,\overline{Y}\}$ *est un motif omniprésent si* $max_{sup} < sup(\mathcal{M}) \leqslant 1$.

Lors de la recherche des motifs fréquents, un motif omniprésent $\mathcal{M}_1$ positif ou négatif ($\mathcal{M}_1 \in \{X, \overline{X}\}$) est combiné avec la majorité des autres motifs fréquents $\mathcal{M}_2$ positif ou négatif ($\mathcal{M}_2 \in \{Y, \overline{Y}\}$), puisque $sup(\mathcal{M}_1\mathcal{M}_2) \simeq sup(\mathcal{M}_2)$ et ceci sans révéler une combinaison $\mathcal{M} = \mathcal{M}_1\mathcal{M}_2$ pertinente. En utilisant les motifs raisonnablement fréquents, nous allons donc éviter de nous retrouver dans cette situation. La suppression des motifs omniprésents va également garantir une diminution du nombre de motifs fréquents, car lorsqu'il existe au moins un motif omniprésent dans la base de données, il est combiné avec pratiquement l'ensemble des autres motifs fréquents. Un exemple d'une telle situation issue du secteur bancaire est le compte courant qui est associé avec la majorité des produits financiers proposés par la banque. En effet, la plupart des clients commencent par ouvrir un compte courant en arrivant dans la banque et c'est pour cette raison que ces combinaisons sont inintéressantes.

Nous allons maintenant prouver que toutes les règles composées d'un motif omniprésent ne sont pas intéressantes. En effet, si le motif $\mathcal{M}_1$ est omniprésent, alors la règle $\mathcal{M}_1 \Rightarrow \mathcal{M}_2$ est inintéressante car elle vérifie rarement le seuil minimum pour la confiance (*voir la preuve 1*) et la règle $\mathcal{M}_2 \Rightarrow \mathcal{M}_1$ ne peut pas non plus être intéressante

puisque même si elle est valide pour la confiance, elle est trop proche de l'indépendance (*voir la preuve 2*). De plus, une règle $XZ \Rightarrow WY$ dont la prémisse est composée d'un motif X omniprésent et où la conclusion est composée d'un motif Y omniprésent n'est pas intéressante car elle est redondante à la règle $Z \Rightarrow W$ (*voir la preuve 3*).

Preuve 1 : si $\mathcal{M}_1$ est omniprésent alors $\mathcal{M}_1 \Rightarrow \mathcal{M}_2$ est inintéressante.
$conf(\mathcal{M}_1 \Rightarrow \mathcal{M}_2) = \frac{sup(\mathcal{M}_1\mathcal{M}_2)}{sup(\mathcal{M}_1)} \simeq \frac{sup(\mathcal{M}_2)}{sup(\mathcal{M}_1)} \ll 1$ puisque le support de $\mathcal{M}_1$ a une valeur élevée.

La règle $\mathcal{M}_1 \Rightarrow \mathcal{M}_2$ est donc non valide pour la confiance.

Preuve 2 : si $\mathcal{M}_1$ est omniprésent alors $\mathcal{M}_2 \Rightarrow \mathcal{M}_1$ est inintéressante.

Bien que la règle $\mathcal{M}_2 \Rightarrow \mathcal{M}_1$ soit valide pour la confiance, elle est trop proche de l'indépendance pour être réellement intéressante. Pour le prouver, nous utilisons la nouveauté [Lavrac et al., 1999] qui évalue l'écart par rapport à l'indépendance. Plus la valeur de la nouveauté est éloignée de 0 et plus la règle sera considérée comme intéressante.

$$\begin{aligned} nouveauté(\mathcal{M}_2 \Rightarrow \mathcal{M}_1) &= sup(\mathcal{M}_1\mathcal{M}_2) - sup(\mathcal{M}_1) \times sup(\mathcal{M}_2) \\ &= sup(\mathcal{M}_2) - sup(\mathcal{M}_1) \times sup(\mathcal{M}_2) \\ &= sup(\mathcal{M}_2)(1 - sup(\mathcal{M}_1)) \simeq 0 \text{ puisque } 1 - sup(\mathcal{M}_1) \simeq 0. \end{aligned}$$

La règle $\mathcal{M}_2 \Rightarrow \mathcal{M}_1$ est donc non valide pour la nouveauté et plus généralement pour toute mesure d'écart à l'indépendance.

Preuve 3 : si X, Y et XY sont des motifs omniprésents alors $XZ \Rightarrow WY$ est inintéressante.

Bien que la règle $XZ \Rightarrow WY$ puisse être valide pour la confiance et pour la nouveauté, la règle est redondante avec la règle $Z \Rightarrow W$ puisqu'elle n'apporte aucune information supplémentaire. En effet si $Z \Rightarrow W$ est valide pour le support et la confiance alors $XZ \Rightarrow WY$ sera également valide puisque : $sup(XZ) = sup(Z)$, $sup(WY) = sup(W)$ et $sup(WXYZ) = sup(WZ)$.

La règle $XZ \Rightarrow WY$ est donc redondante à la règle $Z \Rightarrow W$.

Pour conclure, la recherche des motifs raisonnablement fréquents nous permet d'élaguer des règles inintéressantes. Ceci est d'autant plus intéressant que cette étape intervient au début du processus d'extraction des règles d'association et non pas dans la phase de post-traitement (*cf. étape 6 figure* 1.1). Par ailleurs, il faudra également faire en sorte de ne pas utiliser les motifs omniprésents pour construire les prochains niveaux de motifs raisonnablement fréquents afin d'éliminer certaines règles redondantes.

La prochaine sous-section introduit la mesure d'intérêt M_G que nous ajoutons à l'approche support/confiance. Nous mettons ensuite en avant les règles inintéressantes qu'elle permet d'élaguer.

3.2.2 Mesure d'intérêt M_G

Nous commençons par présenter la mesure M_G [Guillaume, 2010] en expliquant sa sémantique puis mettons en évidence les règles inintéressantes qu'elle permet d'élaguer. Et enfin, nous présentons une propriété intéressante de M_G entre les règles antinomiques $X \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ que nous utilisons pour déduire facilement l'intérêt d'une règle à partir

de sa règle antinomique.

➤ Sémantique

Cette mesure est une amélioration du facteur de certitude [Shortliffe, 1976] et de la mesure M_{GK} [Guillaume, 2000], notamment utilisée dans l'approche de [Wu et al., 2004], puisqu'elle prend en compte le point d'équilibre. L'équilibre intervient quand le nombre d'exemples et de contre-exemples sont égaux, c'est-à-dire lorsqu'il y a autant de chances de voir se réaliser la conclusion Y que sa négation $\overline{Y}$ lorsque la prémisse X est réalisée. Cet état est décrit dans la figure 3.1. Au point d'équilibre, nous avons donc l'égalité suivante : $conf(X \Rightarrow Y) = conf(X \Rightarrow \overline{Y}) = \frac{1}{2}$. L'intérêt de prendre en compte l'équilibre fut mise en évidence par [Blanchard et al., 2005] qui expliquent que le fait d'ignorer cet état peut conduire à extraire des règles inintéressantes, ce que nous allons mettre en évidence dans cette section.

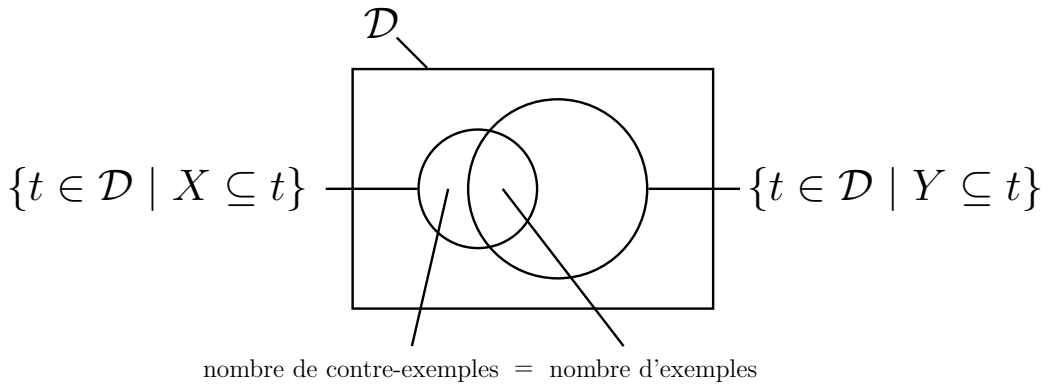

FIGURE 3.1 – Cas représentant l'équilibre entre les motifs X et Y

Dans cette figure, nous rappelons que $\mathcal{D}$ représente l'ensemble des transactions contenues dans la base de données. $\{t \in \mathcal{D} \mid X \subseteq t\}$ et $\{t \in \mathcal{D} \mid Y \subseteq t\}$ sont l'ensemble des transactions qui vérifient respectivement X et Y. Ici, le nombre d'exemples est égal aux nombres de contre-exemples, et par conséquent $sup(XY) = sup(X\overline{Y})$. Il est à noter que dans ce cas précis, nous avons également $sup(XY) = \frac{1}{2}sup(X)$. Il existe donc bien un équilibre entre les motifs X et Y puisque la confiance des règles antinomiques sont égales : $conf(X \Rightarrow Y) = conf(X \Rightarrow \overline{Y}) = \frac{1}{2}$.

La mesure M_G est définie comme suit :

Zone attractive : $conf(X \Rightarrow Y) > max(\frac{1}{2}, sup(Y))$

$$M_G(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - max(\frac{1}{2}, sup(Y))}{1 - max(\frac{1}{2}, sup(Y))}$$

Zone répulsive : $conf(X \Rightarrow Y) < min(\frac{1}{2}, sup(Y))$

$$M_G(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - min(\frac{1}{2}, sup(Y))}{min(\frac{1}{2}, sup(Y))}$$

Zone inintéressante : $min(\frac{1}{2}, sup(Y)) \leqslant conf(X \Rightarrow Y) \leqslant max(\frac{1}{2}, sup(Y))$

$$M_G(X \Rightarrow Y) = 0$$

La mesure M_G commence par déterminer la zone d'appartenance de la règle (*zone attractive, répulsive et inintéressante*). Il est à noter que deux de ces zones (*zone attractive*

et répulsive) ont été mises en évidence par [Guillaume, 2000]. Pour déterminer cette zone, nous devons calculer la confiance de la règle positive $X \Rightarrow Y$ et comparer sa valeur avec $max(\frac{1}{2}, sup(Y))$ et $min(\frac{1}{2}, sup(Y))$. Si la confiance $conf(X \Rightarrow Y)$ de la règle positive $X \Rightarrow Y$ est :

- **supérieure à max($\frac{1}{2}$, sup(Y))**, alors la règle $X \Rightarrow Y$ est dans la zone attractive. La règle se situe donc au-delà de l'indépendance (*puisque* $conf(X \Rightarrow Y) > sup(Y)$), ce qui révèle que la présence de X augmente les chances d'apparition de Y. La règle se situe également au-delà de l'équilibre (*puisque* $conf(X \Rightarrow Y) > \frac{1}{2}$), ce qui révèle que l'on a plus d'exemples que de contre-exemples.
  Dans la zone attractive, la mesure M_G va donc évaluer la distance de la règle $X \Rightarrow Y$ entre l'indépendance ou l'équilibre (*l'état retenu sera celui dont la valeur est la plus élevée*) et l'implication logique. Ainsi, plus la valeur de M_G est proche de 1 et plus la règle est proche de l'implication logique ; et plus la valeur de M_G est proche de 0 et plus la règle sera proche de l'indépendance ou de l'équilibre.

- **inférieure à min($\frac{1}{2}$, sup(Y))**, alors la règle $X \Rightarrow Y$ est dans la zone répulsive. La règle se situe donc en deçà de l'indépendance (*puisque* $conf(X \Rightarrow Y) < sup(Y)$), ce qui révèle que la présence de X diminue les chances d'apparition de Y et augmente les chances d'apparition de $\overline{Y}$. Et la règle se situe également en deçà de l'équilibre (*puisque* $conf(X \Rightarrow Y) < \frac{1}{2}$), ce qui révèle que l'on a plus de contre-exemples que d'exemples. Par conséquent la règle la plus intéressante sera la règle $X \Rightarrow \overline{Y}$.
  Dans la zone répulsive, la mesure M_G va donc évaluer la distance de la règle $X \Rightarrow Y$ entre l'incompatibilité et entre l'indépendance ou l'équilibre (*l'état retenu sera celui dont la valeur est la plus faible*). Ainsi, plus la valeur de M_G est proche de -1 et plus la règle est proche de l'incompatibilité ; et plus la valeur de M_G est proche de 0 et plus la règle sera proche de l'indépendance ou de l'équilibre.

- **supérieure ou égale à min($\frac{1}{2}$, sup(Y)) mais inférieure ou égale à max($\frac{1}{2}$, sup(Y))**, alors la règle se trouve dans la zone inintéressante. Dans cette zone, la confiance de la règle peut se situer dans deux intervalles : $[sup(Y), \frac{1}{2}]$ ou $[\frac{1}{2}, sup(Y)]$ en fonction du point d'indépendance, si celui-ci se situe avant ou après le point d'équilibre.

  Dans l'intervalle $[sup(Y), \frac{1}{2}]$, le point d'indépendance se situe avant le point d'équilibre. Dans ce cas, la règle se situe donc à ou au-delà de l'indépendance (*puisque* $conf(X \Rightarrow Y) \geqslant sup(Y)$), ce qui révèle que la présence de X augmente les chances d'apparition de Y (*si les motifs X et Y ne sont pas indépendants, c'est-à-dire si on n'a pas l'égalité* $conf(X \Rightarrow Y) = sup(Y)$). Par conséquent, la règle la plus intéressante est la règle $X \Rightarrow Y$. Cependant la règle se situe également à ou en deçà de l'équilibre (*puisque* $conf(X \Rightarrow Y) \leqslant \frac{1}{2}$), ce qui révèle dans le cas d'une inégalité stricte que l'on a plus de contre-exemples que d'exemples (*si les motifs XY et $X\overline{Y}$ ne sont pas équilibrés*). Cette fois-ci, la règle la plus intéressante serait donc la règle $X \Rightarrow \overline{Y}$. Dans cette situation, il y a donc contradiction pour trouver la règle la plus intéressante.

  Dans l'intervalle $[\frac{1}{2}, sup(Y)]$, le point d'équilibre se situe avant le point d'indépendance. Dans ce cas, la règle se situe donc à ou au-delà de l'équilibre (*puisque* $conf(X \Rightarrow Y) \geqslant \frac{1}{2}$), ce qui révèle que l'on a plus d'exemples que de contre-exemples dans le cas d'une inégalité stricte. Par conséquent, la règle la plus intéressante est la règle $X \Rightarrow Y$. Cependant la règle se situe également à ou en

deçà de l'indépendance (*puisque* $conf(X \Rightarrow Y) \leqslant sup(Y)$), ce qui révèle que la présence de X diminue les chances d'apparition de Y et augmente les chances d'apparition de $\overline{Y}$. Cette fois-ci, la règle la plus intéressante est la règle $X \Rightarrow \overline{Y}$. Dans cette situation, il y a donc également contradiction pour trouver la règle la plus intéressante.
En conséquence dans cette zone, quelle que soit la situation envisagée, toutes les règles sont jugées inintéressantes et la valeur de la mesure M_G est égale à 0.

La figure 3.2 nous montre la courbe d'évolution de la mesure M_G en fonction de la valeur de la confiance.

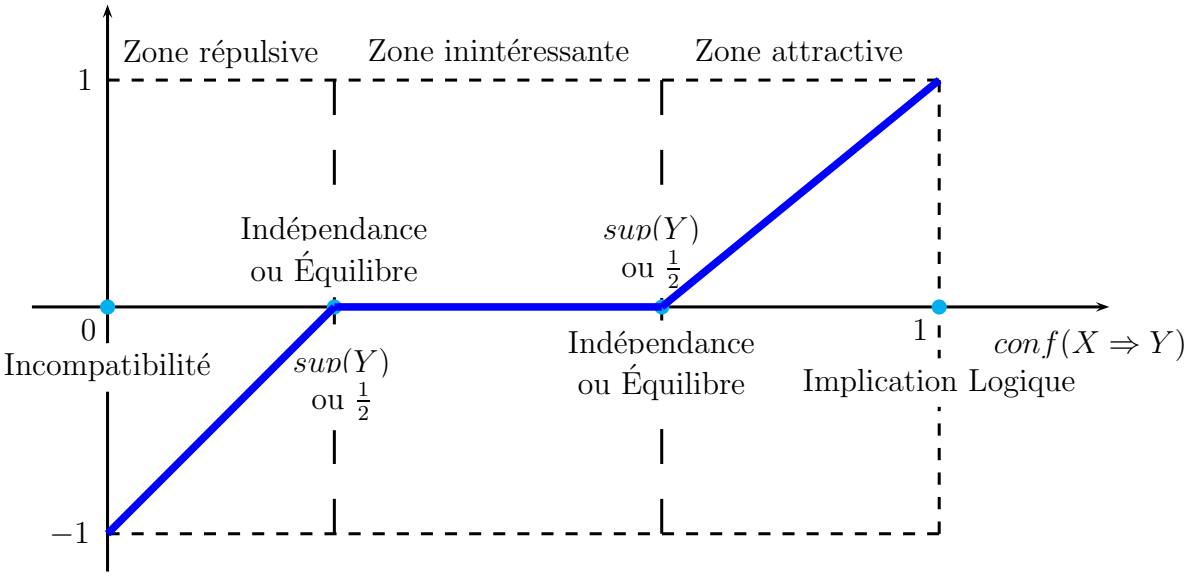

FIGURE 3.2 – Évolution de la mesure M_G

Sur cette figure on peut voir que M_G prend des valeurs négatives dans la zone de répulsion, des valeurs positives dans la zone d'attraction et enfin, une valeur nulle entre l'indépendance et l'équilibre puisque les règles situées dans cette zone sont jugées non intéressantes. Les points d'indépendance et d'équilibre ne sont pas clairement établis puisque la position représentant ces états dépend de la valeur du support de la conclusion par rapport à $\frac{1}{2}$. En effet, si $sup(Y) > \frac{1}{2}$ alors l'indépendance se situe après l'équilibre et si $sup(Y) < \frac{1}{2}$ l'indépendance se situe avant l'équilibre. Dans le cas où $sup(Y) = \frac{1}{2}$ le point d'équilibre et le point d'incompatibilité sont confondus. En conclusion, la mesure M_G va évaluer la distance par rapport à l'implication logique ou par rapport à l'incompatibilité.

Après avoir présenté la sémantique de la mesure M_G, nous allons mettre en évidence son intérêt en exposant les règles qu'elle permet d'éliminer.

➤ Intérêt de la mesure M_G

L'utilisation du support et de la confiance ont deux avantages majeurs. Premièrement, ces deux mesures sont facilement interprétables, ce qui permet une meilleure compréhension des règles extraites et facilite leur exploitation. Deuxièmement, elles possèdent une propriété qui permet d'accélérer les calculs. Cependant, l'utilisation de la confiance peut amener à extraire des règles inintéressantes malgré un seuil relativement

élevé. Prenons deux exemples pour mettre en avant ce problème.

Dans un **premier exemple** repris de la thèse de Laurent Fleury [Fleury, 1996], prenons une base de données décrivant des individus nés en Suède. Nous avons les items suivants : *blond* symbolisé par la lettre B et *parler chinois* symbolisé par la lettre C. Les différentes valeurs pour le support sont $sup(B) = 0,90$, $sup(C) = 0,05$ et $sup(BC) = 0,04$. La règle *parler chinois* $\Rightarrow$ *blond* a un support de 4% et une confiance de 80%. Par conséquent, si l'utilisateur prend $min_{sup} \geqslant 0,04$ et $min_{conf} \geqslant 0,80$ comme seuils d'acceptabilité alors cette règle est valide. Cependant, la prémisse C n'augmente pas les chances d'apparition de la conclusion B puisque sans connaissance préalable, il y a une plus grande probabilité d'obtenir B car sa fréquence est de 90%. Une règle est jugée intéressante si la présence de la prémisse augmente les chances d'apparition de la conclusion, c'est-à-dire si $conf(X \Rightarrow Y) > sup(Y)$. Par conséquent, dans cet exemple, la règle la plus intéressante est *parler chinois* $\Rightarrow \overline{blond}$ même si la confiance est seulement de 20%. En effet, même avec une confiance jugée faible, C augmente les chances d'apparition de $\overline{B}$. Cependant, cette condition n'est pas suffisante puisqu'il existe des cas où la règle $X \Rightarrow Y$ vérifie la condition $conf(X \Rightarrow Y) > sup(Y)$ et la confiance la plus élevée est celle de la règle antinomique $X \Rightarrow \overline{Y}$.

Dans un **second exemple**, prenons une base de données décrivant des individus originaires d'Irlande. Cette fois-ci nous avons les items suivants : *roux* symbolisé par la lettre R et *parler gaélique* symbolisé par la lettre G. Les différentes valeurs pour le support sont $sup(R) = 0,30$, $sup(G) = 0,10$ et $sup(RG) = 0,04$. La règle *parler gaélique* $\Rightarrow$ *roux* est pertinente au regard du critère précédent ($conf(X \Rightarrow Y) > sup(Y)$) puisque la confiance de cette règle est de 40%, ce qui est supérieur au support de la conclusion qui est égale à 30%. Par conséquent, la prémisse G augmente bien les chances d'apparition de la conclusion R. Cependant, la règle antinomique *parler gaélique* $\Rightarrow \overline{roux}$ possède une confiance plus élevée que la règle *parler gaélique* $\Rightarrow$ *roux* puisque $conf(parler\ gaélique \Rightarrow \overline{roux}) = 60\%$.

En conclusion, la règle $X \Rightarrow Y$ est intéressante si et seulement si la réalisation de X augmente les chances d'apparition de Y, (*i.e.* $conf(X \Rightarrow Y) > sup(Y)$), et si la règle antinomique possède une confiance moins élevée, (*i.e.* $conf(X \Rightarrow Y) > conf(X \Rightarrow \overline{Y})$). C'est pour ces deux raisons que la zone attractive de la mesure M_G est définie comme $conf(X \Rightarrow Y) > max(\frac{1}{2}, sup(Y))$. Présentons maintenant une propriété intéressante de cette mesure que nous pourrons exploiter par la suite pour déduire facilement l'intérêt d'une règle à partir de sa règle antinomique.

➤ Propriété intéressante de la mesure M_G

Nous allons simplifier l'écriture de la mesure M_G :
- M_{G_a} correspondra à la mesure M_G dans la zone attractive.
- M_{G_r} correspondra à la mesure M_G dans la zone répulsive.
- M_{G_i} correspondra à la mesure M_G dans la zone inintéressante.

Il existe des relations simples entre les règles antinomiques et qui sont les suivantes :

1. $M_{G_a}(X \Rightarrow \overline{Y}) = -M_{G_r}(X \Rightarrow Y)$,
2. $M_{G_r}(X \Rightarrow \overline{Y}) = -M_{G_a}(X \Rightarrow Y)$,
3. $M_{G_i}(X \Rightarrow \overline{Y}) = -M_{G_i}(X \Rightarrow Y)$.

Preuve 1 : $M_{G_a}(X \Rightarrow \overline{Y}) = -M_{G_r}(X \Rightarrow Y)$

$$\begin{aligned} M_{G_a}(X \Rightarrow \overline{Y}) &= \frac{conf(X \Rightarrow \overline{Y}) - max(\frac{1}{2},\ sup(\overline{Y}))}{1 - max(\frac{1}{2},\ sup(\overline{Y}))} \\ &= \frac{1 - conf(X \Rightarrow Y) - max(\frac{1}{2},\ 1 - sup(Y))}{1 - max(\frac{1}{2},\ 1 - sup(Y))} \\ &= -\frac{conf(X \Rightarrow Y) - min(\frac{1}{2},\ sup(Y))}{min(\frac{1}{2},\ sup(Y))} \\ &= -M_{G_r}(X \Rightarrow Y) \end{aligned}$$

Preuve 2 : $M_{G_r}(X \Rightarrow \overline{Y}) = -M_{G_a}(X \Rightarrow Y)$

$$\begin{aligned} M_{G_r}(X \Rightarrow \overline{Y}) &= \frac{conf(X \Rightarrow \overline{Y}) - min(\frac{1}{2},\ sup(\overline{Y}))}{min(\frac{1}{2},\ sup(\overline{Y}))} \\ &= \frac{1 - conf(X \Rightarrow Y) - min(\frac{1}{2},\ 1 - sup(Y))}{min(\frac{1}{2},\ 1 - sup(Y))} \\ &= -\frac{conf(X \Rightarrow Y) - max(\frac{1}{2},\ sup(Y))}{1 - max(\frac{1}{2},\ sup(Y))} \\ &= -M_{G_a}(X \Rightarrow Y) \end{aligned}$$

Preuve 3 : $M_{G_i}(X \Rightarrow \overline{Y}) = -M_{G_i}(X \Rightarrow Y)$

$$\begin{aligned} & min(\tfrac{1}{2},\ sup(\overline{Y})) \leqslant conf(X \Rightarrow \overline{Y}) \leqslant max(\tfrac{1}{2},\ sup(\overline{Y})) \\ \Leftrightarrow\ & min(\tfrac{1}{2},\ 1 - sup(Y)) \leqslant 1 - conf(X \Rightarrow Y) \leqslant max(\tfrac{1}{2},\ 1 - sup(Y)) \\ \Leftrightarrow\ & 1 - max(\tfrac{1}{2},\ sup(Y)) \leqslant 1 - conf(X \Rightarrow Y) \leqslant 1 - min(\tfrac{1}{2},\ sup(Y)) \\ \Leftrightarrow\ & min(\tfrac{1}{2},\ sup(Y)) \leqslant conf(X \Rightarrow Y) \leqslant max(\tfrac{1}{2},\ sup(Y)) \end{aligned}$$

Donc si la règle $X \Rightarrow \overline{Y}$ est dans la zone inintéressante alors la règle $X \Rightarrow Y$ l'est aussi. Par souci d'harmonisation avec les autres relations, et comme la valeur de M_{G_i} vaut 0, nous mettrons le $-$, par conséquent nous obtenons bien $M_{G_i}(X \Rightarrow \overline{Y}) = -M_{G_i}(X \Rightarrow Y)$.

Ainsi, si la règle $X \Rightarrow Y$ a une valeur pour la mesure M_G strictement négative, nous savons que c'est la règle $X \Rightarrow \overline{Y}$ qui sera intéressante et nous n'aurons pas besoin de calculer sa valeur puisque nous pourrons la déduire facilement. De plus, la zone répulsive possède les mêmes propriétés que la zone attractive : la règle $X \Rightarrow \overline{Y}$ est intéressante si la réalisation de X favorise les chances d'apparition de la conclusion $\overline{Y}$, c'est-à-dire si $conf(X \Rightarrow \overline{Y}) > sup(\overline{Y})$, et si la règle antinomique possède une confiance moins élevée, c'est-à-dire si $conf(X \Rightarrow \overline{Y}) > conf(X \Rightarrow Y)$. C'est pour ces raisons que la zone répulsive de la mesure M_G est définie par l'inégalité $conf(X \Rightarrow Y) < min(\frac{1}{2}, sup(Y))$. Après avoir mis en évidence l'intérêt de la mesure M_G, nous présentons le nouveau type de règles que nous allons rechercher $\overline{x_1}..\overline{x_p} \Rightarrow \overline{y_1}..\overline{y_q}$ puis indiquons l'intérêt de le prendre en compte.

3.2.3 Extension aux règles du type $\overline{x_1}..\overline{x_p} \Rightarrow \overline{y_1}..\overline{y_q}$

Comme nous l'avons vu dans la définition 11 du chapitre 1, une règle d'association négative est une règle qui possède au moins un item négatif dans la prémisse et/ou dans

la conclusion. Or, comme nous l'avons dit précédemment, afin de limiter l'explosion combinatoire, la majorité des approches a tendance à considérer l'ensemble des items composant les prémisses et les conclusions comme un unique motif soit entièrement positif soit entièrement négatif. Cependant la définition implique qu'il existe des règles qui possèdent plusieurs motifs négatifs dans la prémisse et/ou dans la conclusion. Nous allons donc essayer d'étendre les approches traditionnelles pour rechercher des règles possédant plusieurs items négatifs dans la prémisse et dans la conclusion, c'est-à-dire des règles de la forme $\overline{x_1}..\overline{x_p} \Rightarrow \overline{y_1}..\overline{y_q}$. Cet apport est un premier pas vers la généralisation pour rechercher des règles du type $(X_1 \wedge X_2) \vee X_3 \Rightarrow Y_1 \wedge (Y_2 \vee Y_3)$, c'est-à-dire des règles possédant en prémisse et/ou en conclusion des conjonctions ou disjonctions de motifs positifs ou négatifs.

Lors de la recherche des motifs raisonnablement fréquents, nous allons ajouter une condition pour rechercher les conjonctions de motifs négatifs que nous notons $\ddot{X}$. Cette contrainte additionnelle va renforcer notre souhait d'extraire les règles les plus intéressantes possibles. En effet, la contrainte supplémentaire $sup(\ddot{X}) \geqslant min_{s\ddot{u}p}$ impose, comme la contrainte du support maximum pour les motifs raisonnablement fréquents ($sup(X) \leqslant max_{sup}$), que les motifs X ne soient pas omniprésents.

Pour un seuil identique (*i.e.* $min_{sup} = min_{s\ddot{u}p}$ *et* $max_{sup} = 1 - min_{sup}$), cette nouvelle contrainte est plus restrictive puisque si nous avons : $sup(\ddot{X}) \geqslant min_{s\ddot{u}p}$ alors $sup(X)$ est plus largement inférieur à max_{sup}.

Preuve :

$$\begin{aligned} sup(X) \leqslant max_{sup} &\Leftrightarrow 1 - sup(\overline{X}) \leqslant max_{sup} \\ &\Leftrightarrow sup(\overline{X}) \geqslant 1 - max_{sup} \\ &\Leftrightarrow sup(\overline{X}) \geqslant min_{sup} \end{aligned}$$

Comme $sup(\ddot{X}) \leqslant sup(\overline{X})$ et dans ce cas particulier où $min_{sup} = min_{s\ddot{u}p}$, la contrainte $sup(\ddot{X}) \geqslant min_{s\ddot{u}p}$ prouve que le niveau d'exigence en matière de recherche de motifs non omniprésents est plus important.

De plus, cette contrainte va permettre de générer uniquement les motifs XY où X et Y sont relativement bien corrélés puisque XY et $\ddot{X}Y$ doivent être fréquents. Pour justifier nos propos, regardons la figure 3.3 qui représente le diagramme de Venn d'une règle $X \Rightarrow Y$ dans deux contextes différents.

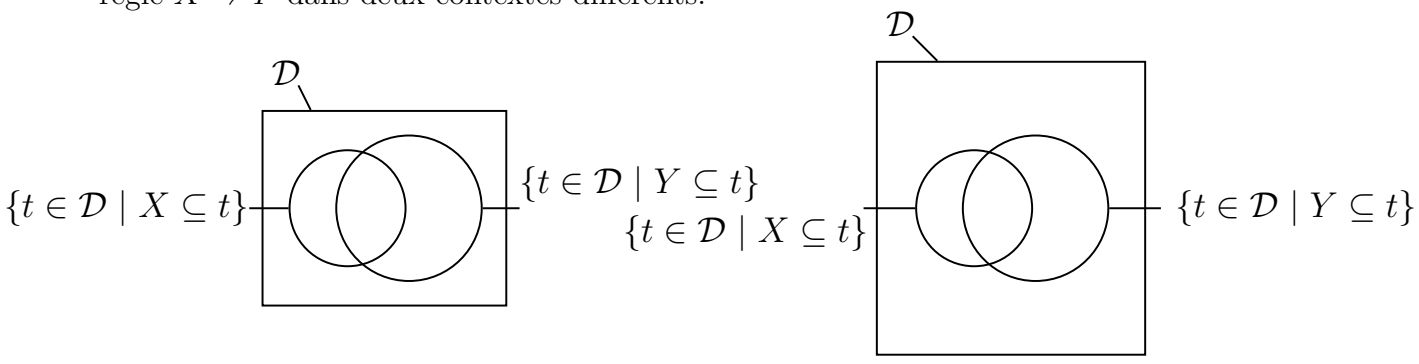

FIGURE 3.3 – Diagrammes de Venn d'une règle $X \Rightarrow Y$ dans deux contextes différents

La contingence des ensembles $\{t \in \mathcal{D} \mid X \subseteq t\}$, $\{t \in \mathcal{D} \mid Y \subseteq t\}$ et $\{t \in \mathcal{D} \mid XY \subseteq t\}$ est la même pour les deux diagrammes de Venn. La seule différence est la contingence

de l'ensemble $\{t \in \mathcal{D} \mid \overline{X}\ \overline{Y} \subseteq t\}$ qui est plus faible pour la figure de gauche. Comme la contingence de $\{t \in \mathcal{D} \mid X \subseteq t\}$ et $\{t \in \mathcal{D} \mid XY \subseteq t\}$ est la même pour les deux figures, la confiance de la règle $X \Rightarrow Y$ est identique dans les deux cas. Cependant, la règle associée à la figure de droite est plus intéressante que celle de la figure de gauche puisque la probabilité d'obtenir cette intersection entre $\{t \in \mathcal{D} \mid X \subseteq t\}$ et $\{t \in \mathcal{D} \mid Y \subseteq t\}$ est plus faible pour le diagramme de droite. La confiance ne peut distinguer ces deux cas mais en ajoutant une nouvelle contrainte pour les motifs $\ddot{X}$, cela permet d'élaguer certaines règles inintéressantes du type de celle de la figure de gauche. Par ailleurs, nous n'ajoutons pas de seuil maximum pour ce nouveau type de motifs $\ddot{X}$ puisque cet ajout serait partiellement redondant avec le support minimum pour les motifs positifs.

Prenons la règle *croissant* $\Rightarrow$ *pain au chocolat* dans une boulangerie de 100 clients et dans un supermarché de 1000 clients pour mettre en avant ce problème. Fixons le seuil du support à $0,03$ et celui de la confiance à 0.5. Supposons que 60 clients achètent des croissants, 50 des pains au chocolat et 30 achètent les deux produits simultanément dans les deux magasins alors tous ces motifs sont fréquents. La confiance de la règle *croissant* $\Rightarrow$ *pain au chocolat* est la même pour les deux magasins et est valide puisque la confiance de cette règle est de $0,5$ dans les deux magasins. Cependant, la probabilité d'obtenir un panier contenant des croissants et des pains au chocolat est plus faible dans le supermarché que dans la boulangerie puisque le nombre de clients du supermarché (*1000*) est significativement plus grand que dans une boulangerie (*100*). Fixons le seuil du support des conjonctions de motifs négatifs $min_{s\ddot{u}p}$ à $0,3$ puis vérifions le support de la règle *croissant* $\Rightarrow$ *pain au chocolat* associée aux deux magasins.

Pour calculer ce support, on peut adapter la formule du crible de Poincaré attribué à Abraham de Moivre qui permet de calculer le cardinal de la réunion de n ensembles. Notre formule permettant de calculer $min_{s\ddot{u}p}$ est définie comme suit :

Soient X_1, ..., X_n n motifs. Nous avons :

$$sup(\bigcup_{i=1}^{n} \overline{X_i}) = 1 - \left(\sum_{i=1}^{n} sup(X_i) - \sum_{(i,j)|1 \leqslant i < j \leqslant n} sup(X_i X_j) + \sum_{(i,j,k)|1 \leqslant i < j < k \leqslant n} sup(X_i X_j X_k) - ... + (-1)^{n+1} sup(X_1 ... X_n) \right)$$

Cette formule peut s'écrire de façon plus condensée :

$$sup(\bigcup_{i=1}^{n} \overline{X_i}) = 1 - \left(\sum_{k=1}^{n} \left((-1)^{k-1} \sum_{1 \leqslant i_1 < i_2 < ... < i_k \leqslant n} sup(X_{i_1} X_{i_2} ... X_{i_k}) \right) \right)$$

Si nous appliquons cette formule à notre cas :

$sup(\overline{croissant},\ \overline{pain\ au\ chocolat}) = 1 - sup(croissant) - sup(pain\ au\ chocolat) + sup(croissant,\ pain\ au\ chocolat)$

Pour la boulangerie :

$sup(\overline{croissant},\ \overline{pain\ au\ chocolat}) = 1 - \frac{60}{100} - \frac{50}{100} + \frac{30}{100} = 1 - 0,6 - 0,5 + 0,3 = 0,2$

Pour le supermarché :

$$sup(\overline{croissant},\ \overline{pain\ au\ chocolat}) = 1 - \frac{60}{1000} - \frac{50}{1000} + \frac{30}{1000} = 1 - 0,06 - 0,05 + 0,03 = 0,92$$

Avec le support min_{sup} fixé à $0,3$, la règle *croissant* $\Rightarrow$ *pain au chocolat* attachée à la boulangerie n'est pas valide alors qu'elle l'est pour le supermarché. La règle *croissant* $\Rightarrow$ *pain au chocolat* attachée au supermarché est plus intéressante que celle de la boulangerie puisque la probabilité d'avoir un panier composé de croissants et de pains au chocolat est plus faible dans un supermarché que dans une boulangerie. En effet les clients du supermarché ne viennent pas spécifiquement pour acheter ces articles, ce qui peut être le cas pour une boulangerie.

La prochaine section présente les optimisations apportées au parcours de recherche des règles.

3.3 Optimisation du parcours de recherche des règles

Dans la première partie de cette section, nous montrons qu'il est inutile d'étudier l'ensemble des règles puisque, seule une moitié des règles est intéressante à rechercher. Dans la seconde partie, nous limitons encore le nombre de règles à générer en utilisant des méta-règles basées sur la mesure M_G. Ces méta-règles vont permettre l'élagage de certaines règles.

3.3.1 Étude de la moitié des règles

Dans les différents travaux similaires étudiés dans cette thèse (*cf. chapitre* 2), aucune stratégie d'élagage n'est utilisée pour extraire les règles négatives hormis le coefficient de corrélation dans [Antonie and Zaïane, 2004]. En effet, cette dernière permet de savoir quelles règles étudier en fonction de sa valeur. La seule autre stratégie d'élagage existante est la propriété de la confiance (*cf. propriété* 1), mais elle ne peut être appliquée que pour les règles positives. Cependant, il est possible de restreindre et de diviser par deux le nombre de règles à étudier. En effet, nous pouvons diviser notre espace de recherche en répondant à la question suivante : la réalisation de la prémisse X augmente-t-elle les chances d'apparition de la conclusion Y ? En d'autres mots, nous devons vérifier que la confiance de la règle $X \Rightarrow Y$ est supérieure au support de la conclusion Y. Nous allons donc étudier l'incidence d'une réponse positive à cette question sur trois types de règles ($X \Rightarrow \overline{Y}$, $\overline{Y} \Rightarrow \overline{X}$ *et* $Y \Rightarrow X$). Les autres règles ($\overline{Y} \Rightarrow X$, $\overline{X} \Rightarrow \overline{Y}$, $\overline{X} \Rightarrow Y$ *et* $Y \Rightarrow \overline{X}$) peuvent être déduites des précédentes règles d'après le lien existant entre les règles symétriques $X \Rightarrow Y$ et $Y \Rightarrow X$ que nous allons étudier dans cette section.

➤ 1) Lien entre les règles antinomiques $X \Rightarrow Y$ et $X \Rightarrow \overline{Y}$

Il existe un lien entre les règles antinomiques $X \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. En effet, si la réalisation de X augmente les chances d'apparition de Y (*i.e.* $conf(X \Rightarrow Y) > sup(Y)$) alors la réalisation de X diminue les chances d'apparition de $\overline{Y}$ (*i.e.* $conf(X \Rightarrow \overline{Y}) < sup(\overline{Y})$).

Preuve :

$$\begin{aligned} conf(X \Rightarrow Y) > sup(Y) &\Leftrightarrow conf(X \Rightarrow Y) > 1 - sup(\overline{Y}) \\ &\Leftrightarrow 1 - conf(X \Rightarrow \overline{Y}) > 1 - sup(\overline{Y}) \\ &\Leftrightarrow conf(X \Rightarrow \overline{Y}) < sup(\overline{Y}) \end{aligned}$$

➤ 2) Lien entre les règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$

Nous trouvons également un lien entre les règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$. Ce lien est le suivant : si la réalisation de X augmente les chances d'apparition de Y (*i.e.* $conf(X \Rightarrow Y) > sup(Y)$) alors la réalisation de $\overline{Y}$ augmente les chances d'apparition de $\overline{X}$ (*i.e.* $conf(\overline{Y} \Rightarrow \overline{X}) > sup(\overline{X})$).

Preuve :

$$\begin{aligned} conf(X \Rightarrow Y) > sup(Y) &\Leftrightarrow sup(XY) > sup(X)sup(Y) \\ &\Leftrightarrow 1 - sup(X) - sup(Y) + sup(XY) \\ &\qquad > 1 - sup(X) - sup(Y) + sup(X)sup(Y) \\ &\Leftrightarrow 1 - sup(X \vee Y) > (1 - sup(X))(1 - sup(Y)) \\ &\Leftrightarrow sup(\overline{X \vee Y}) > sup(\overline{X})sup(\overline{Y}) \\ &\Leftrightarrow \frac{sup(\overline{X}\,\overline{Y})}{sup(\overline{Y})} > sup(\overline{X}) \\ &\Leftrightarrow conf(\overline{Y} \Rightarrow \overline{X}) > sup(\overline{X}) \end{aligned}$$

➤ 3) Lien entre les règles symétriques $X \Rightarrow Y$ et $Y \Rightarrow X$

Pour terminer, un dernier lien existe entre les règles symétriques $X \Rightarrow Y$ et $Y \Rightarrow X$. En effet, si la réalisation de X augmente les chances d'apparition de Y (*i.e.* $conf(X \Rightarrow Y) > sup(Y)$) alors la réalisation de Y augmente les chances d'apparition de X (*i.e.* $conf(Y \Rightarrow X) > sup(X)$).

Preuve :

$$\begin{aligned} conf(X \Rightarrow Y) > sup(Y) &\Leftrightarrow \frac{sup(XY)}{sup(X)} > sup(Y) \\ &\Leftrightarrow \frac{sup(XY)}{sup(Y)} > sup(X) \\ &\Leftrightarrow conf(Y \Rightarrow X) > sup(X) \end{aligned}$$

Si la réponse à la question précédente est affirmative, c'est-à-dire si la réalisation de la prémisse X augmente les chances d'apparition de la conclusion Y et que par conséquent $conf(X \Rightarrow Y) > sup(Y)$, alors la règle $X \Rightarrow Y$ est potentiellement intéressante [Piatetsky-Shapiro, 1991]. En utilisant ces trois liens, nous pouvons donc déduire les trois propriétés suivantes :

1. si la règle $X \Rightarrow Y$ est potentiellement intéressante alors les règles $Y \Rightarrow X$ (*cf. lien 3*), $\overline{Y} \Rightarrow \overline{X}$ (*cf. lien 2*) et $\overline{X} \Rightarrow \overline{Y}$ (*cf. liens 2 puis 3*) le sont également,

   Preuve : si la règle $X \Rightarrow Y$ est potentiellement intéressante alors $\overline{X} \Rightarrow \overline{Y}$ l'est également

$$
\begin{aligned}
conf(X \Rightarrow Y) > sup(Y) &\Leftrightarrow conf(\overline{Y} \Rightarrow \overline{X}) > sup(\overline{X}) \ (cf.\ lien\ 2) \\
&\Leftrightarrow \frac{sup(\overline{X}\,\overline{Y})}{sup(\overline{Y})} > sup(\overline{X}) \\
&\Leftrightarrow \frac{sup(\overline{X}\,\overline{Y})}{sup(\overline{X})} > sup(\overline{Y}) \\
&\Leftrightarrow conf(\overline{X} \Rightarrow \overline{Y}) > sup(\overline{Y})
\end{aligned}
$$

2. si la règle $X \Rightarrow \overline{Y}$ est potentiellement intéressante alors les règles $\overline{Y} \Rightarrow X$ (*cf. lien 3*), $Y \Rightarrow \overline{X}$ (*cf. lien 2*) et $\overline{X} \Rightarrow Y$ (*cf. liens 2 puis 3*) le sont également,
   Preuve : si la règle $X \Rightarrow \overline{Y}$ est potentiellement intéressante alors $\overline{X} \Rightarrow Y$ l'est également

$$
\begin{aligned}
conf(X \Rightarrow \overline{Y}) > sup(\overline{Y}) &\Leftrightarrow conf(Y \Rightarrow \overline{X}) > sup(\overline{X}) \ (cf.\ lien\ 2) \\
&\Leftrightarrow \frac{sup(\overline{X}Y)}{sup(Y)} > sup(\overline{X}) \\
&\Leftrightarrow \frac{sup(\overline{X}Y)}{sup(\overline{X})} > sup(Y) \\
&\Leftrightarrow conf(\overline{X} \Rightarrow Y) > sup(Y)
\end{aligned}
$$

3. si la règle $X \Rightarrow Y$ est potentiellement intéressante alors la règle $X \Rightarrow \overline{Y}$ n'est pas potentiellement intéressante (*cf. lien 1*).

La figure 3.4 montre les règles potentiellement intéressantes et les règles inintéressantes en fonction de l'intérêt de la règle positive.

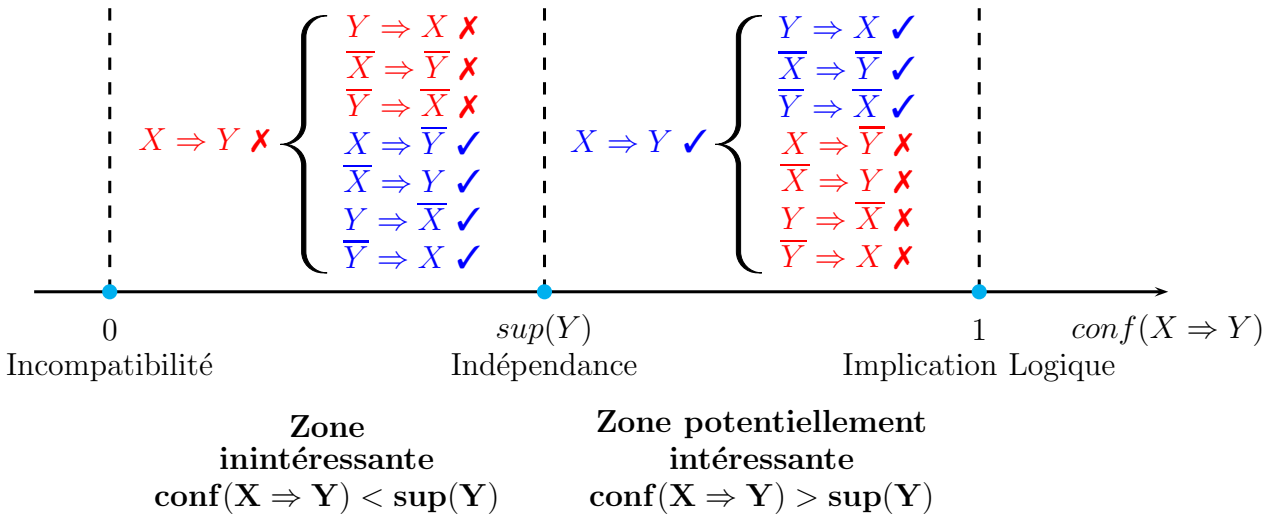

FIGURE 3.4 – Règles potentiellement intéressantes

Dans cette figure, l'axe des abscisses correspond à la confiance de la règle. Trois des quatre états caractéristiques d'une règle sont présents :

— l'incompatibilité, cas où la confiance de la règle est égale à 0,
— l'indépendance, cas où la confiance est égale au support de la conséquence (*i.e.* $sup(Y)$),
— et l'implication logique, cas où la confiance est égale à 1.

Deux zones sont apparentes sur la figure.

La première zone, qui est la zone potentiellement intéressante, est la zone où la présence de X augmente les chances d'apparition de Y (*i.e.* $conf(X \Rightarrow Y) > sup(Y)$). Cette zone

s'étend donc de l'indépendance (*i.e.* $conf(X \Rightarrow Y) = sup(Y)$) à l'implication logique (*i.e.* $conf(X \Rightarrow Y) = 1$). Si la règle $X \Rightarrow Y$ est dans la zone potentiellement intéressante, alors nous allons focaliser notre étude sur les quatre règles suivantes : $X \Rightarrow Y$, $Y \Rightarrow X$, $\overline{X} \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow \overline{X}$ en sachant que les quatre autres règles $X \Rightarrow \overline{Y}$, $\overline{Y} \Rightarrow X$, $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$ ne seront pas potentiellement intéressantes.

La seconde zone, qui est la zone inintéressante, est la zone où la présence de X diminue les chances d'apparition de Y (*i.e.* $conf(X \Rightarrow Y) < sup(Y)$). Cette zone s'étend donc de l'incompatibilité (*i.e.* $conf(X \Rightarrow Y) = 0$) à l'indépendance (*i.e.* $conf(X \Rightarrow Y) = sup(Y)$). Si la règle $X \Rightarrow Y$ est dans la zone inintéressante, alors nous allons focaliser notre étude sur les quatre règles suivantes : $X \Rightarrow \overline{Y}$, $\overline{Y} \Rightarrow X$, $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$ en sachant que les quatre autres règles $X \Rightarrow Y$, $Y \Rightarrow X$, $\overline{X} \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow \overline{X}$ ne seront pas potentiellement intéressantes.

Dans le dernier cas, c'est-à-dire lorsque la confiance de la règle positive est égale au support de la conséquence (*i.e.* $conf(X \Rightarrow Y) = sup(Y)$), aucune règle intéressante ne peut être extraite puisque cela correspond à l'indépendance entre X et Y mais également entre $\overline{X}$ et Y, X et $\overline{Y}$ et pour finir entre $\overline{X}$ et $\overline{Y}$.

Preuve : X et Y sont indépendants $\Leftrightarrow$ X et $\overline{Y}$ sont indépendants

$$\begin{aligned} P(X\overline{Y}) &= P(X) - P(XY) \\ &= P(X) - P(X)P(Y) \ (car\ X\ et\ Y\ sont\ indépendants) \\ &= P(X) \times (1 - P(Y)) \\ &= P(X) \times P(\overline{Y}) \end{aligned}$$

Preuve : X et Y sont indépendants $\Leftrightarrow$ $\overline{X}$ et Y sont indépendants

$$\begin{aligned} P(\overline{X}Y) &= P(Y) - P(XY) \\ &= P(Y) - P(X)P(Y) \ (car\ X\ et\ Y\ sont\ indépendants) \\ &= P(Y) \times (1 - P(X)) \\ &= P(Y) \times P(\overline{X}) \end{aligned}$$

Preuve : X et Y sont indépendants $\Leftrightarrow$ $\overline{X}$ et $\overline{Y}$ sont indépendants

$$\begin{aligned} P(\overline{X}\,\overline{Y}) &= 1 - P(X) - P(Y) + P(XY) \\ &= 1 - P(X) - P(Y) + P(X)P(Y) \ (car\ X\ et\ Y\ sont\ indépendants) \\ &= (1 - P(X)) \times (1 - P(Y)) \\ &= P(\overline{X}) \times P(\overline{Y}) \end{aligned}$$

En conclusion, l'étude de la règle positive $X \Rightarrow Y$ va permettre d'orienter notre étude et de diviser le nombre de règles à étudier par deux. Les résultats obtenus sont synthétisés dans le tableau 3.1.

Nous venons de montrer que l'étude de la moitié des règles est suffisante. Par ailleurs, nous constatons une certaine similarité entre la figure 3.4, qui montre les règles potentiellement intéressantes et les règles inintéressantes en fonction de l'intérêt de la règle positive, et la courbe d'évolution de la mesure M_G (*cf. figure* 3.2). La différence entre les deux figures provient de la prise en compte du point d'équilibre par la mesure M_G. La zone attractive définie par la mesure M_G correspond à la zone potentiellement intéressante que nous ve-

$conf(X \Rightarrow Y) < sup(Y)$	$conf(X \Rightarrow Y) > sup(Y)$
$X \Rightarrow \overline{Y}$	$X \Rightarrow Y$
$\overline{Y} \Rightarrow X$	$Y \Rightarrow X$
$\overline{X} \Rightarrow Y$	$\overline{X} \Rightarrow \overline{Y}$
$Y \Rightarrow \overline{X}$	$\overline{Y} \Rightarrow \overline{X}$

TABLEAU 3.1 – Ensemble des règles à étudier en fonction de la confiance de la règle positive par rapport au support de la conclusion

nons de définir ici à laquelle nous ajoutons la prise en compte du point d'équilibre. De même pour la zone répulsive définie par la mesure M_G qui correspond à la zone potentiellement inintéressante à laquelle nous ajoutons la prise en compte du point d'équilibre. Par conséquent, lorsque la règle positive $X \Rightarrow Y$ sera dans la zone attractive, nous étudierons les règles $X \Rightarrow Y$, $Y \Rightarrow X$, $\overline{X} \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow \overline{X}$. Lorsque la règle positive $X \Rightarrow Y$ sera dans la zone répulsive, nous étudierons les règles $X \Rightarrow \overline{Y}$, $\overline{Y} \Rightarrow X$, $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$. Dans la prochaine sous-section nous continuons d'élaguer certaines règles de l'étude en utilisant des méta-règles reposant sur la mesure M_G. Ces travaux sur les méta-règles ont été publiés dans [Guillaume and Papon, 2012].

3.3.2 Stratégie d'élagage

Pour continuer l'optimisation, nous souhaitons utiliser des méta-règles pour déduire l'intérêt de certaines règles à partir de l'intérêt de la règle positive $X \Rightarrow Y$. Plus précisément, nous voulons connaître sous quelles conditions nous pouvons exclure certaines règles de l'étude. Nous commençons tout d'abord par déduire l'intérêt des règles symétriques $Y \Rightarrow X$ à partir des règles $X \Rightarrow Y$.

➤ Méta-règles pour en déduire l'intérêt des règles symétriques $Y \Rightarrow X$

Définissons tout d'abord ce que l'on entend par règle non pertinente.

Définition 15 - Règle non pertinente :
Une règle $X \Rightarrow Y$ est jugée non pertinente si elle n'est pas dans la zone attractive (cf. figure 3.2) (i.e. $conf(X \Rightarrow Y) \leqslant max(\frac{1}{2},\ sup(Y))$) ou si elle est dans la zone attractive (i.e. $conf(X \Rightarrow Y) > max(\frac{1}{2},\ sup(Y))$) mais pas assez proche de l'implication logique (i.e. $conf(X \Rightarrow Y) < seuil$).

Nous allons tout naturellement utiliser la mesure M_G pour cette étude car c'est la seule mesure qui travaille sur les zones mises en évidence dans la figure 3.2. Le seuil utilisé pour vérifier la pertinence sera donc min_{M_G}. Passons maintenant à l'étude de l'intérêt des règles symétriques.

Si le support de X est inférieur au support de Y (*i.e.* $sup(X) < sup(Y)$), alors la confiance de la règle $X \Rightarrow Y$ est supérieure à celle de la règle $Y \Rightarrow X$ (*i.e.* $conf(X \Rightarrow Y) > conf(Y \Rightarrow X)$).

Preuve :

$$\begin{aligned} sup(X) < sup(Y) &\Leftrightarrow \frac{1}{sup(X)} > \frac{1}{sup(Y)} \\ &\Leftrightarrow \frac{sup(XY)}{sup(X)} > \frac{sup(XY)}{sup(Y)} \\ &\Leftrightarrow conf(X \Rightarrow Y) > conf(Y \Rightarrow X) \end{aligned}$$

Par conséquent, si $conf(X \Rightarrow Y) \leqslant max(\frac{1}{2},\ sup(Y))$ alors on aura également $conf(Y \Rightarrow X) \leqslant max(\frac{1}{2},\ sup(Y))$ puisque $conf(Y \Rightarrow X) < conf(X \Rightarrow Y)$. En conclusion, si la règle $X \Rightarrow Y$ est jugée non pertinente, alors nous pouvons déduire que la règle $Y \Rightarrow X$ sera également jugée non pertinente puisque sa distance à l'implication logique est plus grande que celle de la règle $X \Rightarrow Y$.

Nous pouvons donc en déduire la méta-règle suivante :

> **MR$_1$** : $\forall X \Rightarrow Y$ avec $sup(X) < sup(Y)$
> si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(Y \Rightarrow X) < min_{M_G}$.

À partir de cette première méta-règle, nous pouvons déduire la contraposée suivante :

> **MRC$_1$** : $\forall X \Rightarrow Y$ avec $sup(X) \geqslant sup(Y)$
> si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(Y \Rightarrow X) \geqslant min_{M_G}$.

La seconde étude a pour objectif de déduire l'intérêt de la règle $\overline{Y} \Rightarrow \overline{X}$ à partir de l'intérêt de la règle $X \Rightarrow Y$.

➤ Méta-règles pour en déduire l'intérêt des règles $\overline{Y} \Rightarrow \overline{X}$

Pour trouver des méta-règles déduisant l'intérêt des règles $\overline{Y} \Rightarrow \overline{X}$ à partir de l'intérêt de la règle $X \Rightarrow Y$, nous devons étudier trois cas différents qui dépendent des valeurs des supports de la prémisse et de la conclusion par rapport à $\frac{1}{2}$, cette dernière valeur permettant de repérer le point d'équilibre ($conf(X \Rightarrow Y) = \frac{1}{2}$). Il faut donc étudier si $sup(X)$ et $sup(Y)$ sont supérieurs ou inférieurs à $\frac{1}{2}$. Ce positionnement permet de savoir si l'indépendance pour les règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$ intervient avant ou après le point d'équilibre. En effet, nous rappelons que si $sup(Y) < \frac{1}{2}$ alors nous pouvons en déduire que l'indépendance pour la règle $X \Rightarrow Y$ (*cas où* $conf(X \Rightarrow Y) = sup(Y)$) intervient avant le point d'équilibre (*cas où* $conf(X \Rightarrow Y) = \frac{1}{2}$). Dans le cas contraire, si $sup(Y) > \frac{1}{2}$ alors l'indépendance intervient après l'équilibre.

Remarquons tout d'abord que dans le cas de l'implication logique pour la règle $X \Rightarrow Y$, l'ensemble $\{t \in \mathcal{D} \mid X \subseteq t\}$ des individus vérifiant la prémisse X, est inclus dans l'ensemble $\{t \in \mathcal{D} \mid Y \subseteq t\}$ des individus vérifiant la conclusion Y. Dans ce cas là, nous pouvons déduire que l'ensemble $\{t \in \mathcal{D} \mid \overline{Y} \subseteq t\}$ des individus ne vérifiant pas Y est inclus dans l'ensemble $\{t \in \mathcal{D} \mid \overline{X} \subseteq t\}$ des individus ne vérifiant pas X. Ainsi, à l'implication logique de la règle $X \Rightarrow Y$ correspond également l'implication logique de la règle $\overline{Y} \Rightarrow \overline{X}$. Par conséquent, lorsque $X \Rightarrow Y$ est une règle logique (*cf. définition* 12), la règle $\overline{Y} \Rightarrow \overline{X}$ l'est également.

☛ **Cas 1 :** $sup(X) < sup(Y) < \frac{1}{2}$

Dans ce cas, l'indépendance pour la règle $X \Rightarrow Y$ intervient avant le point d'équilibre ($sup(Y) < \frac{1}{2}$) et l'indépendance pour la règle $\overline{Y} \Rightarrow \overline{X}$ intervient après le point d'équilibre ($sup(\overline{X}) > \frac{1}{2}$). La figure 3.5 restitue les courbes d'évolution de la mesure M_G dans la zone attractive pour les règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$.

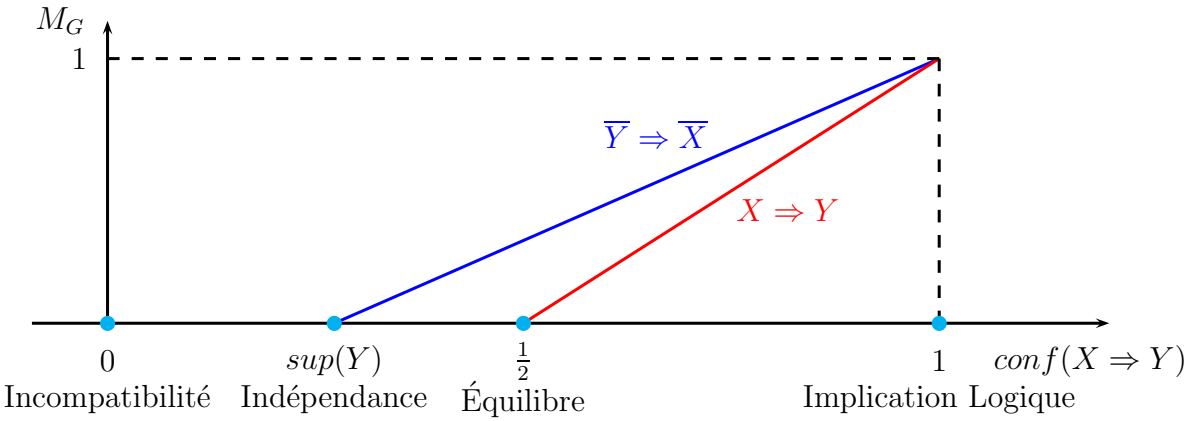

FIGURE 3.5 – Courbe de M_G pour $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$ dans le cas 1

Comme la courbe d'évolution de la règle $\overline{Y} \Rightarrow \overline{X}$ se situe au-dessus de celle de $X \Rightarrow Y$, si la règle $X \Rightarrow Y$ est jugée pertinente par l'utilisateur, c'est-à-dire jugée assez proche de l'implication logique, alors la règle négative $\overline{Y} \Rightarrow \overline{X}$ le sera également puisque sa distance à l'implication logique est supérieure. Nous pouvons en déduire la méta-règle suivante :

$\forall X \Rightarrow Y$ avec $sup(X) < sup(Y) < \frac{1}{2}$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) \geqslant min_{M_G}$.

☛ **Cas 2 :** $\frac{1}{2} < sup(X) < sup(Y)$

Dans ce cas, l'indépendance pour la règle $X \Rightarrow Y$ intervient après le point d'équilibre ($sup(Y) > \frac{1}{2}$) et l'indépendance pour la règle $\overline{Y} \Rightarrow \overline{X}$ intervient avant le point d'équilibre ($sup(\overline{X}) < \frac{1}{2}$). La figure 3.6 restitue les courbes d'évolution de la mesure M_G dans la zone attractive pour les règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$.

Sur cette courbe, le point d'équilibre pour la règle $\overline{Y} \Rightarrow \overline{X}$ intervient pour la valeur a qui est égale à $-\frac{sup(\overline{Y})}{2 \times sup(X)} + 1$.

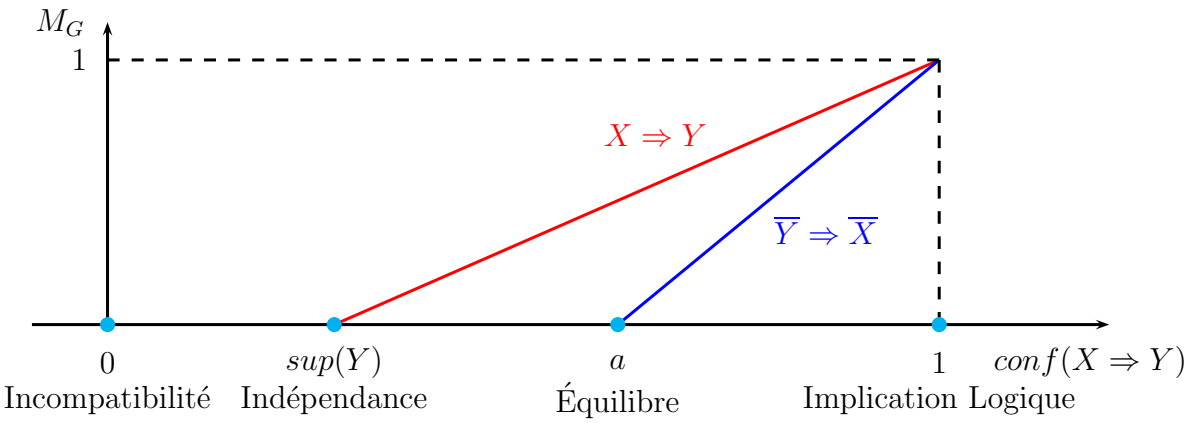

FIGURE 3.6 – Courbe de M_G pour $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$ dans le cas 2

Preuve :

$$
\begin{aligned}
conf(\overline{Y} \Rightarrow \overline{X}) = \tfrac{1}{2} \quad &\Leftrightarrow \quad \tfrac{sup(\overline{X}\,\overline{Y})}{sup(\overline{Y})} = \tfrac{1}{2} \\
&\Leftrightarrow \quad \tfrac{1-sup(X)-sup(Y)+sup(XY)}{sup(\overline{Y})} = \tfrac{1}{2} \\
&\Leftrightarrow \quad 1 - sup(X) - sup(Y) + sup(XY) = \tfrac{1}{2} \times sup(\overline{Y}) \\
&\Leftrightarrow \quad sup(XY) = \tfrac{1}{2} \times sup(\overline{Y}) - 1 + sup(X) + sup(Y) \\
&\Leftrightarrow \quad sup(XY) = \tfrac{1}{2} \times sup(\overline{Y}) - 1 + sup(X) + (1 - sup(\overline{Y})) \\
&\Leftrightarrow \quad sup(XY) = -\tfrac{1}{2} \times sup(\overline{Y}) + sup(X) \\
&\Leftrightarrow \quad \tfrac{sup(XY)}{sup(X)} = \tfrac{-\frac{1}{2} \times sup(\overline{Y}) + sup(X)}{sup(X)} \\
&\Leftrightarrow \quad conf(X \Rightarrow Y) = -\tfrac{sup(\overline{Y})}{2 \times sup(X)} + 1 = a
\end{aligned}
$$

Comme la courbe d'évolution de la règle $X \Rightarrow Y$ se situe au-dessus de celle de la règle $\overline{Y} \Rightarrow \overline{X}$, si la règle $X \Rightarrow Y$ est jugée non pertinente par l'utilisateur, c'est-à-dire jugée pas assez proche de l'implication logique, alors la règle négative $\overline{Y} \Rightarrow \overline{X}$ sera également non pertinente puisque sa distance à l'implication logique est inférieure. Nous pouvons en déduire la méta-règle suivante :

> $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$
> si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) < min_{M_G}$.

☛ **Cas 3 :** $sup(X) < \frac{1}{2} < sup(Y)$

Dans ce cas, les deux règles $X \Rightarrow Y$ et $\overline{Y} \Rightarrow \overline{X}$ ont leur point d'indépendance après le point d'équilibre ($sup(Y) > \frac{1}{2}$ *et* $sup(\overline{X}) > \frac{1}{2}$). Comme le point d'indépendance est le même pour les deux types de règles, les courbes d'évolution de la mesure M_G sont confondues dans la zone attractive. Nous pouvons donc étendre les méta-règles précédentes pour prendre en compte ce troisième cas. Nous nommons ces méta-règles respectivement MR_2 et MR_3 :

MR₂ : $\forall X \Rightarrow Y$ avec $sup(X) < sup(Y) < \frac{1}{2}$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) \geqslant min_{M_G}$.

MR₃ : $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) < min_{M_G}$.

Après avoir dégagé ces deux méta-règles afin d'inférer l'intérêt des règles $\overline{Y} \Rightarrow \overline{X}$ à partir de l'intérêt des règles $X \Rightarrow Y$, nous allons maintenant étudier comment induire les règles $\overline{X} \Rightarrow \overline{Y}$.

➤ Méta-règles pour en déduire l'intérêt des règles $\overline{X} \Rightarrow \overline{Y}$

Comme précédemment, nous considérons trois cas possibles pour la recherche de ces méta-règles.

☛ Cas 1 : $sup(X) < sup(Y) < \frac{1}{2}$

Dans ce cas, l'indépendance pour la règle $X \Rightarrow Y$ intervient avant le point d'équilibre ($sup(Y) < \frac{1}{2}$), et l'indépendance pour la règle $\overline{X} \Rightarrow \overline{Y}$ intervient après le point d'équilibre ($sup(\overline{Y}) > \frac{1}{2}$). La figure 3.7 restitue les courbes d'évolution de la mesure M_G dans la zone attractive pour les règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$.

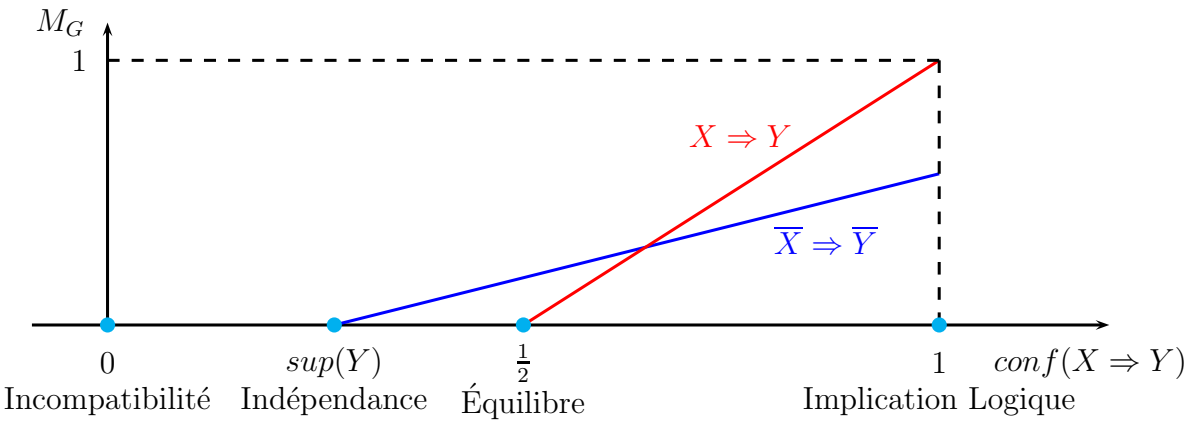

FIGURE 3.7 – Courbe de M_G pour $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ dans le cas 1

Nous remarquons que la courbe de la règle $\overline{X} \Rightarrow \overline{Y}$ n'atteint pas la valeur 1. En effet, l'implication logique ne peut pas être atteinte par la règle $\overline{X} \Rightarrow \overline{Y}$ puisque $sup(\overline{X}) > sup(\overline{Y})$. Les deux courbes se croisent et nous ne pouvons donc faire aucune déduction sur l'intérêt de la règle $\overline{X} \Rightarrow \overline{Y}$ dans le cas où $sup(X) < sup(Y) < \frac{1}{2}$.

☛ **Cas 2 :** $\frac{1}{2} < sup(X) < sup(Y)$

Dans ce cas, nous pouvons réutiliser les méta-règles précédemment extraites. En effet, la méta-règle MR_3 peut s'appliquer puisque $\frac{1}{2} < sup(X) < sup(Y)$:

MR_3 : si $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$ et $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) < min_{M_G}$.

On peut ensuite appliquer la méta-règle MR_1 à la règle $\overline{Y} \Rightarrow \overline{X}$ car le support de la prémisse est bien inférieur au support de la conclusion puisque $sup(\overline{Y}) < sup(\overline{X})$:

MR_1 : si $sup(X) \leqslant sup(Y)$ et $M_G(X \Rightarrow Y) \leqslant min_{M_G}$ alors $M_G(Y \Rightarrow X) \leqslant min_{M_G}$.

Nous avons d'après MR_3, $M_G(\overline{Y} \Rightarrow \overline{X}) < min_{M_G}$, et comme $sup(\overline{Y}) < sup(\overline{X})$, nous pouvons en déduire d'après MR_1 que $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$. Nous pouvons donc déduire la méta-règle suivante :

$\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$.

☛ **Cas 3 :** $sup(X) < \frac{1}{2} < sup(Y)$

Dans ce cas, l'indépendance pour la règle $X \Rightarrow Y$ intervient après le point d'équilibre ($sup(Y) > \frac{1}{2}$), et l'indépendance pour la règle $\overline{X} \Rightarrow \overline{Y}$ intervient avant le point d'équilibre ($sup(\overline{Y}) < \frac{1}{2}$). La figure 3.8 restitue les courbes d'évolution de la mesure M_G dans la zone attractive pour les règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$.

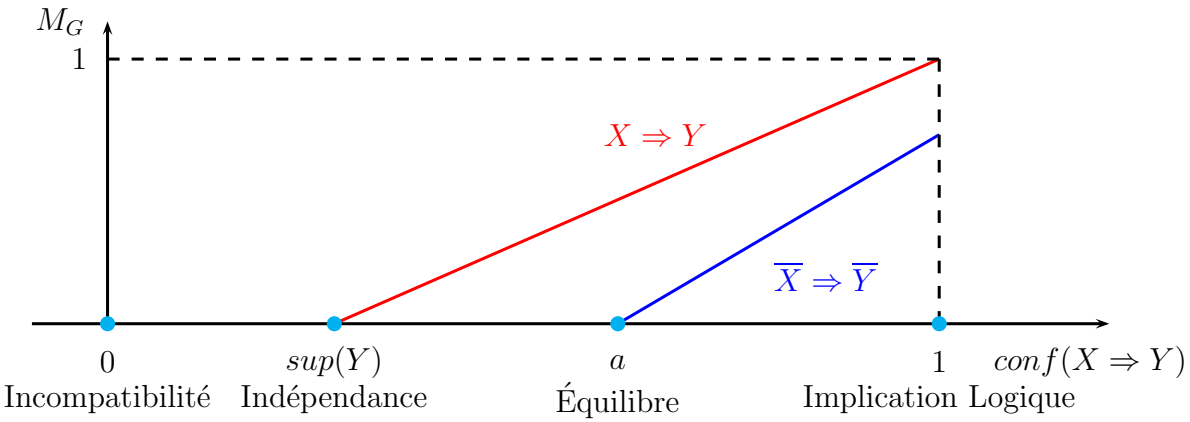

FIGURE 3.8 – Courbe de M_G pour $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ dans le cas 3

Le point d'équilibre pour la règle $\overline{X} \Rightarrow \overline{Y}$ intervient pour la valeur a qui est égale à $-\frac{sup(\overline{X})}{2 \times sup(X)} + \frac{sup(Y)}{sup(X)}$.

Preuve :

$$
\begin{array}{lll}
conf(\overline{X} \Rightarrow \overline{Y}) = \frac{1}{2} & \Leftrightarrow & \frac{sup(\overline{X}\,\overline{Y})}{sup(\overline{X})} = \frac{1}{2} \\
& \Leftrightarrow & \frac{1-sup(X)-sup(Y)+sup(XY)}{sup(\overline{X})} = \frac{1}{2} \\
& \Leftrightarrow & 1 - sup(X) - sup(Y) + sup(XY) = \frac{1}{2} \times sup(\overline{X}) \\
& \Leftrightarrow & sup(XY) = \frac{1}{2} \times sup(\overline{X}) - 1 + sup(X) + sup(Y) \\
& \Leftrightarrow & sup(XY) = \frac{1}{2} \times sup(\overline{X}) - 1 + (1 - sup(\overline{X})) + sup(Y) \\
& \Leftrightarrow & sup(XY) = -\frac{1}{2} \times sup(\overline{X}) + sup(Y) \\
& \Leftrightarrow & \frac{sup(XY)}{sup(X)} = \frac{-\frac{1}{2} \times sup(\overline{X}) + sup(Y)}{sup(X)} \\
& \Leftrightarrow & conf(X \Rightarrow Y) = -\frac{sup(\overline{X})}{2 \times sup(X)} + \frac{sup(Y)}{sup(X)} = a
\end{array}
$$

Comme la courbe d'évolution de la règle $X \Rightarrow Y$ se situe au-dessus de celle de la règle $\overline{X} \Rightarrow \overline{Y}$, si la règle $X \Rightarrow Y$ est jugée non pertinente par l'utilisateur, c'est-à-dire jugée pas assez proche de l'implication logique alors la règle négative $\overline{X} \Rightarrow \overline{Y}$ sera également non pertinente puisque sa distance à l'implication logique est inférieure. Nous pouvons en déduire la méta-règle suivante :

$\forall X \Rightarrow Y$ avec $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$.

La contraposée de la méta-règle précédente conduit à :

$\forall X \Rightarrow Y$ avec $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{M_G}$ alors $M_G(X \Rightarrow Y) \geqslant min_{M_G}$.

En modifiant cette contraposée pour déduire l'intérêt de $\overline{X} \Rightarrow \overline{Y}$ à partir de l'intérêt de $X \Rightarrow Y$, on obtient :

$\forall X \Rightarrow Y$ avec $sup(Y) < \frac{1}{2} < sup(X)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{M_G}$.

En conclusion, si on combine les conditions des méta-règles trouvées dans le cas où $\frac{1}{2} < sup(X) < sup(Y)$ (*cas 2*) et dans le cas où $sup(X) < \frac{1}{2} < sup(Y)$ (*cas 3*), nous pouvons déduire la méta-règle MR_4 ainsi que sa contraposée :

MR$_4$: $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$.

MRC$_4$: $\forall X \Rightarrow Y$ avec $sup(Y) < sup(X) < \frac{1}{2}$ ou $sup(Y) < \frac{1}{2} < sup(X)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{M_G}$.

La dernière étude concerne les règles antinomiques $X \Rightarrow \overline{Y}$.

➤ Méta-règles pour en déduire l'intérêt des règles antinomiques $X \Rightarrow \overline{Y}$

Comme nous l'avons dit à la section 3.2.2, il existe une relation entre les règles antinomiques. Nous rappelons cette propriété ci-dessous :

- $M_{G_a}(X \Rightarrow \overline{Y}) = -M_{G_r}(X \Rightarrow Y)$,
- $M_{G_r}(X \Rightarrow \overline{Y}) = -M_{G_a}(X \Rightarrow Y)$,
- $M_{G_i}(X \Rightarrow \overline{Y}) = -M_{G_i}(X \Rightarrow Y)$.

Cette propriété nous permet de déduire ces quatre méta-règles :

MR$_5$: $\forall X \Rightarrow Y$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$.

MR$_6$: $\forall X \Rightarrow \overline{Y}$
si $M_G(X \Rightarrow \overline{Y}) \geqslant min_{M_G}$ alors $M_G(X \Rightarrow Y) < min_{M_G}$.

MR$_7$: $\forall X \Rightarrow Y$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$.

MR$_8$: $\forall X \Rightarrow \overline{Y}$
si $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$ alors $M_G(X \Rightarrow Y) < min_{M_G}$.

En conclusion, dans cette étude nous avons découvert huit méta-règles qui permettent de déduire sous certaines conditions l'intérêt de certaines règles à partir de l'intérêt de la règle positive $X \Rightarrow Y$ ou de la règle négative $X \Rightarrow \overline{Y}$. Les méta-règles découvertes dans cette partie sont synthétisées dans la figure 3.9.

Méta-règles

$\mathbf{MR_1}$: $\forall X \Rightarrow Y$ avec $sup(X) < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(Y \Rightarrow X) < min_{M_G}$.

$\mathbf{MRC_1}$: $\forall X \Rightarrow Y$ avec $sup(X) \geqslant sup(Y)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(Y \Rightarrow X) \geqslant min_{M_G}$.

$\mathbf{MR_2}$: $\forall X \Rightarrow Y$ avec $sup(X) < sup(Y) < \frac{1}{2}$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) \geqslant min_{M_G}$.

$\mathbf{MR_3}$: $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{Y} \Rightarrow \overline{X}) < min_{M_G}$.

$\mathbf{MR_4}$: $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$.

$\mathbf{MRC_4}$: $\forall X \Rightarrow Y$ avec $sup(Y) < sup(X) < \frac{1}{2}$ ou $sup(Y) < \frac{1}{2} < sup(X)$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) \geqslant min_{M_G}$.

$\mathbf{MR_5}$: $\forall X \Rightarrow Y$
si $M_G(X \Rightarrow Y) \geqslant min_{M_G}$ alors $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$.

$\mathbf{MR_6}$: $\forall X \Rightarrow \overline{Y}$
si $M_G(X \Rightarrow \overline{Y}) \geqslant min_{M_G}$ alors $M_G(X \Rightarrow Y) < min_{M_G}$.

$\mathbf{MR_7}$: $\forall X \Rightarrow Y$
si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$.

$\mathbf{MR_8}$: $\forall X \Rightarrow \overline{Y}$
si $M_G(X \Rightarrow \overline{Y}) < min_{M_G}$ alors $M_G(X \Rightarrow Y) < min_{M_G}$.

FIGURE 3.9 – Récapitulatif des 10 méta-règles extraites

Trois méta-règles permettent de déduire qu'une règle est intéressante si la valeur de M_G pour la règle positive $X \Rightarrow Y$ est valide : $\mathbf{MRC_1}$, $\mathbf{MR_2}$ et $\mathbf{MRC_4}$. Quatre méta-règles permettent d'éviter l'étude d'une règle si la valeur de M_G pour la règle positive $X \Rightarrow Y$ est non valide : $\mathbf{MR_1}$, $\mathbf{MR_3}$, $\mathbf{MR_4}$ et $\mathbf{MR_7}$. Une méta-règle permet d'éviter l'étude d'une règle si la valeur de M_G pour la règle positive $X \Rightarrow Y$ est non valide : $\mathbf{MR_5}$. Deux autres méta-règles permettent d'éviter l'étude d'une règle à partir de l'intérêt valide ou pas de la règle $X \Rightarrow \overline{Y}$: $\mathbf{MR_6}$ et $\mathbf{MR_8}$.

3.4 Conclusion

Dans ce chapitre, nous avons présenté les solutions que nous proposons pour résoudre les problèmes soulevés dans le chapitre précédent. Au lieu de baser notre extraction sur les motifs fréquents, nous proposons d'utiliser les motifs raisonnablement fréquents qui vont permettre d'éliminer les règles composées de motifs omniprésents qui, comme nous l'avons démontré, conduisent à des règles inintéressantes. Notre deuxième proposition consiste à utiliser la mesure M_G. Après avoir présenté la sémantique de cette mesure, nous avons présenté les règles inintéressantes qu'elle permet de supprimer.

Dans la seconde partie du chapitre nous avons optimisé le parcours de recherche des règles. Cette étude a commencé par distinguer les règles potentiellement intéressantes des règles inintéressantes et nous a permis de diviser le nombre de règles étudiées par deux. En conclusion, les règles à étudier dépendent de la confiance de la règle positive par rapport au support de la conclusion.

Nous avons ensuite poursuivi l'optimisation du parcours de recherche en restreignant l'espace de recherche par l'utilisation de méta-règles basées sur la mesure M_G. Plus précisément, nous avons voulu connaître sous quelles conditions nous pouvons exclure certaines règles de l'étude.

Et enfin, nous avons défini un nouveau type de règles négatives représentant des conjonctions de motifs négatifs $\overline{x}_1..\overline{x}_p \Rightarrow \overline{y}_1..\overline{y}_q$ que nous jugeons utile d'extraire. De plus, l'ajout de ce type de règle nous oblige à ajouter un nouveau seuil qui renforce un peu plus notre souhait d'extraire les règles les plus intéressantes possibles.

Dans le prochain chapitre, nous allons incorporer ces optimisations dans notre méthode d'extraction et proposer un algorithme pour extraire efficacement les règles positives et négatives.

Chapitre 4

Algorithme d'extraction

Sommaire

4.1 Introduction

L'objectif de ce chapitre est de proposer un nouvel algorithme d'extraction de règles d'association positives et négatives utilisant les optimisations que nous avons proposées dans le chapitre précédent. Ces travaux ont été publiés dans [Guillaume and Papon, 2013a] et [Guillaume and Papon, 2013b].

La première partie de ce chapitre présente les différentes contraintes que doivent respecter les règles pour être considérées comme valides et être extraites par notre algorithme. Nous présentons ensuite les différents algorithmes que nous utilisons au cours du processus d'extraction. Le processus d'extraction se divise en trois parties, à savoir la recherche des motifs raisonnablement fréquents, la recherche des motifs négatifs minimaux raisonnablement fréquents et enfin l'extraction des règles valides à partir des motifs raisonnablement fréquents précédemment trouvés.

Nous terminons ce chapitre en déroulant notre algorithme sur notre exemple fil-rouge.

4.2 Règles d'association positives et négatives valides

Chaque règle, quel que soit son type, est générée à partir des motifs raisonnablement fréquents XY. Il faut donc que le support du motif XY vérifie min_{sup}, max_{sup} et que le support du motif $\ddot{XY}$ vérifie $min_{s\ddot{u}p}$. La règle est ensuite générée si son support, sa confiance et sa valeur de M_G sont valides. Pour les règles négatives $X \Rightarrow \overline{Y}$, $\overline{X} \Rightarrow \overline{Y}$ et

$\overline{X} \Rightarrow Y$, il faut vérifier que les prémisses et/ou les conclusions négatives sont constituées de motifs négatifs minimaux raisonnablement fréquents dont nous donnons la définition :

Définition 16 - Motif négatif minimal raisonnablement fréquent :
Un motif $\overline{XY}$ est motif négatif minimal raisonnablement fréquent si $min_{sup} \leqslant sup(\overline{XY}) \leqslant max_{sup}$ *et* $sup(\overline{X}) < min_{sup}$ *et* $sup(\overline{Y}) < min_{sup}$.

Une dernière contrainte est appliquée au nouveau type de règle $\ddot{X} \Rightarrow \ddot{Y}$, où le motif $\ddot{X}\ddot{Y}$ doit avoir une taille strictement supérieure à 2. L'omission de cette contrainte amène à générer des règles en double. En effet, pour un *2*-motif $i_1 i_2$, les règles $\overline{i}_1 \Rightarrow \overline{i}_2$ et $\ddot{i}_1 \Rightarrow \ddot{i}_2$ sont identiques. Afin d'éviter cette redondance, nous pouvons appliquer cette dernière contrainte sur les règles $\ddot{X} \Rightarrow \ddot{Y}$ ou sur les règles $\overline{X} \Rightarrow \overline{Y}$. Notre approche étant la seule à notre connaissance à générer les règles composées de conjonctions de motifs négatifs, il est plus pertinent de garder les règles $\overline{X} \Rightarrow \overline{Y}$ afin de pouvoir les comparer avec celles extraites par les autres approches. C'est pourquoi, nous choisissons d'appliquer cette dernière contrainte sur les règles $\ddot{X} \Rightarrow \ddot{Y}$ (*i.e taille*$(\ddot{X}\ddot{Y}) > 2$).

Le tableau 4.1 récapitule les contraintes pour qu'une règle soit valide pour notre approche.

En conclusion, l'ensemble des règles $X \Rightarrow Y$, $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$, $\overline{X} \Rightarrow \overline{Y}$ et $\ddot{X} \Rightarrow \ddot{Y}$ doit respecter le support minimum, le support maximum, le support du motif négatif $\ddot{X}\ddot{Y}$, la mesure M_G et la confiance.

Dans les prochaines sections, nous présentons les différentes étapes de l'algorithme. L'algorithme va se dérouler en trois étapes :

- rechercher l'ensemble RF des motifs raisonnablement fréquents,
- rechercher l'ensemble $NMRF$ des motifs négatifs minimaux raisonnablement fréquents,
- générer les règles valides.

La prochaine section se focalise sur la première étape de l'algorithme qui correspond à la recherche des motifs raisonnablement fréquents.

$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$
$sup(XY) \geq min_{sup}$	$sup(XY) \geq min_{sup}$
$sup(\ddot{XY}) \geq min_{s\ddot{u}p}$	$sup(\ddot{XY}) \geq min_{s\ddot{u}p}$
$sup(XY) \leq max_{sup}$	$sup(XY) \leq max_{sup}$
	$sup(\overline{X}Y) \geq min_{sup}$
	$sup(\overline{X}Y) \leq max_{sup}$
$M_G(X \Rightarrow Y) \geq min_{M_G}$	$M_G(\overline{X} \Rightarrow Y) \geq min_{M_G}$
$conf(X \Rightarrow Y) \geq min_{conf}$	$conf(\overline{X} \Rightarrow Y) \geq min_{conf}$
	$\overline{X}$ *minimal*
$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$
$sup(XY) \geq min_{sup}$	$sup(XY) \geq min_{sup}$
$sup(\ddot{XY}) \geq min_{s\ddot{u}p}$	$sup(\ddot{XY}) \geq min_{s\ddot{u}p}$
$sup(XY) \leq max_{sup}$	$sup(XY) \leq max_{sup}$
$sup(X\overline{Y}) \geq min_{sup}$	$sup(\overline{X}\,\overline{Y}) \geq min_{sup}$
$sup(X\overline{Y}) \leq max_{sup}$	$sup(\overline{X}\,\overline{Y}) \leq max_{sup}$
$M_G(X \Rightarrow \overline{Y}) \geq min_{M_G}$	$M_G(\overline{X} \Rightarrow \overline{Y}) \geq min_{M_G}$
$conf(X \Rightarrow \overline{Y}) \geq min_{conf}$	$conf(\overline{X} \Rightarrow \overline{Y}) \geq min_{conf}$
$\overline{Y}$ *minimal*	$\overline{X}$ et $\overline{Y}$ *minimaux*
$\ddot{X} \Rightarrow \ddot{Y}$	
$sup(XY) \geq min_{sup}$	$M_G(\ddot{X} \Rightarrow \ddot{Y}) \geq min_{M_G}$
$sup(\ddot{XY}) \geq min_{s\ddot{u}p}$	$conf(\ddot{X} \Rightarrow \ddot{Y}) \geq min_{conf}$
$sup(XY) \leq max_{sup}$	taille($\ddot{XY}$)> 2

TABLEAU 4.1 – Règles valides

4.3 Génération des motifs raisonnablement fréquents

La recherche des motifs raisonnablement fréquents est effectuée par la fonction *MRF* (*cf. algorithme* 12).

Algorithme 12 : *MRF* - Recherche des Motifs Raisonnablement Fréquents

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, support maximum max_{sup}, support minimum des conjonctions négatives $min_{s\ddot{u}p}$
Sortie : ensemble des motifs Raisonnablement Fréquents RF

```
1  si max_sup est non défini alors
2      max_sup = 1 − min_sup
3  si min_süp est non défini alors
4      min_süp = min_sup
5  RF = ∅
6  C_1 = {i ∈ I tel que sup(i) ⩽ max_sup et sup(ī) ⩽ max_sup}
7  pour (k = 1; C_k ≠ ∅; k++) faire
8      RF_k = ∅
9      pour tout motif X ∈ C_k faire
10         s = support(D, X)
11         s̈ = calculSupportConjonction(X, s, F)
12         si s ⩾ min_sup et s̈ ⩾ min_süp alors
13             RF_k = RF_k ∪ {X}
14     C_{k+1} = candidats(RF_k)
15     RF = RF ∪ RF_k
16 retourner RF
```

Cette étape est similaire à celle proposée dans l'algorithme `Apriori` pour générer les motifs fréquents comme nous l'avons dit précédemment. Nous ajoutons deux contraintes supplémentaires : un seuil maximum max_{sup} pour le support du motif X et un seuil minimum $min_{s\ddot{u}p}$ pour le support du motif $\ddot{X}$. Si l'utilisateur ne fournit pas de valeurs pour ces deux seuils, l'algorithme va utiliser les valeurs par défaut (*lignes 1 à 4*). Ensuite (*lignes 5 et 6*), nous initialisons l'ensemble RF des motifs raisonnablement fréquents à l'ensemble vide et l'ensemble C_1 des *1*-motifs candidats à l'ensemble de tous les items i de la base de données $\mathcal{D}$ passée en paramètre tel que $sup(i) \leqslant max_{sup}$ et $sup(\bar{i}) \leqslant max_{sup}$. Le support maximum étant une mesure monotone (*cf. définition* 10), en vérifiant cette contrainte dès la recherche des items, nous nous assurons que tous les motifs respecteront ces contraintes. Le processus suivant (*lignes 7 à 15*) va être réitéré jusqu'à ce que l'on n'obtienne plus de candidat ($C_k \neq \varnothing$). Les candidats sont générés à partir de l'ensemble RF_k des motifs fréquents (*ligne 15*) afin d'éviter de générer des règles redondantes (*cf. preuve 3 section* 3.2.1). La construction des candidats à partir de l'ensemble RF_k des motifs fréquents va se dérouler de la même façon que la recherche des candidats dans `Apriori`. Par conséquent, nous utilisons la même fonction *candidats* (*cf. algorithme* 2) pour accomplir cette tâche. Pour un niveau k donné, nous commençons par initialiser l'ensemble RF_k des motifs fréquents à l'ensemble vide (*ligne 8*). Puis, pour chaque candidat $X \in C_k$ (*ligne 9*), nous calculons tout d'abord le support de X (*ligne 10*) puis le support de $\ddot{X}$ (*ligne 11*). Le support de $\ddot{X}$ est déduit du support des motifs positifs avec la

fonction *calculSupportConjonction* (*ligne 11*) grâce à la formule du crible de Poincaré (*cf. section* 3.2.3). Si les motifs X et $\ddot{X}$ sont fréquents (*ligne 12*) alors le motif X est ajouté à l'ensemble RF_k des motifs fréquents (*ligne 13*). Lorsque tous les candidats $X \in C_k$ sont parcourus, nous générons l'ensemble C_{k+1} des candidats de taille supérieure (*ligne 14*). La dernière étape consiste à ajouter l'ensemble RF_k des k-motifs raisonnablement fréquents à l'ensemble RF des motifs raisonnablement fréquents (*ligne 15*). Lorsqu'il n'existe plus de candidats à parcourir, on retourne l'ensemble RF des motifs raisonnablement fréquents (*ligne 16*).

Le second algorithme utilisé dans notre méthode d'extraction des règles positives et négatives consiste à rechercher les motifs négatifs minimaux raisonnablement fréquents.

4.4 Génération des motifs négatifs minimaux raisonnablement fréquents

La fonction *MNMRF* (*cf. algorithme* 13) procède à l'extraction des motifs négatifs minimaux raisonnablement fréquents.

Algorithme 13 : *MNMRF* - Recherche des Motifs Négatifs Minimaux Raisonnablement Fréquents

Entrées : base de données $\mathcal{D}$, support minimum min_{sup}, support maximum max_{sup}, ensemble des motifs Raisonnablement Fréquents RF
Sortie : ensemble des motifs Négatifs Minimaux Raisonnablement Fréquents $NMRF$

```
1  NMRF = ∅
2  NMRF_1 = {ī tel que i ∈ RF_1 et min_sup ⩽ sup(ī) ⩽ max_sup} ;
3  NMRF = NMRF ∪ NMRF_1
4  CP_1 = {ī tel que i ∈ I} − NMRF_1 − {ī tel que i ∈ RF_1 et sup(ī) ⩾ max_sup} ;
5  C_2 = candidats(CP_1)
6  pour (k = 2; C_k ≠ ∅; k++) faire
7      CP_k = ∅
8      pour tout motif X̄ ∈ C_k faire
9          si min_sup ⩽ sup(X̄) ⩽ max_sup alors
10             NMRF_k = NMRF_k ∪ {X̄}
11         sinon si sup(X̄) < min_sup alors
12             CP_k = CP_k ∪ {X̄}
13     C_{k+1} = candidats(CP_k)
14     NMRF = NMRF ∪ NMRF_k
15 retourner NMRF
```

Cette fonction est très utile pour dégager uniquement les règles négatives pertinentes. En effet si le support d'un motif $\overline{X}$ est supérieur au support minimum, alors le support du motif $\overline{XY}$ le sera également et ceci quel que soit Y. Comme $sup(\overline{X}) = 1 - sup(X)$, $sup(\overline{XY}) = 1 - sup(XY)$ et que la propriété anti-monotone du support (*cf. définition* 9) garantit $sup(X) \geqslant sup(XY)$, la relation $sup(\overline{XY}) \geqslant sup(\overline{X})$ est vérifiée. Cette propriété va donc engendrer une quantité importante de règles non pertinentes. Supposons

par exemple que la règle $\overline{café} \Rightarrow \overline{thé}$ soit valide pour le support et la confiance. Les règles $\overline{café} \Rightarrow \overline{thé,\ sucre}$, $\overline{café} \Rightarrow \overline{thé,\ beurre}$ ainsi que la règle $\overline{café} \Rightarrow \overline{thé,\ sucre,\ beurre}$ seront également valides pour le support et la confiance puisque $sup(\overline{café,\ thé}) \leqslant sup(\overline{café,\ thé,\ X})$ et ceci quel que soit X. Pour élaguer ces règles non pertinentes, nous allons devoir rechercher les motifs négatifs minimaux raisonnablement fréquent, c'est-à-dire les motifs négatifs tel qu'il n'existe pas de sous-motifs négatifs également raisonnablement fréquents.

Cet algorithme commence par initialiser l'ensemble $NMRF$ à l'ensemble vide (*ligne 1*), puis $NMRF_1$ des *1*-motifs négatifs minimaux raisonnablement fréquents à l'ensemble des négations d'items de taille 1 raisonnablement fréquentes (*ligne 2*). Cette initialisation est possible grâce à la formule suivante : $sup(\overline{i}) = 1 - sup(i)$ et à la connaissance des supports des items i calculés lors de la recherche des motifs raisonnablement fréquents (*cf. algorithme* 12). Nous ajoutons ensuite cet ensemble $NMRF_1$ des *1*-motifs négatifs minimaux raisonnablement fréquents à l'ensemble $NMRF$ des motifs négatifs minimaux raisonnablement fréquents (*ligne 3*). Puis, nous stockons dans l'ensemble CP_1 les *1*-motifs négatifs candidats. Ces candidats permettront par la suite de générer les motifs négatifs minimaux raisonnablement fréquents de taille supérieure. Pour se faire, nous ôtons à l'ensemble des négations d'items $\overline{i}$ tel que $i \in \mathcal{I}$ l'ensemble des *1*-motifs négatifs minimaux raisonnablement fréquents mais également toutes les négations d'items $\overline{i}$ dont le support est supérieur au seuil maximum (*ligne 4*). En d'autres mots, l'ensemble CP_1 contient uniquement les items $\overline{i}$ qui possèdent un support plus faible que min_{sup}. En effet, si le support de $\overline{i}$ est supérieur à max_{sup} alors tous les sur-ensembles $\overline{iX}$ auront un support supérieur ou égal à $\overline{i}$ et ne pourront donc pas devenir des motifs négatifs minimaux raisonnablement fréquents à la prochaine itération. L'étape suivante génère l'ensemble C_2 des *2*-motifs candidats à partir de l'ensemble CP_1 (*ligne 5*). Cette fois encore, nous utilisons la fonction *candidats* d'`Apriori` (*cf. algorithme* 2) pour générer nos candidats. Le processus suivant (*lignes 6 à 14*) va être réitéré jusqu'à ce que l'on n'obtienne plus de candidat ($C_k \neq \varnothing$). Nous commençons par initialiser l'ensemble CP_k des k-motifs candidats potentiels à l'ensemble vide (*ligne 7*). Ensuite pour chaque candidat $\overline{X} \in C_k$ (*ligne 8*) nous vérifions la contrainte du support (*ligne 9*). Si $\overline{X}$ est raisonnablement fréquent alors nous l'ajoutons à l'ensemble $NMRF_k$ des k-motifs négatifs minimaux raisonnablement fréquents (*ligne 10*). Sinon, si son support est inférieur au seuil du support minimum (*ligne 11*), alors nous ajoutons le motif $\overline{X}$ à l'ensemble des k-motifs candidats potentiels (*ligne 12*) qui sera utilisé par la suite pour générer le prochain niveau de candidats (*ligne 13*). Lorsque tous les candidats $\overline{X} \in C_k$ sont parcourus, nous générons l'ensemble C_{k+1} des candidats de taille supérieure (*ligne 13*) à partir de l'ensemble CP_k des k-motifs candidats potentiels en utilisant, encore une fois, la fonction *candidats*. La dernière étape ajoute l'ensemble $NMRF_k$ des k-motifs négatifs minimaux raisonnablement fréquents à l'ensemble $NMRF$ des motifs négatifs minimaux raisonnablement fréquents (*ligne 14*). Lorsqu'il n'existe plus de candidats à parcourir, on retourne l'ensemble $NMRF$ (*ligne 15*).

La prochaine section présente l'algorithme principal que nous utilisons pour extraire l'ensemble des règles valides.

4.5 Génération des règles

Avant de présenter l'algorithme d'extraction des règles d'association positives et négatives, nous allons modéliser la propriété de la confiance présentée dans la propriété 1, sous forme de méta-règle :

$MR_9 : \forall(X, Y, Z)$ tel que $Z \subsetneq Y \subsetneq X$,
si $conf(Z \Rightarrow X \backslash Z) < min_{conf}$ alors $conf(Y \Rightarrow X \backslash Y) < min_{conf}$.

Cette méta-règle sera utilisée pour l'extraction des règles du nouveau type et permettra d'éviter l'étude de règles dont nous savons par avance que la valeur de la confiance est insuffisante. Nous aurions également pu utiliser cette méta-règle pour les règles positives, seulement nous devons calculer la confiance de la règle positive pour déterminer la zone d'appartenance de la règle. Par conséquent, nous sommes obligés de calculer la confiance de la règle positive et la méta-règle n'est donc plus utile pour les règles positives.

L'algorithme utilise également une méta-règle que nous avons présentée au chapitre précédent (*cf. section* 3.3.2). Nous la rappelons ci-dessous. La méta-règle dont nous allons nous servir est MR_4.

$$MR_4 : \forall X \Rightarrow Y \text{ avec } \left(\tfrac{1}{2} < sup(X) < sup(Y)\right) \text{ ou } \left(sup(X) < \tfrac{1}{2} < sup(Y)\right)$$
$$\text{si } M_G(X \Rightarrow Y) < min_{M_G} \text{ alors } M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}.$$

Cette méta-règle MR_4 nous révèle, que si la règle $X \Rightarrow Y$ est invalide pour la mesure M_G dans le cas où $\left(\frac{1}{2} < sup(X) < sup(Y)\right)$ ou $\left(sup(X) < \frac{1}{2} < sup(Y)\right)$, alors la règle $\overline{X} \Rightarrow \overline{Y}$ sera également invalide. Cette méta-règle sera utilisée dans la zone attractive mais également dans la zone répulsive (*cf. section* 3.2.2). Dans la zone répulsive l'invalidité de la règle $\overline{X} \Rightarrow Y$ sera déduite à partir de l'intérêt de la règle $X \Rightarrow \overline{Y}$.

La génération des règles d'association positives et négatives, à partir des motifs raisonnablement fréquents préalablement trouvés, est effectuée par la fonction *RAPN* (*cf. algorithme* 14). Cet algorithme étant assez long, nous l'avons divisé en plusieurs fonctions. Chaque type de règles va être étudié par une fonction différente : l'*algorithme* 15 pour les règles du type $X \Rightarrow Y$, l'*algorithme* 16 pour les règles du type $\overline{X} \Rightarrow \overline{Y}$, l'*algorithme* 17 pour les règles du type $X \Rightarrow \overline{Y}$, l'*algorithme* 18 pour les règles du type $\overline{X} \Rightarrow Y$ et l'*algorithme* 19 pour les règles du type $\ddot{X} \Rightarrow \ddot{Y}$.

Après avoir initialisé l'ensemble R des règles valides à l'ensemble nul (*ligne 1*), le processus va extraire l'ensemble des règles valides (*lignes 2 à 14*). Ainsi, pour chaque motif raisonnablement fréquent $X \in RF$ avec une taille strictement supérieure à 1 (*ligne 2*) (*puisqu'on ne peut pas générer une règle comportant qu'un seul item*) et pour chaque motif conclusion $Y \subsetneq X$ (*ligne 3*) ordonné par taille croissante (*comme pour* `Apriori`), nous commençons par déterminer la zone d'appartenance de la règle $X \backslash Y \Rightarrow Y$ en comparant la confiance de cette règle au support de la conclusion Y. Pour se faire, nous devons d'abord calculer la confiance de la règle $X \backslash Y \Rightarrow Y$ (*ligne 4*).

Si la règle est dans la zone attractive (*i.e.* $conf(X \backslash Y \Rightarrow Y) > max(\frac{1}{2}, sup(Y))$) (*ligne 5*), alors nous étudions uniquement deux règles : les règles $X \backslash Y \Rightarrow Y$ et $\overline{X \backslash Y} \Rightarrow \overline{Y}$. En effet, comme nous pouvons le voir dans la figure renseignant sur les règles potentiellement intéressantes et les règles inintéressantes en fonction de l'intérêt de la règle positive (*cf. figure* 3.4), nous pouvons voir que lorsque la confiance de la règle est supérieure au support de la conclusion, la règle se situe dans la zone potentiellement intéressante, et seules les

Algorithme 14 : *RAPN* - Extraction des Règles d'Association Positives et Négatives

Entrées : base de données $\mathcal{D}$, ensemble des motifs Raisonnablement Fréquents RF, ensemble des motifs Négatifs Minimaux Raisonnablement Fréquents $NMRF$, support minimum min_{sup}, support maximum max_{sup}, confiance minimum min_{conf}, M_G minimum min_{M_G}

Sortie : ensemble des règles valides R

1 $R = \varnothing$
2 **pour tout** *k-motif* $X \in RF$ *tel que* $k > 1$ **faire**
3 **pour toute** *conclusion* $Y \subsetneq X$ *telle que taille*$(Y)\uparrow$ **faire**
4 $c = conf(X \backslash Y \Rightarrow Y)$
5 **si** $c > max(\frac{1}{2}, sup(Y))$ **alors**
6 Étude de $X \backslash Y \Rightarrow Y$
7 **si** $\overline{X \backslash Y} \in NMRF$ ***et*** $\overline{Y} \in NMRF$ **alors**
8 $[\neg MR_4]$ Étude de $\overline{X \backslash Y} \Rightarrow \overline{Y}$
9 **sinon si** $c < min(\frac{1}{2}, sup(Y))$ **alors**
10 **si** $\overline{Y} \in NMRF$ **alors**
11 Étude de $X \backslash Y \Rightarrow \overline{Y}$
12 **si** $\overline{X \backslash Y} \in NMRF$ **alors**
13 $[\neg MR_4]$ Étude de $\overline{X \backslash Y} \Rightarrow Y$
14 $[\neg MR_9]$ Étude de $X \ddot{\backslash} Y \Rightarrow \ddot{Y}$
15 **retourner** R

règles du type $X \backslash Y \Rightarrow Y$, $Y \Rightarrow X \backslash Y$, $\overline{X \backslash Y} \Rightarrow \overline{Y}$ et $\overline{Y} \Rightarrow \overline{X \backslash Y}$ peuvent être intéressantes. La zone attractive de la mesure M_G correspond à la zone potentiellement intéressante à laquelle nous ajoutons la prise en compte de l'équilibre, c'est-à-dire qu'il faut également vérifier que la confiance de la règle est supérieure à $\frac{1}{2}$. Dans la zone attractive, seules ces quatre règles peuvent être intéressantes à condition qu'elles s'éloignent suffisamment du point d'équilibre. Par ailleurs, vu que notre algorithme parcourt l'ensemble des conclusions Y possibles, nous allons uniquement étudier les règles $X \backslash Y \Rightarrow Y$ et $\overline{X \backslash Y} \Rightarrow \overline{Y}$ dans la zone attractive. Réciproquement, nous étudierons uniquement les règles $X \backslash Y \Rightarrow \overline{Y}$ et $\overline{X \backslash Y} \Rightarrow Y$ dans la zone répulsive.

Nous débutons donc par l'étude de $X \backslash Y \Rightarrow Y$ (*ligne 6*) avec l'*algorithme 15*.

Algorithme 15 : Étude des règles du type $X \backslash Y \Rightarrow Y$

1 **si** $c \geqslant min_{conf}$ **alors**
2 **si** $M_G(X \backslash Y \Rightarrow Y) \geqslant min_{M_G}$ **alors**
3 $R = R \cup \{X \backslash Y \Rightarrow Y\}$

Dans l'*algorithme 15*, nous vérifions que la valeur de la confiance (*déjà calculée dans l'algorithme principal (algorithme 14 ligne 4)* est supérieure au seuil (*ligne 1 algorithme 15*). Puis, si la confiance de la règle est valide alors on vérifie la validité de la règle pour la mesure M_G (*ligne 2 algorithme 15*). Si M_G vérifie le seuil alors la règle $X \backslash Y \Rightarrow Y$ est

ajoutée à l'ensemble R des règles valides (*ligne 3 algorithme* 15).

Nous retournons ensuite à l'algorithme principal (*algorithme* 14) pour étudier la règle négative $\overline{X \backslash Y} \Rightarrow \overline{Y}$. Si $\overline{X \backslash Y}$ et $\overline{Y}$ sont des motifs négatifs minimaux raisonnablement fréquents, c'est-à-dire si $\overline{X \backslash Y}$ et $\overline{Y}$ sont contenus dans $NMRF$ (*ligne 7*), alors nous étudions la règle $\overline{X \backslash Y} \Rightarrow \overline{Y}$ (*ligne 8*) si la méta-règle MR_4 ne peut être appliquée. L'étude de cette règle est réalisée avec l'*algorithme* 16.

Algorithme 16 : Étude des règles du type $\overline{X \backslash Y} \Rightarrow \overline{Y}$

```
1 si min_sup ⩽ sup(X̄ Ȳ) ⩽ max_sup alors
2     si conf(X̄\Y ⇒ Ȳ) ⩾ min_conf alors
3         si M_G(X̄\Y ⇒ Ȳ) ⩾ min_MG alors
4             R = R ∪ {X̄\Y ⇒ Ȳ}
```

Cet *algorithme* 16 commence par vérifier si le support de la règle $\overline{X \backslash Y} \Rightarrow \overline{Y}$ est valide (*ligne 1 algorithme* 16). Pour vérifier le support, nous utilisons la formule suivante : $sup(\overline{X}\,\overline{Y}) = 1 - sup(X) - sup(Y) + sup(XY)$. Cette formule nous évite d'interroger la base de données $\mathcal{D}$. Si le support est vérifié, nous étudions ensuite la confiance (*ligne 2 algorithme* 16). Si la confiance est valide nous calculons la valeur de M_G (*ligne 3 algorithme* 16). Si la règle est valide pour la mesure M_G, la règle $\overline{X \backslash Y} \Rightarrow \overline{Y}$ est ajoutée à l'ensemble des règles valides R puis nous retournons à l'algorithme principal (*algorithme* 14).

Si la règle $X \backslash Y \Rightarrow Y$ est dans la zone répulsive (*i.e.* $conf(X \backslash Y \Rightarrow Y) < min(\frac{1}{2}, sup(Y))$) (*ligne 9*), alors nous étudions les règles : $X \backslash Y \Rightarrow \overline{Y}$ et $\overline{X \backslash Y} \Rightarrow Y$. Si $\overline{Y}$ est minimal (*ligne 10*), nous étudions la règle $X \backslash Y \Rightarrow \overline{Y}$ (*ligne 11*). Si $\overline{X \backslash Y}$ est un motif minimal (*ligne 12*) et que MR_4 n'est pas vérifiée alors la règle $X \backslash Y \Rightarrow \overline{Y}$ est ensuite étudiée (*ligne 13*). La vérification des contraintes (*support, confiance et* M_G) se fait avec l'*algorithme* 17 pour les règles du type $X \Rightarrow \overline{Y}$ et l'*algorithme* 18 pour les règles du type $\overline{X} \Rightarrow Y$. Expliquons maintenant ces deux algorithmes très semblables aux *algorithmes* 15 et 16.

Algorithme 17 : Étude des règles du type $X \backslash Y \Rightarrow \overline{Y}$

```
1 si min_sup ⩽ sup(X Ȳ) ⩽ max_sup alors
2     si conf(X\Y ⇒ Ȳ) ⩾ min_conf alors
3         si M_G(X\Y ⇒ Ȳ) ⩾ min_MG alors
4             R = R ∪ {X\Y ⇒ Ȳ}
```

L'*algorithme* 17 commence par vérifier si le support de la règle $X \backslash Y \Rightarrow \overline{Y}$ est valide (*ligne 1 algorithme* 17). Le support peut être calculé avec la formule $sup(X\overline{Y}) = sup(X) - sup(XY)$ et évite d'interroger la base de données $\mathcal{D}$. Nous vérifions ensuite, si le support est valide, que la contrainte de la confiance (*ligne 2 algorithme* 17) est respectée. Si c'est

le cas, nous vérifions enfin la valeur de M_G (*ligne 3 algorithme* 17). Si la valeur de M_G est suffisante, la règle $X \Rightarrow \overline{Y}$ est ajoutée à l'ensemble des règles valides R (*ligne 4 algorithme* 17) et nous retournons à l'algorithme principal (*algorithme* 14). L'*algorithme* 18 effectue les mêmes opérations mais utilise une formule différente pour calculer le support.

Algorithme 18 : Étude des règles du type $\overline{X \backslash Y} \Rightarrow Y$

```
1 si min_sup ⩽ sup(X̄Y) ⩽ max_sup alors
2     si conf(X̄\Y ⇒ Y) ⩾ min_conf alors
3         si M_G(X̄\Y ⇒ Y) ⩾ min_MG alors
4             R = R ∪ {X̄\Y ⇒ Y}
```

En effet, dans l'*algorithme* 18 la formule que nous allons utiliser est la suivante : $sup(\overline{X}Y) = sup(Y) - sup(XY)$. Les règles du type $\overline{X} \Rightarrow Y$ sont ajoutées à l'ensemble des règles valides R si les trois contraintes successives sont respectées. Nous retournons ensuite à l'algorithme principal (*algorithme* 14) pour étudier le dernier type de règles.

Le nouveau type de règles $\overline{x}_1..\overline{x}_p \Rightarrow \overline{y}_1..\overline{y}_q$ est ensuite étudié à l'aide de l'*algorithme* 19 si la méta-règle MR_9 nous l'autorise. En effet, la contrainte de la confiance fonctionne sur les motifs $\ddot{X}$ puisque le support pour ce motif possède une propriété anti-monotone.

Algorithme 19 : Étude des règles du type $X\ddot{\backslash}Y \Rightarrow \ddot{Y}$

```
1 si taille(Ẍ) > 2 alors
2     si conf(X¨\Y ⇒ Ÿ) ⩾ min_conf alors
3         si M_G(X¨\Y ⇒ Ÿ) ⩾ min_MG alors
4             R = R ∪ {X¨\Y ⇒ Ÿ}
```

Cet *algorithme* 19 commence par vérifier que le motif générant la règle est de taille strictement supérieure à 2 (*ligne 1 algorithme* 19) afin de ne pas extraire des règles en double comme nous l'avons expliqué précédemment. Si la taille du motif est suffisante alors nous vérifions ensuite la contrainte de la confiance (*ligne 2 algorithme* 19). Si la confiance est valide, nous vérifions la dernière contrainte qui est la mesure M_G (*ligne 3 algorithme* 19). Si la règle $\ddot{X} \Rightarrow \ddot{Y}$ possède une valeur de M_G suffisante, alors elle est ajoutée à l'ensemble des règles valides R. Dans cette étude, nous n'avons pas besoin de vérifier que le support de la règle est valide puisque cette contrainte est déjà vérifiée lors de l'extraction des motifs raisonnablement fréquents. En effet, durant la recherche des motifs raisonnablement fréquents, nous avons ajouté la contrainte $min_{s\ddot{u}p}$ afin de vérifier le support des motifs $\ddot{X}$.

La dernière étape de l'*algorithme* 14 consiste à retourner l'ensemble R des règles valides (*ligne 15*).

4.6 Exemple

Le tableau 4.2 rappelle les données que nous allons utiliser pour notre exemple. Nous prenons les mêmes valeurs pour le support et la confiance que celles utilisées dans les exemples précédents. Les paramètres sont donc les suivants : 0,25 pour le support minimum et pour le support minimum du motif négatif, 0,90 pour le support maximum, 0,80 pour la confiance minimum et 0,50 pour la valeur de la mesure M_G.

A	B	C	D	E
1	0	1	1	0
0	1	1	0	1
1	1	1	0	0
0	1	0	0	1

Tableau 4.2 – Exemple fil-rouge

Nous rappelons que l'algorithme se déroule en 3 étapes. La première étape consiste à rechercher les motifs raisonnablement fréquents. Nous cherchons ensuite les motifs négatifs minimaux raisonnablement fréquents. La dernière étape concerne la génération des règles valides. Nous commençons donc par chercher les motifs raisonnablement fréquents.

➤ Étape 1 : Recherche des motifs raisonnablement fréquents

Nous débutons en récupérant les items i contenus dans la base de données dont le support est inférieur ou égal au support maximum et dont le support de l'item négatif $\overline{i}$ est également inférieur ou égal au support maximum. Le tableau 4.3 fournit le support des items de la base ainsi que le support de l'item négatif.

Item	Support	Support item négatif	Item	Support	Support item négatif
A	0,50	0,50	B	0,75	0,25
C	0,75	0,25	D	0,25	0,75
E	0,50	0,50			

Tableau 4.3 – Items de taille 1

L'ensemble $\mathcal{I}$ des items contenus dans le tableau 4.3 respecte bien la contrainte du support maximum $(0,90)$ et la contrainte du support maximum pour le motif négatif $(0,90)$. Par ailleurs, l'ensemble des items respecte la contrainte du support minimum $(0,25)$ et la contrainte du support minimum pour le motif négatif $(0,25)$, ils sont par conséquent ajoutés à l'ensemble RF des motifs raisonnablement fréquents. Les items sont ensuite combinés à l'aide de la fonction *candidats* d'`Apriori` pour générer les candidats de taille 2. Le support ainsi que le support de l'item négatif sont ensuite vérifiés. Le

résultat est visible dans le tableau 4.4.

2-Motif	Support	Support motif négatif	2-Motif	Support	Support motif négatif
AB	0,25	0	**AC**	0,50	0,25
AD	0,25	0,50	AE	0	0
BC	0,50	0	BD	0	0
BE	0,50	0,25	**CD**	0,25	0,25
CE	0,25	0	DE	0	0,25

Tableau 4.4 – Candidats de taille 2

Les motifs AC, AD, BE et CD vérifient les deux contraintes suivantes : celle du support ainsi que celle du support du motif négatif. Ils sont donc ajoutés à l'ensemble des motifs raisonnablement fréquents. La génération des candidats se poursuit et va engendrer uniquement le motif ACD possédant un support à 0,25 et un support négatif à 0,25. Le motif ACD est donc raisonnablement fréquent. Le tableau 4.5 récapitule l'ensemble des motifs raisonnablement fréquents trouvés dans cet exemple.

Motif	Support	Support motif négatif	Motif	Support	Support motif négatif
A	0,50	0,50	AC	0,50	0,25
ACD	0,25	0,25	AD	0,25	0,50
B	0,75	0,25	BE	0,50	0,25
C	0,75	0,25	CD	0,25	0,25
D	0,25	0,75	E	0,50	0,50

Tableau 4.5 – Ensemble RF des motifs raisonnablement fréquents extraits

➤ Étape 2 : Recherche des motifs négatifs minimaux raisonnablement fréquents

La seconde étape de l'algorithme consiste à rechercher l'ensemble des motifs négatifs minimaux raisonnablement fréquents. Dans cet exemple, la recherche est rapide puisque dès l'initialisation des motifs négatifs minimaux raisonnablement fréquents, nous obtenons l'ensemble des motifs négatifs minimaux raisonnablement fréquents. En effet, lors de l'initialisation, nous cherchons les négations d'items tel que le support de cette négation soit supérieur ou égal au seuil du support minimum et inférieur ou égal au seuil du support maximum. Les résultats sont visibles dans le tableau 4.6.

Motif	Support	Motif	Support	Motif	Support
$\overline{A}$	0,50	$\overline{B}$	0,25	$\overline{C}$	0,25
$\overline{D}$	0,75	$\overline{E}$	0,50		

TABLEAU 4.6 – Ensemble $NMRF$ des motifs négatifs minimaux raisonnablement fréquents extraits

➤ Étape 3 : Génération des règles valides

Une fois les motifs raisonnablement fréquents et les motifs négatifs minimaux raisonnablement fréquents trouvés, la dernière phase de l'algorithme peut commencer. En effet, la recherche des règles se fait à partir des motifs raisonnablement fréquents, et la contrainte de minimalité pour les règles négatives est vérifiée grâce aux motifs négatifs minimaux raisonnablement fréquents. Prenons le motif raisonnablement fréquent ACD et cherchons les règles valides. La première étape de la recherche des règles va être de rechercher les règles possédant un seul item en conclusion. Tout d'abord, nous commençons par calculer la confiance afin de connaître les types de règles à étudier :

$$conf(CD \Rightarrow A) = \frac{sup(ACD)}{sup(CD)} = \frac{0{,}25}{0{,}25} = 1$$

$$conf(AD \Rightarrow C) = \frac{sup(ACD)}{sup(AD)} = \frac{0{,}25}{0{,}25} = 1$$

$$conf(AC \Rightarrow D) = \frac{sup(ACD)}{sup(AC)} = \frac{0{,}25}{0{,}50} = 0,50$$

Les règles $CD \Rightarrow A$ et $AD \Rightarrow C$ possèdent une confiance égale à 1. De plus, la confiance de ces règles est bien supérieure au maximum entre $\frac{1}{2}$ et le support de leur conclusion (*sup(A) pour la première et sup(C) pour la seconde*). Par conséquent, nous étudions les règles de type $X \backslash Y \Rightarrow Y$ et $\overline{X \backslash Y} \Rightarrow \overline{Y}$ pour ces deux combinaisons. Quant à la confiance de la règle $AC \Rightarrow D$, elle n'est pas supérieure au maximum entre $\frac{1}{2}$ et le support de leur conclusion et elle n'est pas non plus inférieure au minimum entre $\frac{1}{2}$ et le support de sa conclusion. Par conséquent, pour cette combinaison nous n'étudions ni les règles du type $X \backslash Y \Rightarrow Y$ et $\overline{X \backslash Y} \Rightarrow \overline{Y}$ ni les règles du type $\overline{X \backslash Y} \Rightarrow Y$ et $X \backslash Y \Rightarrow \overline{Y}$ puisqu'elle se situe dans la zone inintéressante.

Pour la combinaison $CD \cup A$, nous commençons par étudier la règle positive $CD \Rightarrow A$ en vérifiant si la valeur de la confiance est valide. La confiance a déjà été calculée pour déterminer la zone d'appartenance de la règle, nous la comparons simplement à la valeur de min_{conf}. Comme le seuil est égal à 0,80, la règle est valide pour la confiance. Nous continuons en étudiant la valeur de M_G de la règle. La règle étant dans la zone attractive, nous prenons le calcul associé pour M_G :

$$M_{G_a}(CD \Rightarrow A) = \frac{conf(CD \Rightarrow A) - max(\frac{1}{2}, sup(A))}{1 - max(\frac{1}{2}, sup(A))} = \frac{1 - max(\frac{1}{2}, 0{,}50)}{1 - max(\frac{1}{2}, 0{,}50)} = \frac{0{,}50}{0{,}50} = 1$$

La valeur de M_{G_a} pour la règle $CD \Rightarrow A$ est supérieure au seuil minimum de M_G $(0, 50)$. La règle est donc valide et sera conservée. L'étude se poursuit avec la règle négative $\overline{CD} \Rightarrow \overline{A}$. Nous vérifions tout d'abord que les motifs $\overline{CD}$ et $\overline{A}$ sont des motifs négatifs minimaux. Or $\overline{CD}$ n'en est pas un (*absent du tableau 4.6*) et par conséquent

il est inutile de générer la règle négative. Nous cherchons ensuite le nouveau type de règle : $X\ddot{\setminus}Y \Rightarrow \ddot{Y}$. Nous devons d'abord vérifier que les motifs ont une taille strictement supérieure à 2. Cette vérification permet de s'assurer de ne pas générer des règles redondantes. En effet, comme nous l'avons dit précédemment, pour un motif X de taille 2, la combinaison $X\ddot{\setminus}Y \cup \ddot{Y}$ est similaire à la combinaison $\overline{X \setminus Y} \cup \overline{Y}$. ACD est un motif de taille 3 donc nous pouvons étudier cette nouvelle règle. Le support a déjà été vérifié lors de l'extraction des motifs raisonnablement fréquents. Nous étudions donc la valeur de la confiance. Si cette dernière est valide alors nous calculerons la valeur de M_G :

$$conf(C\ddot{D} \Rightarrow \ddot{A}) = \frac{sup(A\ddot{C}D)}{sup(C\ddot{D})} = \frac{0{,}25}{0{,}25} = 1$$

La confiance de la règle $C\ddot{D} \Rightarrow \ddot{A}$ est valide et est supérieure au support de la conclusion $(0{,}5)$, par conséquent on calcule M_{G_a} :

$$M_{G_a}(C\ddot{D} \Rightarrow \ddot{A}) = \frac{conf(C\ddot{D} \Rightarrow \ddot{A}) - max(\frac{1}{2}, sup(\ddot{A}))}{1 - max(\frac{1}{2}, sup(\ddot{A}))} = \frac{1 - max(\frac{1}{2}, 0{,}50)}{1 - max(\frac{1}{2}, 0{,}50)} = \frac{0{,}50}{0{,}50} = 1$$

La valeur de M_{G_a} est valide : la règle $C\ddot{D} \Rightarrow \ddot{A}$ ou encore $\overline{C}\ \overline{D} \Rightarrow \overline{A}$ est donc conservée et ajoutée à l'ensemble R.

Pour la combinaison $AD \cup C$, nous étudions tout d'abord la règle $AD \Rightarrow C$. Comme la confiance de la règle $AD \Rightarrow C$ est valide, et que celle-ci est supérieure au support de la conclusion, nous calculons la mesure M_{G_a} :

$$M_{G_a}(AD \Rightarrow C) = \frac{conf(AD \Rightarrow C) - max(\frac{1}{2}, sup(C))}{1 - max(\frac{1}{2}, sup(C))} = \frac{1 - max(\frac{1}{2}, 0{,}70)}{1 - max(\frac{1}{2}, 0{,}70)} = \frac{0{,}25}{0{,}25} = 1$$

La valeur de M_{G_a} est valide et la règle $AD \Rightarrow C$ est également conservée et ajoutée à l'ensemble R. Nous passons ensuite à l'étude de la règle $\overline{AD} \Rightarrow \overline{C}$, cependant le motif $\overline{AD}$ n'est pas minimal. Par conséquent, l'étude de cette règle s'arrête.

Nous cherchons ensuite le nouveau type de règle : $X\ddot{\setminus}Y \Rightarrow \ddot{Y}$. Le motif ACD est de taille 3 donc nous pouvons étudier la règle :

$$conf(A\ddot{D} \Rightarrow \ddot{C}) = \frac{sup(A\ddot{C}D)}{sup(A\ddot{D})} = \frac{0{,}25}{0{,}50} = 0,50$$

La confiance n'est pas assez élevée donc la règle ne sera pas conservée.

Pour la combinaison $AC \cup D$, nous étudions directement les règles du type $\ddot{X} \cup \ddot{Y}$ car la règle positive tombe dans la zone inintéressante et par conséquent $M_{G_i} = 0$. Le motif ACD étant de taille 3, nous pouvons donc étudier la valeur de la confiance :

$$conf(\ddot{AC} \Rightarrow \ddot{D}) = \frac{sup(A\ddot{C}D)}{sup(\ddot{AC})} = \frac{0{,}25}{0{,}25} = 1$$

Cette dernière est valide, par conséquent nous calculons la valeur de M_G. Comme la confiance de la règle $\ddot{AC} \Rightarrow \ddot{D}$ est supérieure au support de la conclusion $(0{,}75)$ la règle est dans la zone attractive. Nous utilisons donc la formule :

$$M_{G_a}(\ddot{AC} \Rightarrow \ddot{D}) = \frac{conf(\ddot{AC} \Rightarrow \ddot{D}) - max(\frac{1}{2}, sup(\ddot{D}))}{1 - max(\frac{1}{2}, sup(\ddot{D}))} = \frac{1 - max(\frac{1}{2}, 0{,}75)}{1 - max(\frac{1}{2}, 0{,}75)} = \frac{0{,}50}{0{,}50} = 1$$

La valeur pour M_G étant valide, la règle $\ddot{AC} \Rightarrow \ddot{D}$ est donc conservée et ajoutée à l'ensemble R.

La prochaine étape va être la recherche des règles possédant plusieurs items en conclusion. Nous ajoutons un item à la conclusion puis calculons la confiance des règles afin de connaître les types de règles à étudier :

$$conf(D \Rightarrow AC) = \frac{sup(ACD)}{sup(D)} = \frac{0{,}25}{0{,}25} = 1$$

$$conf(C \Rightarrow AD) = \frac{sup(ACD)}{sup(C)} = \frac{0{,}25}{0{,}75} = \frac{1}{3} \simeq 0.33$$

$$conf(A \Rightarrow CD) = \frac{sup(ACD)}{sup(A)} = \frac{0{,}25}{0{,}50} = 0,50$$

Seule la règle $D \Rightarrow AC$ possède une confiance supérieure au maximum entre $\frac{1}{2}$ et le support de sa conclusion D. Les autres règles se situent dans la zone inintéressante : $min(\frac{1}{2},\ sup(AD)) = 0,25 \leqslant conf(C \Rightarrow AD) \simeq 0,33 \leqslant max(\frac{1}{2},\ sup(AD)) = 0,50$ et $min(\frac{1}{2},\ sup(CD)) = 0,25 \leqslant conf(A \Rightarrow CD) = 0,50 \leqslant max(\frac{1}{2},\ sup(CD)) = 0,50$. Par conséquent, nous étudions uniquement les règles $D \Rightarrow AC$ et $\overline{D} \Rightarrow \overline{AC}$. Nous commençons par l'étude de la règle $D \Rightarrow AC$. La règle vérifie la contrainte de la confiance, nous étudions donc la valeur de M_G :

$$M_{G_a}(D \Rightarrow AC) = \frac{conf(D \Rightarrow AC) - max(\frac{1}{2}, sup(AC))}{1 - max(\frac{1}{2}, sup(AC))} = \frac{1 - max(\frac{1}{2}, 0{,}50)}{1 - max(\frac{1}{2}, 0{,}50)} = \frac{0{,}50}{0{,}50} = 1$$

La règle $D \Rightarrow AC$ est conservée car la valeur de M_G est valide. Nous passons ensuite à l'étude de la règle $\overline{D} \Rightarrow \overline{AC}$. L'étude de cette règle s'arrête car $\overline{AC}$ n'est pas un motif négatif minimal.

Étudions maintenant le nouveau type de règle pour les trois combinaisons. Commençons par l'étude de la règle $\ddot{D} \Rightarrow \ddot{AC}$. La méta-règle de la confiance MR_9 nous empêche de l'étudier car la règle $\ddot{AD} \Rightarrow \ddot{C}$ n'est pas valide pour la confiance. Comme $\ddot{AD} \Rightarrow \ddot{C}$ ne possède pas une confiance suffisante et que l'inégalité suivante existe : $conf(\ddot{AD} \Rightarrow \ddot{C}) \geqslant conf(\ddot{D} \Rightarrow \ddot{AC})$, la confiance de la règle $\ddot{D} \Rightarrow \ddot{AC}$ ne peut pas être valide.

Pour la règle $\ddot{C} \Rightarrow \ddot{AD}$, la méta-règle nous autorise à l'étudier car les règles $\ddot{CD} \Rightarrow \ddot{A}$ et $\ddot{AC} \Rightarrow \ddot{D}$ ont une confiance valide. Calculons maintenant la confiance de cette règle :

$$conf(\ddot{C} \Rightarrow \ddot{AD}) = \frac{sup(\ddot{ACD})}{sup(\ddot{C})} = \frac{0{,}25}{0{,}25} = 1$$

La confiance de la règle est valide. La règle est également dans la zone attractive, nous continuons donc en calculant sa valeur pour M_G avec la formule :

$$M_{G_a}(\ddot{C} \Rightarrow \ddot{A}D) = \frac{conf(\ddot{C} \Rightarrow \ddot{AD}) - max(\frac{1}{2}, sup(\ddot{C}))}{1 - max(\frac{1}{2}, sup(\ddot{C}))} = \frac{1 - max(\frac{1}{2}, 0{,}25)}{1 - max(\frac{1}{2}, 0{,}25)} = \frac{0{,}50}{0{,}50} = 1$$

M_G étant valide, la règle $\ddot{C} \Rightarrow \ddot{AD}$ est ajoutée à l'ensemble R des règles valides.

Et enfin pour la règle $\ddot{A} \Rightarrow \ddot{CD}$, notre algorithme s'arrête à la vérification de la méta-règle MR_9 car la confiance de la règle $\ddot{AD} \Rightarrow \ddot{C}$ n'est pas valide.

Le tableau 4.7 expose les règles extraites par notre algorithme.

Règle	Confiance	M_G	Règle	Confiance	M_G
$A \Rightarrow C$	1	1	$AD \Rightarrow C$	1	1
$CD \Rightarrow A$	1	1	$D \Rightarrow A$	1	1
$D \Rightarrow AC$	1	1	$D \Rightarrow C$	1	1
$E \Rightarrow B$	1	1			
$\overline{B} \Rightarrow \overline{E}$	1	1	$\overline{C} \Rightarrow \overline{A}$	1	1
$\overline{A}\,\overline{C} \Rightarrow \overline{D}$	1	1	$\overline{C} \Rightarrow \overline{A}\,\overline{D}$	1	1
$\overline{C}\,\overline{D} \Rightarrow \overline{A}$	1	1			

TABLEAU 4.7 – Règles extraites sur la base d'exemple par notre algorithme classées par type de règles

En conclusion, notre algorithme génère 12 règles au total : 7 règles positives $X \backslash Y \Rightarrow Y$, 0 règle négative du type $\overline{X \backslash Y} \Rightarrow Y$, 0 règle négative du type $X \backslash Y \Rightarrow \overline{Y}$, 2 règles négatives du type $\overline{X \backslash Y} \Rightarrow \overline{Y}$ et 3 règles du nouveau type $X \ddot{\backslash} Y \Rightarrow \ddot{Y}$. Le *chapitre 6* contient une analyse sur les règles extraites afin de comparer cet algorithme aux algorithmes d'`Apriori`, de [Wu et al., 2004], de [Antonie and Zaïane, 2004] et de [Cornelis et al., 2006].

4.7 Conclusion

Dans ce chapitre, nous avons introduit un nouvel algorithme pour extraire plus efficacement les règles d'association positives et négatives d'une base de données. Bien que reposant sur l'algorithme fondateur `Apriori`, notre approche est différente de celles présentes dans la littérature. Notre algorithme essaye de répondre aux deux problématiques présentes dans les autres approches, à savoir l'extraction d'un trop grand nombre de règles inintéressantes, ainsi qu'un parcours non optimisé de recherche des règles.

Notre extraction repose sur un nouveau type de motifs, à savoir les motifs raisonnablement fréquents. L'avantage de ces motifs par rapport aux motifs fréquents repose notamment sur l'élimination des motifs omniprésents. Ces derniers entraînent soit la génération de règles dont la confiance est invalide, soit la génération de règles dont l'écart à l'indépendance est jugé trop faible. Par conséquent, les motifs raisonnablement fréquents permettent d'éliminer dès la première étape de l'algorithme ces deux types de règles inintéressantes sans avoir à les étudier. L'utilisation de la mesure M_G, plus sélective que les mesures utilisées par [Wu et al., 2004] et [Antonie and Zaïane, 2004], a également permis d'éliminer un autre type de règles non pertinentes. En effet, les règles possédant un écart trop faible par rapport à l'équilibre sont également éliminées.

Notre algorithme va également cibler les règles potentiellement intéressantes en fonction de la zone d'appartenance de la règle positive $X \Rightarrow Y$ et va donc permettre d'étudier uniquement la moitié des règles. L'utilisation des méta-règles dégagées dans le chapitre 3 permet d'inférer la non validité des règles $Y \Rightarrow X$ et $\overline{X} \Rightarrow \overline{Y}$ à partir de l'intérêt de la règle $X \Rightarrow Y$, mais également des règles $\overline{Y} \Rightarrow X$ et $\overline{X} \Rightarrow Y$ à partir de l'intérêt de la règle $X \Rightarrow \overline{Y}$.

Le prochain chapitre va présenter les expérimentations que nous avons réalisées sur différentes bases de données, afin d'évaluer notre algorithme et le comparer avec les autres approches de la littérature.

Chapitre 5

Expérimentations quantitatives

Sommaire

5.1 Introduction

Dans ce chapitre nous allons comparer notre algorithme avec les autres algorithmes que nous avons étudiés, à savoir `Apriori`, [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006]. Nous commençons par présenter le logiciel `Weka` dans lequel nous avons implémenté les différents algorithmes puis le format de base de données utilisé par celui-ci. Nous mesurons ensuite l'impact des différents paramètres de notre algorithme. Pour se faire, nous faisons varier indépendamment chacun des paramètres et regardons

le nombre de règles extraites ainsi que les temps d'extraction. Nous mesurons ensuite l'impact des différentes améliorations que nous avons proposées dans notre algorithme. Nous commençons par regarder l'impact d'un support maximum, puis celui de l'utilisation de la mesure M_G en comparaison au facteur de certitude utilisé par [Wu et al., 2004]. Nous analysons ensuite l'impact des méta-règles puis terminons par l'application de la contrainte de minimalité sur les motifs négatifs. La dernière partie de ce chapitre concerne une comparaison quantitative de notre algorithme sur différentes bases de données par rapport aux autres algorithmes d'extraction de règles d'association positives et négatives que nous avons présentés dans le chapitre 2.

5.2 Weka

`Weka` [Hall et al., 2009] est un logiciel open source développé en java à l'université de Waikato en Nouvelle-Zélande. Il contient de nombreux algorithmes utilisés dans le domaine de l'extraction de connaissances à partir des données. Il permet notamment d'effectuer les tâches suivantes : prétraitement des données, classification, clustering, extraction des règles d'association et visualisation. Son code ouvert et sa modularité facilitent l'intégration de nouveaux algorithmes et de nouveaux modules d'analyse, ce qui fait de lui un logiciel très populaire dans le domaine de la fouille de données. Depuis sa création, `Weka` a été téléchargé plus de six millions de fois sur le site de Sourceforge.

Pour les règles d'association, `Weka` contient notamment une implémentation de l'algorithme `Apriori`. Cependant, à l'heure actuelle aucun algorithme pour l'extraction de règles d'association positives et négatives n'est présent. Nous avons donc choisi d'ajouter à la panoplie d'algorithmes déjà présents, l'ensemble des méthodes évoquées dans ce manuscrit. Les méthodes de [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006] ainsi que notre algorithme ont donc été implémentés en tant que paquets `Weka` et peuvent donc s'installer facilement à l'aide du gestionnaire de paquets internes au logiciel.

5.3 Bases de données

`Weka` est optimisé pour travailler avec son propre format de fichiers : les fichiers `ARFF`. Un fichier `ARFF` (*Attribute-Relation File Format*) est un fichier texte décrivant une liste de transactions partageant un ensemble d'attributs. La figure 5.1 représente l'exemple fil-rouge de la thèse dans le format `ARFF`.

Comme nous pouvons le voir sur la figure 5.1, un fichier `ARFF` est composé de deux sections distinctes. La première section est une entête qui contient les informations sur les données. Parmi les informations renseignées, nous retrouvons le nom de la base de données, une liste d'attributs (*correspondant au nom des colonnes*), et le type de données pour chaque attribut. Tous les types de données peuvent être représentés : chaînes de caractères, entiers, réels, numériques (*nombres entiers ou réels*), dates. Dans le cas des variables nominales, l'ensemble des valeurs existantes doit être spécifié. Dans notre exemple, les données binaires sont représentées comme des variables nominales en spécifiant l'ensemble des valeurs existantes : {0,1}. La seconde section contient les données. Chaque ligne représente une transaction et chaque attribut est séparé par une virgule. Par ailleurs, les lignes commençant par % correspondent à des commen-

```
% 1. Title : Exemple fil-rouge
%
% 2. Sources :
%     (a) Creator : Papon PA
%     (b) Date : February, 2012
%
% 3. Number of Instances : 4
%
% 4. Number of Attributes : 5

% 1ère section
@RELATION exemple

@ATTRIBUTE A     {0, 1}
@ATTRIBUTE B     {0, 1}
@ATTRIBUTE C     {0, 1}
@ATTRIBUTE D     {0, 1}
@ATTRIBUTE E     {0, 1}

% 2ème section
@DATA
1,0,1,1,0
0,1,1,0,1
1,1,1,0,0
0,1,0,0,1
```

FIGURE 5.1 – Fichier `ARFF` représentant notre exemple fil-rouge

taires. En général, les commentaires se trouvent au début du fichier et fournissent un ensemble de renseignements annexes permettant de mieux comprendre les données. Ici nous avons renseigné d'où provenait la base ainsi que le nombre d'instances et d'attributs mais nous aurions pu dans le cas d'une base de données réelle, décrire les attributs.

Afin de tester les différents algorithmes, nous avons récupéré huit bases de données au format `ARFF` sur le site internet de l'UCI Machine Learning Repository [UCI, 2015]. L'ensemble des bases de données sélectionnées est représenté dans la table 5.1. Les données continues sont discrétisées en trois intervalles de tailles égales.

Le tableau 5.1 décrit, pour chaque base, le nombre de transactions, le nombre d'attributs et le nombre d'items. Nous faisons une distinction entre le nombre d'attributs qui correspond au nombre de colonnes et le nombre d'items qui correspond au nombre de valeurs que peuvent prendre les différents attributs.

Pour choisir les bases, nous avons essayé de faire varier le nombre de transactions et le nombre d'items afin d'avoir des expérimentations plus hétérogènes. Cependant, les bases sélectionnées sont toutes de petites tailles. Nous avons formulé l'hypothèse que les algorithmes d'extraction de règles d'association positives et négatives prenaient plus de temps à s'exécuter que les algorithmes d'extraction de règles d'association positives.

	transactions	attributs	items
Abalone	4 177	9	27
Contraceptive Method Choice (CMC)	1 473	10	31
Ecoli	336	8	29
Iris	150	5	15
Nursery	12 960	9	32
Solar Flare 2	1 066	13	48
Statlog (Heart)	270	14	41
Tic-Tac-Toe Endgame (TTTE)	958	10	29

TABLEAU 5.1 – Description des bases de données

Nous avons donc éliminé de l'analyse les bases où `Apriori` mettait plus de huit heures à restituer les règles pour un support minimum de 0,01.

En outre, afin de gagner en lisibilité sur les tableaux et les figures, nous utilisons les sigles `CMC` et `TTTE` pour mentionner les bases `Contraceptive Method Choice` et `Tic-Tac-Toe Endgame`. Dans la prochaine section, nous allons mesurer l'impact de chaque paramètre sur notre algorithme.

5.4 Impact des différents paramètres

Pour mesurer l'incidence de chaque paramètre de l'algorithme, nous faisons varier un à un ses paramètres. Nous avons donc cinq paramètres à faire varier : le support minimum min_{sup}, le support maximum max_{sup}, le support minimum du motif négatif $\ddot{X}$ $min_{s\ddot{u}p}$, la confiance minimum min_{conf} et la valeur de M_G minimum min_{M_G}.

L'expérimentation de référence sera celle possédant ces valeurs : $min_{sup} = 0,01$, $max_{sup} = 0,80$, $min_{s\ddot{u}p} = 0,01$, $min_{conf} = 0,90$ et $min_{M_G} = 0,60$. Ainsi, lorsque nous faisons varier un paramètre, les autres paramètres prendront ces valeurs. Commençons par faire varier le support minimum.

5.4.1 Impact du support minimum

La valeur que nous utilisons par défaut pour le support minimum est de 0,01. Nous allons donc faire varier le support minimum à 0,05 et 0,10. Le tableau 5.2 restitue le nombre total de règles ainsi que le temps d'exécution en secondes de l'algorithme lorsque le support minimum varie.

L'augmentation du seuil pour le support minimum entraîne une diminution du nombre de règles extraites ainsi que des temps d'extraction. Afin de visualiser plus précisément l'impact, nous avons tracé les courbes d'évolution du nombre de règles (*cf. figure* 5.2) et du temps (*cf. figure* 5.3). Chaque figure est composée de trois sous-figures afin d'avoir une meilleure représentation des différentes courbes (*afin d'éviter les problèmes d'échelle*). Attention cependant, les courbes représentant les bases de données ne se situent pas forcément dans la même sous-figure pour chaque figure. Par exemple, la

	nombre de règles			temps en secondes		
Support / $\mathcal{D}$	0,01	0,05	0,10	0,01	0,05	0,10
Abalone	11 711	8 035	5 070	0,765	0,529	0,337
CMC	979	184	45	0,511	0,16	0,101
Ecoli	801	334	136	0,191	0,131	0,079
Iris	603	456	373	0,093	0,08	0,088
Nursery	2 602	174	48	3,28	0,662	0,236
Solar Flare 2	175	68	32	0,141	0,099	0,083
Statlog (Heart)	104 529	5 490	839	57,962	1,988	0,402
TTTE	4 274	28	0	4,398	0,28	0,118

TABLEAU 5.2 – Impact du support minimum

courbe d'évolution du nombre de règles pour la base `Abalone` (*courbe rouge*) se situe dans le repère central de la figure 5.2 alors que celle du temps d'extraction se situe dans le repère de gauche pour la figure 5.3.

La modification du seuil du support minimum de 0,01 à 0,10 entraîne une forte diminution du nombre de règles pour les bases de données `TTTE` (*courbe marron*), `Statlog (Heart)` (*courbe grise*), `Nursery` (*courbe jaune*) et `CMC` (*courbe bleue*). Pour les autres bases, la diminution est moins forte. Nous pouvons également constater que pour les bases `Abalone` (*courbe rouge*), `Iris` (*courbe rose*) et `Solar Flare 2` (*courbe vert foncé*), le nombre de règles diminue presque linéairement. Par ailleurs, le nombre de règles est très important pour la base `Statlog (Heart)` pour un support de 0,01 où 104 529 règles valides sont extraites.

L'augmentation du support de 0,01 à 0,10 entraîne également une diminution du temps d'extraction plus ou moins prononcée en fonction des bases de données. Les diminutions les plus sensibles ont lieu sur les bases `Statlog (Heart)` (*courbe grise*), `TTTE` (*courbe marron*) et `Nursery` (*courbe jaune*). Globalement, les temps sont relativement rapides. En effet, ils sont de l'ordre de la seconde pour l'ensemble des expérimentations excepté pour `Statlog (Heart)` avec un support minimum à 0,01 où le temps d'extraction est de l'ordre de la minute.

En mettant côte à côte les figures d'évolution du nombre de règles et du temps d'extraction en fonction du support minimum, on remarque une certaine corrélation entre le temps d'extraction et le nombre de règles extraites.

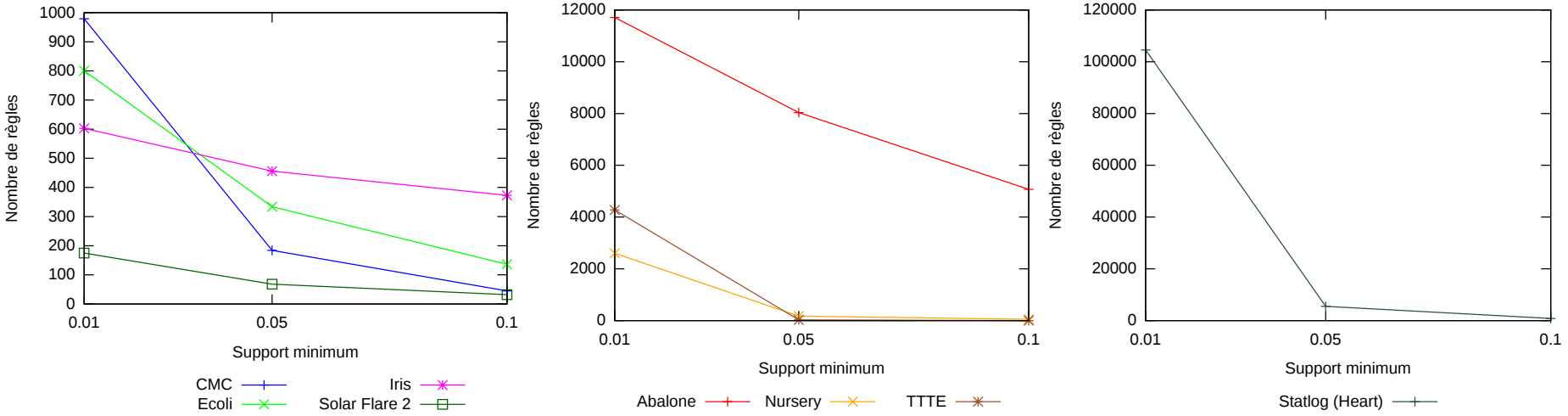

FIGURE 5.2 – Évolution du nombre de règles en fonction du support minimum

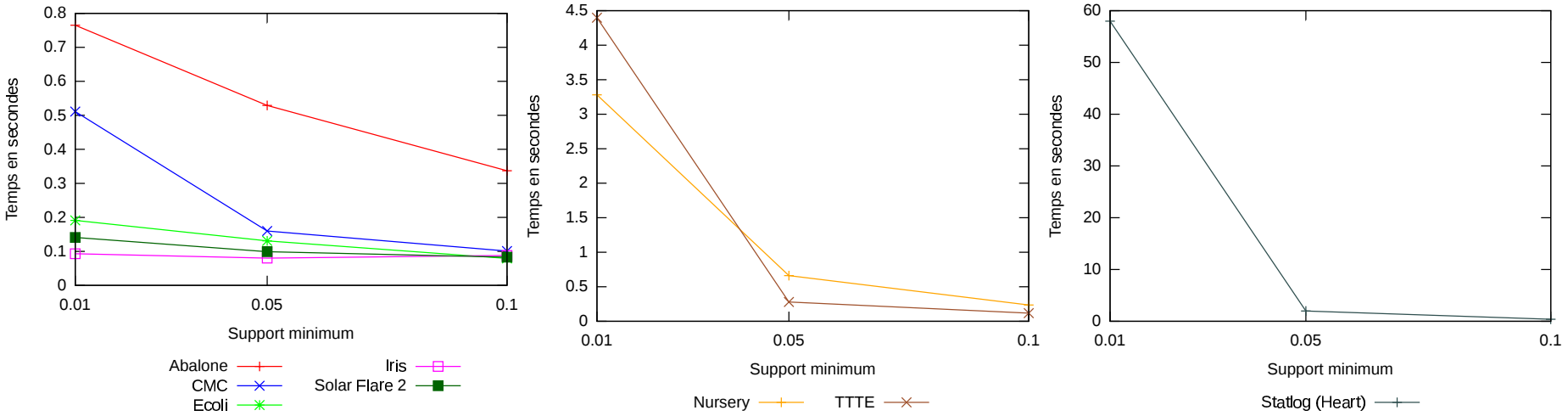

FIGURE 5.3 – Évolution du temps d'extraction en fonction du support minimum

Mesurons maintenant l'impact du support maximum sur l'algorithme d'extraction que nous proposons.

5.4.2 Impact du support maximum

La valeur que nous utilisons par défaut pour le support maximum est de 0,80. Nous allons donc faire varier le support maximum à 0,90 et 1. Le tableau 5.3 restitue le nombre de règles ainsi que le temps d'exécution en secondes de l'algorithme lorsque le support maximum varie.

	nombre de règles			temps en secondes		
$\mathcal{D}$ \ Support	0,80	0,9	1	0,80	0,90	1
Abalone	11 711	11 711	16 529	0,767	0,785	0,867
CMC	979	7 050	16 777	0,512	1,511	3,262
Ecoli	801	1 651	4 142	0,178	0,233	0,344
Iris	603	1 012	1 012	0,137	0,118	0,114
Nursery	2 602	2 602	2 625	3,396	3,269	3,278
Solar Flare 2	175	6 346	25 589	0,129	0,86	2,611
Statlog (Heart)	104 529	687 819	749 249	58,633	542,725	553,697
TTTE	4 274	4 740	4 740	4,481	4,686	4,76

TABLEAU 5.3 – Impact du support maximum

Théoriquement, l'augmentation du seuil pour le support maximum entraîne une augmentation du nombre de règles extraites ainsi que des temps d'extraction. Vérifions visuellement cette propriété à l'aide des figures 5.4 et 5.5 composée chacune de trois sous-figures (*toujours le problème d'échelle*).

L'augmentation du support maximum entraîne une forte augmentation du nombre de règles extraites pour les bases de données `Solar Flare 2` (*courbe vert foncé*), `CMC` (*courbe bleue*) et `Statlog (Heart)` (*courbe grise*). Pour les autres bases, l'augmentation est moins forte et notamment pour `Nursery` qui extrait seulement 23 règles supplémentaires en augmentant le support maximum de 0,80 à 1.

Les temps d'extraction augmentent également avec l'augmentation du seuil pour le support maximum. L'augmentation est plus forte pour les bases `Solar Flare 2` (*courbe vert foncé*), `Statlog (Heart)` (*courbe grise*) et `CMC` (*courbe bleue*). Nous remarquons que pour `Statlog (Heart)` l'ordre de grandeur change pour les temps d'extraction : l'exécution prend environ 9 minutes avec 0,90 et 1. Nous détectons également une anomalie pour les autres bases `Iris` (*courbe rose*) et `Nursery` (*courbe jaune*) où le temps d'extraction diminue très légèrement au lieu d'augmenter. Nous expliquons ceci par le fait que les expérimentations ont lieu sur un serveur où la charge peut varier d'un instant à un autre, ce qui peut entraîner un biais dans le relevé des temps.

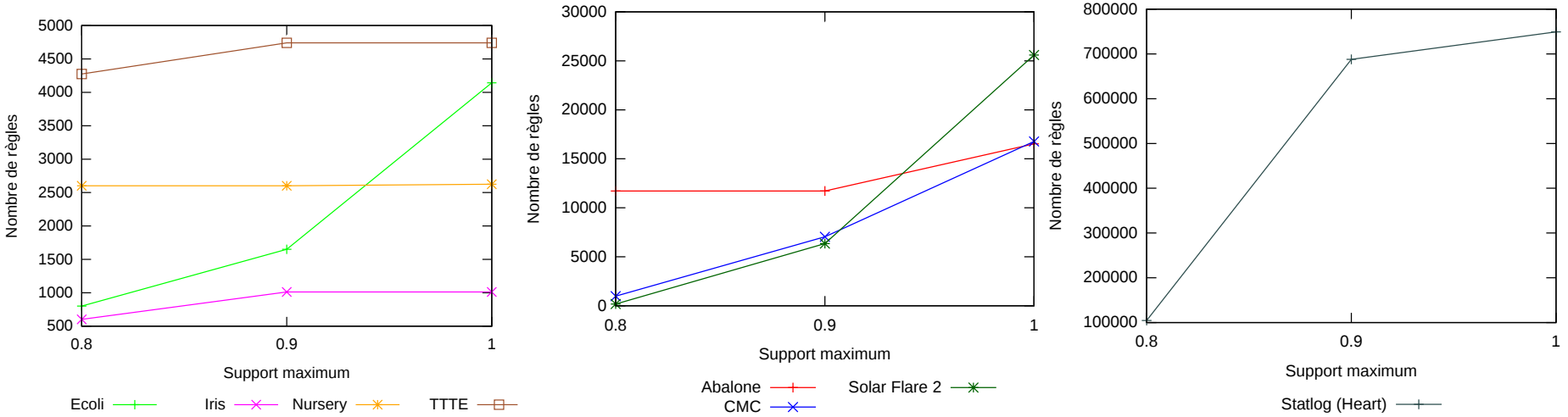

FIGURE 5.4 – Évolution du nombre de règles en fonction du support maximum

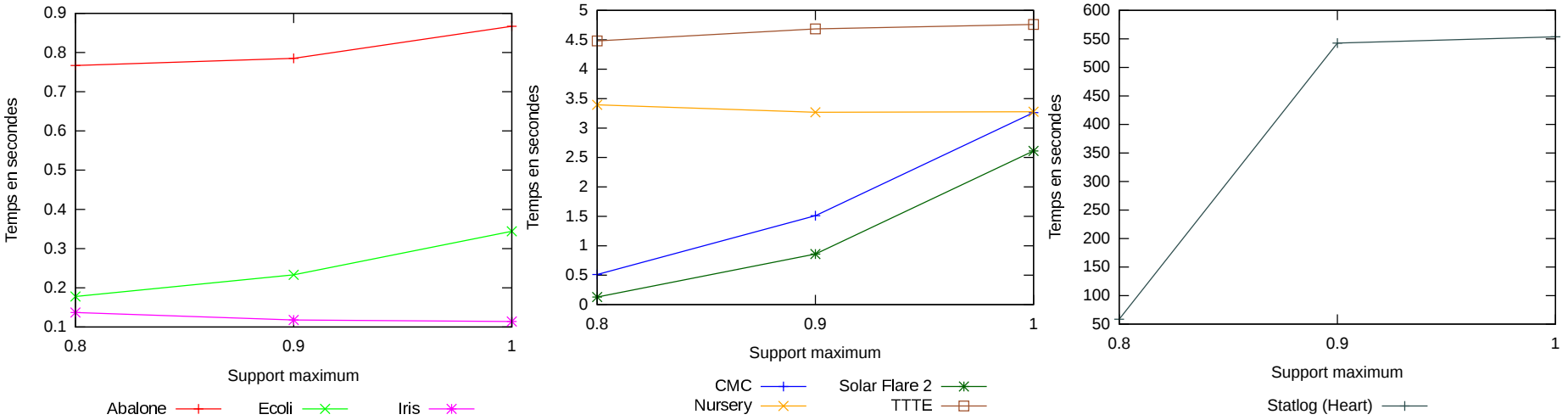

FIGURE 5.5 – Évolution du temps d'extraction en fonction du support maximum

Passons maintenant à l'impact du support minimum du motif négatif $\ddot{X}$ sur les temps d'exécution et sur le nombre de règles extraites.

5.4.3 Impact du support minimum du motif négatif

La valeur que nous utilisons par défaut pour le support minimum du motif négatif $\ddot{X}$ est de 0,01. Nous allons donc faire varier le support minimum du motif $\ddot{X}$ à 0,05 et 0,10. Le tableau 5.4 restitue le nombre total de règles ainsi que le temps d'exécution en secondes de l'algorithme lorsque le support minimum du motif négatif varie.

	nombre de règles			temps en secondes		
Support négatif / $\mathcal{D}$	0,01	0,05	0,10	0,01	0,05	0,10
Abalone	11 711	10 652	7 155	0,744	0,66	0,491
CMC	979	708	508	0,55	0,409	0,319
Ecoli	801	681	528	0,19	0,169	0,138
Iris	603	603	595	0,096	0,096	0,092
Nursery	2 602	2 602	2 602	3,301	3,336	3,293
Solar Flare 2	175	175	169	0,142	0,146	0,13
Statlog (Heart)	104 529	22 828	5 260	60,548	8,417	1,688
TTTE	4 274	3 228	1 294	4,328	3,827	2,271

TABLEAU 5.4 – Impact du support minimum du motif négatif $\ddot{X}$

L'augmentation du seuil pour le support minimum du motif négatif $\ddot{X}$ entraîne théoriquement une diminution du nombre de règles extraites ainsi que des temps d'extraction. Afin de visualiser plus précisément l'impact, nous avons tracé les figures 5.6 et 5.7.

La modification du seuil du support minimum du motif négatif $\ddot{X}$ de 0,01 à 0,10 entraîne une forte diminution du nombre de règles pour `Statlog (Heart)` (*courbe grise*) tandis que les courbes pour `Solar Flare 2` (*courbe vert foncé*), `Iris` (*courbe rose*) et `Nursery` (*courbe jaune*) n'évoluent pas ou très peu.

Pour les temps d'extraction, l'augmentation du seuil entraîne une forte diminution des temps pour `Statlog (Heart)` (*courbe grise*) tandis qu'elle est plus limitée pour les autres bases.

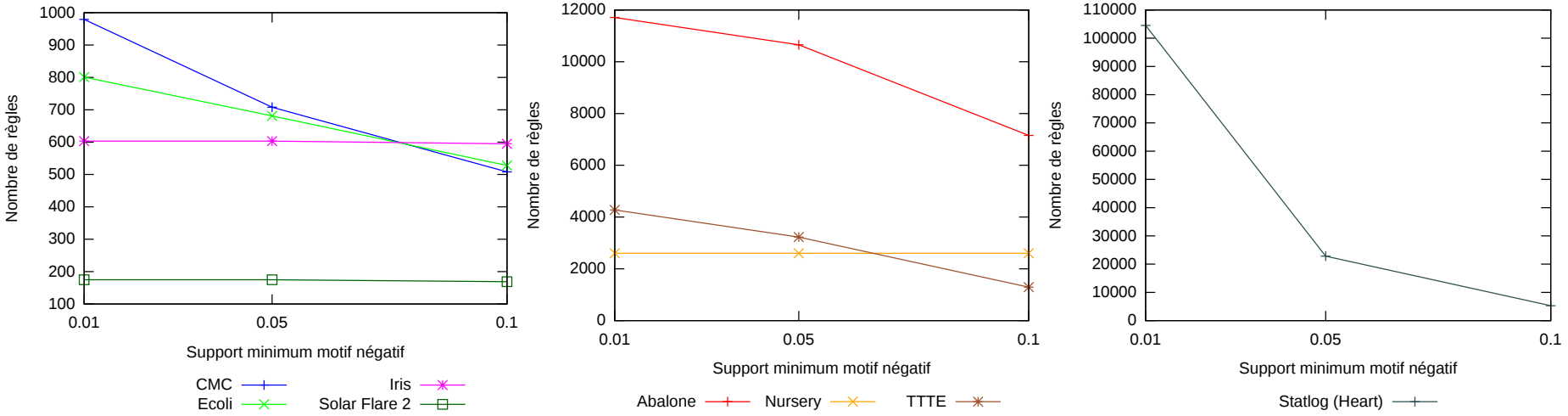

FIGURE 5.6 – Évolution du nombre de règles en fonction du support minimum du motif négatif $\ddot{X}$

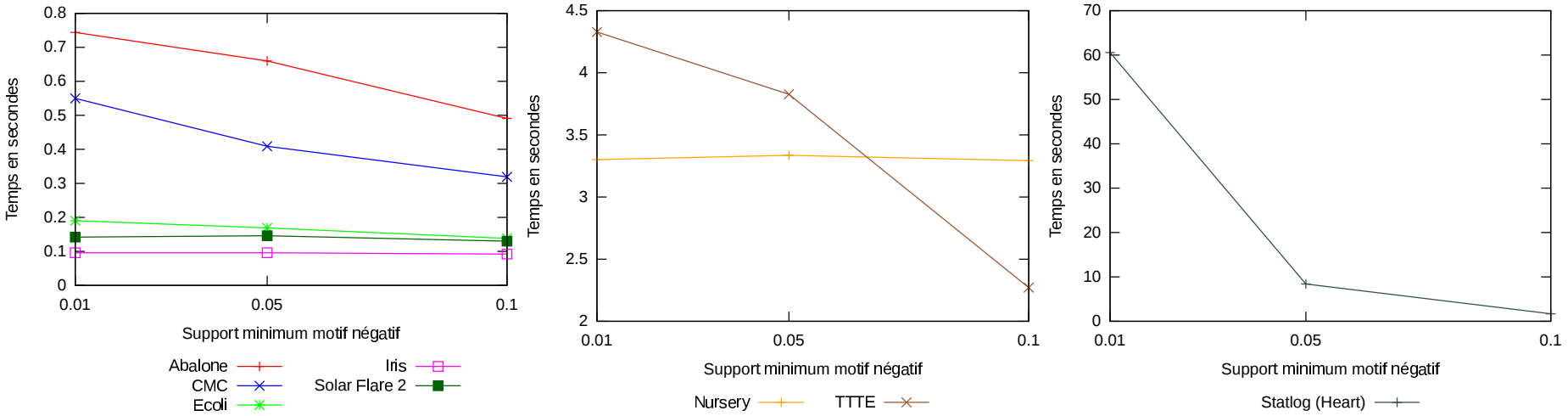

FIGURE 5.7 – Évolution du temps d'extraction en fonction du support minimum du motif négatif $\ddot{X}$

Dans la prochaine sous-section, nous vérifions l'impact du seuil de la confiance minimum sur notre algorithme.

5.4.4 Impact de la confiance minimum

La valeur que nous utilisons par défaut pour la confiance minimum est de 0,90. Nous allons donc faire varier la confiance minimum à 0,85 et 0,95. Le tableau 5.5 restitue le nombre total de règles ainsi que le temps d'exécution en secondes de l'algorithme lorsque la confiance minimum fluctue.

	nombre de règles			temps en secondes		
$\mathcal{D}$ \ Confiance	0,85	0,90	0,95	0,85	0,90	0,95
Abalone	14 840	11 711	8 032	0,812	0,753	0,781
CMC	1 280	979	423	0,499	0,493	0,484
Ecoli	982	801	477	0,196	0,189	0,185
Iris	693	603	406	0,098	0,096	0,092
Nursery	2 678	2 602	2 554	3,248	3,265	3,245
Solar Flare 2	206	175	133	0,134	0,134	0,133
Statlog (Heart)	129 458	104 529	89 240	62,827	60,684	58,75
TTTE	5 944	4 274	2 684	4,269	4,444	4,348

Tableau 5.5 – Impact de la confiance minimum

Théoriquement, l'augmentation du seuil pour la confiance entraîne une diminution du nombre de règles extraites ainsi que des temps d'extraction. Afin de vérifier cette théorie, nous avons tracé les figures 5.8 et 5.9.

L'augmentation du seuil de la confiance minimum conduit à une diminution modérée du nombre de règles extraites pour l'ensemble des bases de données, excepté pour `Nursery` (*courbe jaune*) et pour `Solar Flare 2` (*courbe vert foncé*) où l'effet est moindre.

Quant aux temps d'extraction ils ne varient pas beaucoup.

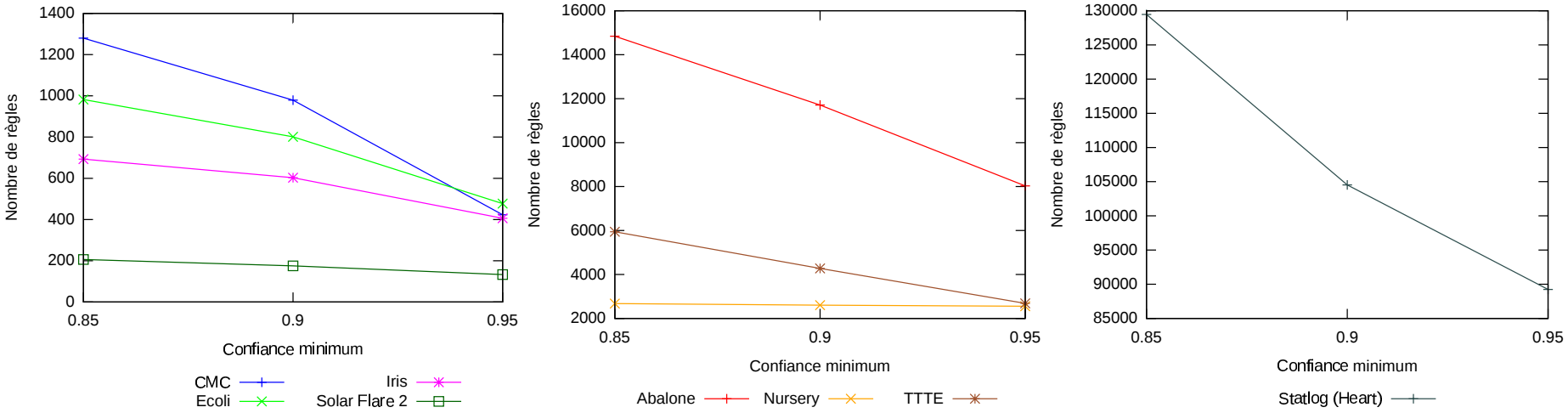

FIGURE 5.8 – Évolution du nombre de règles en fonction de la confiance minimum

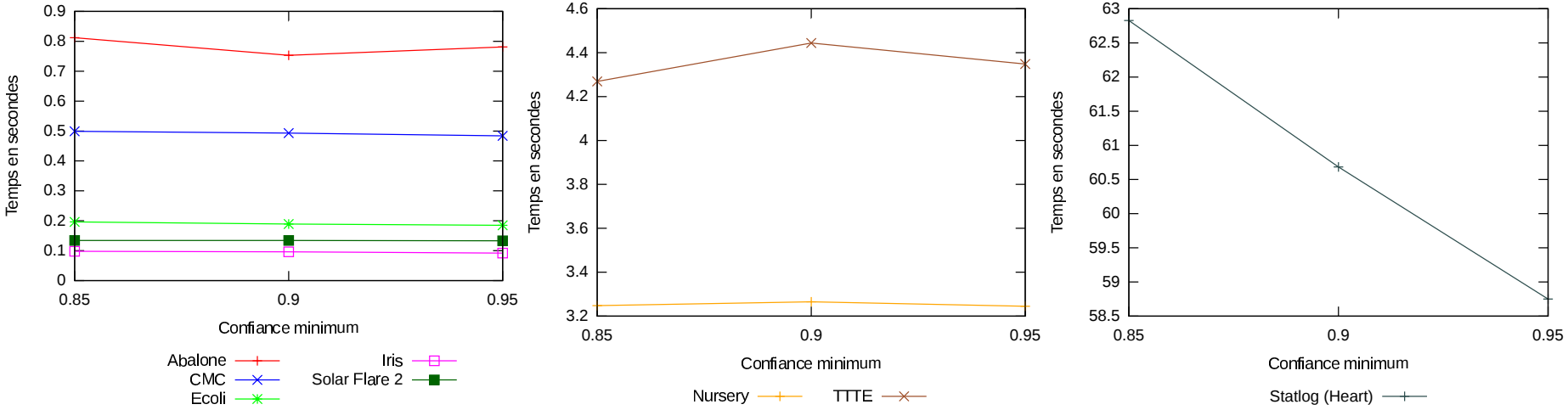

FIGURE 5.9 – Évolution du temps d'extraction en fonction de la confiance minimum

Le dernier paramètre de notre algorithme est le seuil de la mesure M_G. Nous allons maintenant étudier l'impact de ce paramètre sur les temps d'exécution et sur le nombre de règles extraites.

5.4.5 Impact de la valeur de M_G minimum

La valeur que nous utilisons par défaut pour la mesure minimum de M_G est de 0,60. Nous allons donc faire varier M_G à 0,50 et 0,70. Le tableau 5.6 restitue le nombre total de règles ainsi que le temps d'exécution en secondes de l'algorithme lorsque la valeur de M_G varie.

	nombre de règles			temps en secondes		
$\mathcal{D}$ \ M_G	0,50	0,60	0,70	0,50	0,60	0,70
Abalone	11 715	11 711	11 661	0,754	0,74	0,716
CMC	1 006	979	892	0,546	0,505	0,496
Ecoli	826	801	724	0,184	0,191	0,194
Iris	603	603	602	0,092	0,091	0,092
Nursery	2 605	2 602	2 587	3,281	3,299	3,286
Solar Flare 2	196	181	152	0,136	0,141	0,135
Statlog (Heart)	105 281	104 529	102 504	59,658	62,52	58,548
TTTE	4 286	4 274	4 274	4,492	4,515	4,765

TABLEAU 5.6 – Impact de la valeur de M_G minimum

Théoriquement, l'augmentation du seuil pour M_G entraîne une diminution du nombre de règles extraites ainsi que des temps d'extraction. Afin de vérifier cette théorie, nous avons tracé les figures 5.10 et 5.11.

L'augmentation du seuil de M_G conduit à une très légère diminution du nombre de règles extraites pour l'ensemble des bases de données, excepté pour `Iris` qui n'évolue que d'une règle.

Concernant les temps d'extraction, ils varient assez peu. Les variations notables proviennent des bases `TTTE` (*courbe marron*) et `Statlog (Heart)` (*courbe grise*) où des anomalies sont détectées. Cela est peut-être dû à la charge du serveur au moment des expérimentations qui peut entraîner un biais dans le relevé des temps puisque le temps devrait théoriquement diminuer.

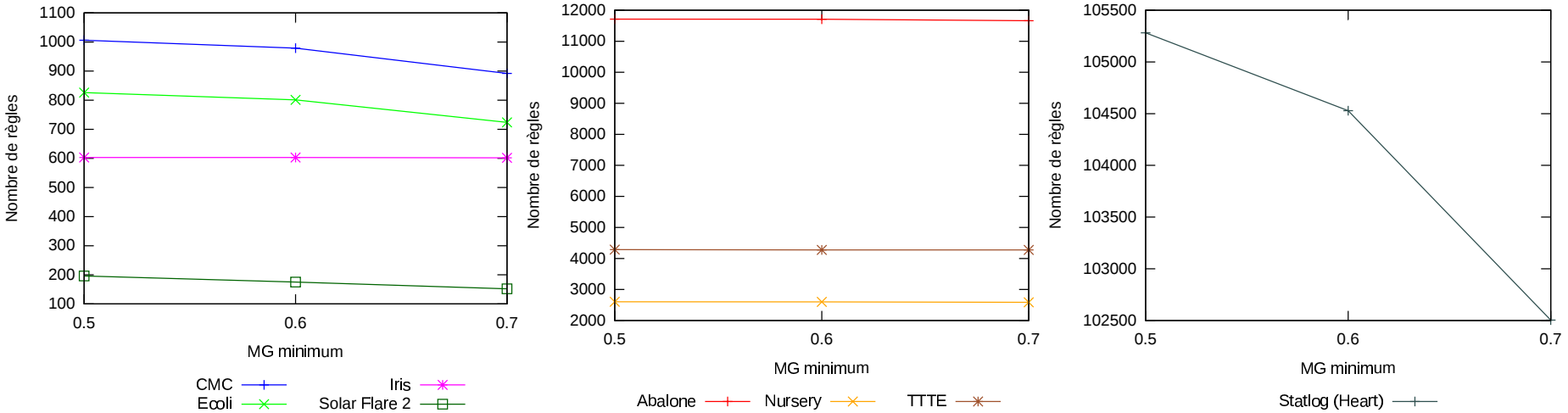

FIGURE 5.10 – Évolution du nombre de règles en fonction de la valeur de M_G minimum

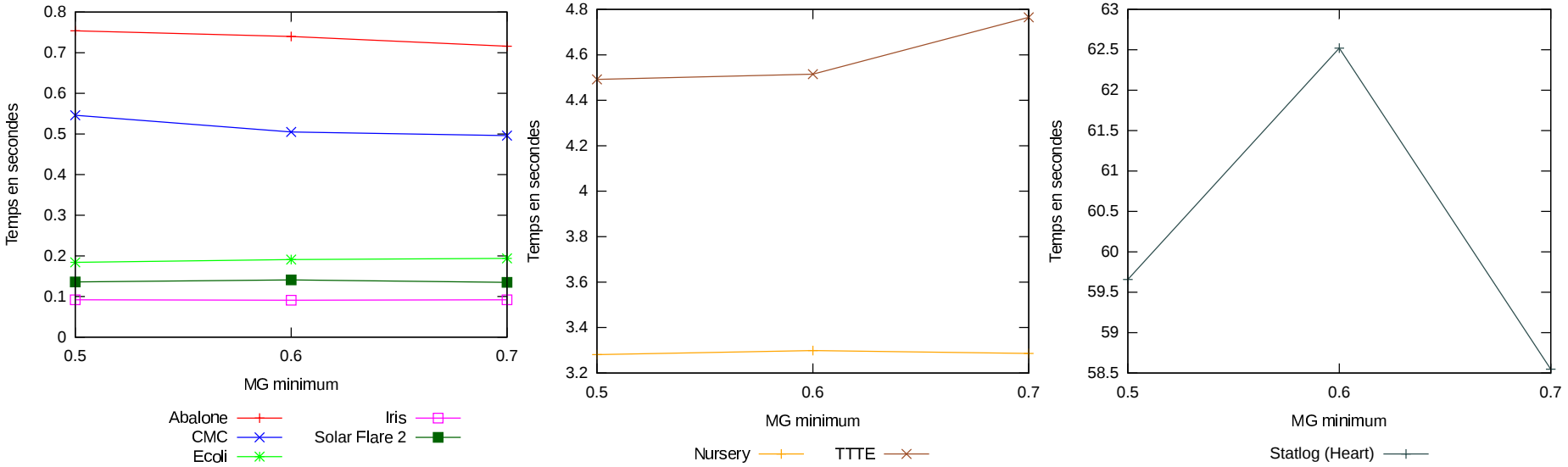

FIGURE 5.11 – Évolution du temps d'extraction en fonction de la valeur de M_G minimum

5.4.6 Synthèse

Pour conclure cette partie, nous avons fait varier les cinq paramètres présents dans notre algorithme afin de mesurer l'impact de chacun sur le nombre de règles extraites ainsi que sur les temps d'extraction.

Pour le nombre de règles extraites, tous les paramètres n'ont pas la même influence. En effet, certains paramètres ont un impact très important pour l'ensemble des bases étudiées (*ou presque*) : le support minimum et le support maximum. D'autres paramètres ont un impact un peu moins important : le support minimum pour le motif négatif $\ddot{X}$ et la confiance minimum. Et enfin, le dernier paramètre, à savoir la valeur de M_G minimum a un impact assez limité sur l'ensemble des bases. Cela peut provenir du fait que les valeurs choisies n'étaient pas assez différentes mais également des bases étudiées. Nous pensons que ce paramètre a un impact plus important sur d'autres bases de données.

Concernant l'impact sur les temps d'extraction, les paramètres du support minimum et du support maximum doivent être choisis avec prudence car ce sont ces deux paramètres qui ont le plus d'impact sur les temps d'extraction. La confiance minimum et le support minimum pour le motif négatif $\ddot{X}$ ont un impact plus ou moins important en fonction des bases étudiées. Concernant M_G l'impact est limité comme ce fut le cas pour le nombre de règles extraites. Ici encore, nous pensons que ce paramètre a un impact plus important sur les grandes bases de données.

Dans la prochaine section, nous allons mesurer l'impact de nos améliorations.

5.5 Impact de nos améliorations

Dans cette section, nous allons mesurer l'impact des améliorations majeures que nous proposons dans notre algorithme. Nous commençons par le nombre de règles élaguées par le support maximum. Nous comparons ensuite l'utilisation de la mesure M_G par rapport au facteur de certitude utilisé par [Wu et al., 2004]. Nous analysons enfin l'impact de l'utilisation des méta-règles puis celui de l'application de la contrainte de minimalité sur les motifs négatifs.

5.5.1 Impact de l'utilisation du support maximum

Afin de mesurer l'impact du support maximum nous reprenons l'étude précédente (*cf. section* 5.4.2) en prenant en compte cette fois-ci le nombre de règles élaguées par celui-ci. Cette étude peut sembler redondante à la précédente, mais elle permet de bien mettre en avant le nombre de règles élaguées par le support maximum. Le nombre de règles au départ correspond à l'algorithme où aucun support maximum n'est appliqué, cela revient à utiliser 1 comme seuil du support maximum. Ainsi tous les motifs respecteront la contrainte du support maximum. Les temps d'extraction étant identiques à l'étude précédente nous ne les rappelons pas ici. Les résultats sont visibles dans le tableau 5.7.

La diminution du support maximum entraîne une forte augmentation du nombre de règles éliminées pour l'ensemble des bases de données excepté pour `Nursery` où le support maximum n'élague que 23 règles. Même si le nombre de règles éliminées par `Statlog`

	nombre de règles au départ	nombre de règles éliminées	
Support / $\mathcal{D}$	1	0,90	0,80
Abalone	16 529	4 818	4 818
CMC	16 777	9 727	15 798
Ecoli	4 142	2 491	3 341
Iris	1 012	0	409
Nursery	2 625	23	23
Solar Flare 2	25 589	19 243	25 414
Statlog (Heart)	749 249	61 430	644 720
TTTE	4 740	0	466

TABLEAU 5.7 – Nombre de règles élaguées par le support maximum

`(Heart)` pour un support maximum à 0,80 semble le plus important, il correspond à 86% du nombre total de règles extraites pour un support de 1, alors que pour `CMC` et `Solar Flare 2` cela représente respectivement 94% et 99% du nombre total de règles générées.

Dans la prochaine section, nous allons comparer la mesure M_G avec le facteur de certitude.

5.5.2 M_G versus facteur de certitude

Afin de comparer M_G avec le facteur de certitude fc, nous avons créé une deuxième version de notre algorithme qui utilise fc à la place de M_G. Dans ces expérimentations, nous avons réutilisé les seuils précédents, à savoir $min_{sup} = 0,01$, $max_{sup} = 0,80$, $min_{süp} = 0,01$, $min_{conf} = 0,90$ et nous faisons varier min_{M_G} et min_{fc} à 0,7, 0,6 et 0,5. Les résultats sont reportés dans le tableau 5.8.

		Abalone		CMC		Ecoli		Iris		Nursery		Solar Flare 2		Statlog (Heart)		TTTE	
	Seuil	M_G	fc	M_G	fc	M_G	fc	M_G	fc	M_G	fc	M_G	fc	M_G	fc	M_G	fc
Nombre	0,70	11661	11664	892	893	724	724	602	602	2587	2587	152	152	102504	102504	4274	4274
	0,60	11711	11714	979	980	801	801	603	603	2602	2602	175	175	104529	104530	4274	4274
	0,50	11715	11718	1006	1007	826	826	603	603	2605	2605	196	196	105281	105283	4286	4286
Temps	0,70	0,673	0,695	0,452	0,467	0,171	0,166	0,099	0,12	3,375	3,334	0,135	0,142	58,261	58,847	4,296	4,459
	0,60	0,7	0,7	0,461	0,464	0,161	0,166	0,1	0,097	3,257	3,303	0,135	0,14	58,535	59,169	4,337	4,444
	0,50	0,695	0,705	0,457	0,469	0,17	0,165	0,098	0,096	3,243	3,299	0,136	0,136	58,225	57,498	4,358	4,433

TABLEAU 5.8 – Règles extraites avec M_G et avec le facteur de certitude

L'algorithme utilisant M_G génère moins de règles que celui utilisant fc pour l'ensemble des expérimentations sur les bases `Abalone` et `CMC` et également sur la base `Statlog (Heart)` lorsque la valeur de la mesure est égale à 0,60 et 0,70. Pour le reste des expérimentations, le nombre de règles extraites est identique pour les deux versions. Ces résultats s'expliquent car la mesure M_G est plus restrictive que la mesure fc. En effet, la zone représentant l'attraction entre les motifs X et Y est plus restrictive pour M_G puisque la confiance de la règle $X \Rightarrow Y$ doit être supérieure ou égale au support de la conclusion Y (*comme pour le facteur de certitude*) afin d'être au-delà de l'indépendance, mais elle doit également être supérieure ou égale à $\frac{1}{2}$ afin d'être au-delà du point d'équilibre. Réciproquement, la zone représentant la répulsion entre les motifs X et Y est plus restrictive pour M_G puisque la confiance de la règle $X \Rightarrow Y$ doit être inférieure au support de la conclusion Y (*comme pour le facteur de certitude*) afin d'être en deçà de l'indépendance, mais elle doit également être inférieure à $\frac{1}{2}$ afin d'être en deçà du point d'équilibre.

Au niveau des temps d'exécution, la version utilisant M_G est globalement plus rapide même si la différence n'est pas significative.

Dans la prochaine section, nous allons mesurer l'impact de l'utilisation des méta-règles.

5.5.3 Impact de l'utilisation des méta-règles

Pour mesurer l'impact des méta-règles, nous avons créé deux nouvelles versions de notre algorithme. Dans la première version, nous ajoutons deux compteurs : le premier compteur prend en compte le nombre de fois où nous utilisons la méta-règle MR_4. Pour rappel, la méta-règle MR_4 correspond à :

> **MR$_4$** : $\forall X \Rightarrow Y$ avec $\frac{1}{2} < sup(X) < sup(Y)$ ou $sup(X) < \frac{1}{2} < sup(Y)$
> si $M_G(X \Rightarrow Y) < min_{M_G}$ alors $M_G(\overline{X} \Rightarrow \overline{Y}) < min_{M_G}$.

Le deuxième compteur, quant à lui, va nous permettre de compter le nombre de règles du type $\ddot{X} \Rightarrow \ddot{Y}$ que nous étudions.

Dans la deuxième version, nous désactivons les méta-règles MR_4 et MR_9, mais nous ajoutons le compteur qui permet de compter le nombre de règles du type $\ddot{X} \Rightarrow \ddot{Y}$ que nous étudions. Pour rappel, la méta-règle MR_9 correspond à :

> **MR$_9$** : $\forall (X, Y, Z)$ tel que $Y \subsetneq Z \subsetneq X \subseteq \mathcal{I}$
> si $conf(X \backslash Y \Rightarrow Y) < min_{conf}$ alors $conf(X \backslash Z \Rightarrow Z) < min_{conf}$.

L'impact de la méta-règle MR_9 correspondra donc à la différence du nombre de règles du type $\ddot{X} \Rightarrow \ddot{Y}$ étudiées entre les deux versions. En effet, notre implémentation nous empêche de procéder comme pour la méta-règle MR_4 en ajoutant directement un compteur. Nous pourrons également mesurer l'impact des méta-règles sur les temps d'extraction puisque la première version les utilise alors que la deuxième version les désactive. Dans ces expérimentations, nous avons repris les seuils précédents, à savoir $min_{süp} = min_{sup}$, $max_{sup} = 0,80$, $min_{M_G} = 0,60$ et nous faisons varier min_{sup} à 0,1, 0,05

	Support	Confiance	Abalone	CMC	Ecoli	Iris	Nursery	Solar Flare 2	Statlog (Heart)	TTTE
MR_4	0,10	0,95	0	0	0	0	0	0	0	0
		0,90	0	0	0	0	0	0	0	0
		0,85	1	0	2	0	0	0	1	0
	0,05	0,95	0	0	0	0	0	0	0	0
		0,90	0	0	0	0	0	0	0	0
		0,85	1	0	2	0	0	0	1	0
	0,01	0,95	0	0	0	0	0	0	0	0
		0,90	0	0	0	0	0	0	0	0
		0,85	1	0	2	0	0	0	1	0
MR_9	0,10	0,95	6 209	107	325	335	25	68	4 416	222
		0,90	5 851	105	317	305	25	68	4 383	222
		0,85	5 157	104	316	284	25	68	4 287	222
	0,05	0,95	17 780	1 397	1 246	478	826	264	51 247	4 304
		0,90	16 875	1 379	1 223	440	826	263	50 874	4 304
		0,85	15 528	1 363	1 209	416	826	256	50 078	4 302
	0,01	0,95	34 162	28 406	4 862	711	63 564	1 400	2 130 389	312 304
		0,90	32 616	28 296	4 769	640	63 584	1 386	2 123 766	312 246
		0,85	30 760	28 155	4 688	606	63 584	1 362	2 108 490	312 016

TABLEAU 5.9 – Nombre de règles non étudiées grâce aux méta-règles MR4 et MR9

et 0,01 ainsi que min_{conf} à 0,95, 0,90 et 0,85. Le tableau 5.9 restitue le nombre de règles non étudiées grâce aux méta-règles MR_4 et MR_9 tandis que le tableau 5.10 fournit les vitesses d'extraction avec et sans les méta-règles.

Concernant la méta-règle MR_4, nous remarquons que le nombre de règles élaguées de l'étude est très faible. En effet, pour un seuil de la confiance à 0,95 et 0,90 et quel que soit le seuil du support, aucune règle n'est élaguée de l'étude grâce à cette méta-règle. Pour une confiance de 0,85 et quel que soit le seuil du support, seulement une règle est élaguée dans `Abalone`, deux dans `Ecoli` et une sur `Statlog (Heart)`. Pour les autres bases, aucune règle n'est élaguée. Ce comportement provient du fait que la méta-règle est difficilement applicable. En effet, pour que la méta-règle puisse s'appliquer il faut que le support de la conséquence Y soit supérieur à $\frac{1}{2}$ ou que le support de la prémisse X et celui de la conséquence Y soient supérieurs à $\frac{1}{2}$, mais également que $M_G(X \Rightarrow Y) < min_{M_G}$. Par conséquent, il est difficile de juger de l'intérêt de la règle $\overline{X} \Rightarrow \overline{Y}$ à partir de la règle $X \Rightarrow Y$ ou de celui de la règle $\overline{X} \Rightarrow Y$ à partir de la règle $X \Rightarrow \overline{Y}$.

En ce qui concerne la méta-règle MR_9, celle-ci est beaucoup plus efficace. En effet, elle permet d'éviter d'étudier plusieurs milliers de règles $\ddot{X} \Rightarrow \ddot{Y}$ et notamment 2 130 389 règles pour un support à 0,01 et une confiance à 0,95 pour la base `Statlog (Heart)`. Cette efficacité n'est pas étonnante puisque cette méta-règle est équivalente à la propriété de la confiance présente dans l'algorithme `Apriori` mais appliquée au nouveau type de règles $\ddot{X} \Rightarrow \ddot{Y}$.

		Abalone		CMC		Ecoli		Iris		Nursery		Solar Flare 2		Statlog (Heart)		TTTE	
Support	Confiance	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec
0,10	0,95	0,533	0,368	0,129	0,147	0,098	0,145	0,089	0,112	0,376	0,414	0,133	0,469	0,404	0,31	0,166	0,165
	0,90	0,56	0,346	0,127	0,13	0,141	0,124	0,116	0,187	0,377	0,392	0,131	0,089	0,398	0,318	0,185	0,171
	0,85	0,444	0,47	0,126	0,139	0,103	0,119	0,118	0,119	0,444	0,396	0,133	0,091	0,378	0,322	0,183	0,181
0,05	0,95	2,087	0,667	0,236	0,228	0,179	0,178	0,13	0,128	1,093	1,086	0,182	0,109	5,767	1,542	0,408	0,393
	0,90	1,253	0,731	0,235	0,243	0,2	0,235	0,268	0,136	1,199	1,331	0,18	0,142	5,744	1,595	0,408	0,399
	0,85	1,372	0,795	0,235	0,242	0,179	0,202	0,24	0,133	1,095	1,098	0,181	0,14	5,817	1,682	0,404	0,401
0,01	0,95	2,54	1,029	1,7	0,727	0,255	0,257	0,353	0,13	6,133	5,473	0,285	0,188	1 043,062	100,154	19,358	6,644
	0,90	2,614	1,127	1,615	0,735	0,449	0,251	0,153	0,135	6,193	5,614	0,255	0,198	975,027	98,139	19,108	5,177
	0,85	2,797	1,209	1,457	0,701	0,274	0,266	0,128	0,156	6,196	5,595	0,264	0,19	996,815	95,732	18,123	6,324

TABLEAU 5.10 – Vitesse d'extraction des règles avec et sans méta-règles

En ce qui concerne les temps d'extraction, la version utilisant les méta-règles est beaucoup plus rapide que la version ne les utilisant pas. Plus le nombre de règles extraites dans la base est important et plus l'impact des méta-règles se fait ressentir. Comme nous pouvons le voir, la version avec les méta-règles est dix fois plus rapide pour un support à 0,01 sur la base `Statlog (Heart)`. Ce comportement est approprié au vu du nombre de règles élaguées de l'étude via la méta-règle MR_9. Des expérimentations plus poussées montrent que le gain provient uniquement de la méta-règle MR_9 alors que la méta-règle MR_4 fait perdre du temps à cause de la vérification de la condition et au faible nombre de règles écartées de l'étude.

Dans la prochaine section, nous allons mesurer l'impact de la contrainte de minimalité sur les motifs négatifs.

5.5.4 Impact de la contrainte de minimalité sur les motifs négatifs

Afin de mesurer l'impact de la contrainte de minimalité sur les motifs négatifs, nous avons fait une autre version de notre algorithme qui n'utilise pas cette contrainte. Dans ces expérimentations nous avons réutilisé les seuils précédents $max_{sup} = 0,80$, $min_{süp} = min_{sup}$ et $min_{M_G} = 0,60$ et nous avons fait varier le support minimum à 0,1, 0,05 et 0,01 et la confiance minimum à 0,95, 0,90 et 0,85. Le tableau 5.11 restitue les résultats.

Nous constatons que l'impact de la minimalité sur les motifs négatifs est très importante sur le nombre de règles extraites. L'impact le plus important se fait ressentir sur la base `Iris` où pour un support à 0,10 et une confiance à 0,85, la version sans la contrainte extrait presque 50% de règles en plus.

Quant aux temps d'exécution, les deux versions sont similaires sur nos expérimentations. La version avec la contrainte de minimalité permet d'un côté de gagner du temps sur la partie concernant la génération des règles. En effet, lors de l'étude d'une règle négative, si la règle possède un motif négatif non minimal, alors on évite (*dans le meilleur des cas*) la vérification de la méta-règle MR_4, le calcul et la vérification des supports (*prémisse négative, conclusion négative et motif négatif*) et le calcul et la vérification de la confiance et de la mesure M_G. Cependant d'un autre côté nous perdons du temps à rechercher l'ensemble des motifs négatifs minimaux. L'impact sur les temps d'extraction est donc nul, cependant nous pensons que sur des bases de données plus conséquentes le gain se fera sentir.

			Abalone		CMC		Ecoli		Iris		Nursery		Solar Flare 2		Statlog (Heart)		TTTE	
	Support	Confiance	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec	Sans	Avec
Nombre	0,10	0,95	3 493	2 959	12	12	81	72	397	267	48	48	38	32	132	132	0	0
		0,90	5 180	4 050	29	29	137	121	527	373	48	48	42	32	420	416	0	0
		0,85	6 344	5 150	34	33	164	147	623	434	48	48	48	36	777	770	0	0
	0,05	0,95	6 115	5572	67	67	190	179	442	312	173	173	68	62	1 490	1 490	18	18
		0,90	9 101	7 962	162	162	314	296	610	456	174	174	78	68	3 309	3 303	28	28
		0,85	11 177	9 974	195	194	380	361	711	522	177	177	89	77	5 380	5 368	44	44
	0,01	0,95	8 837	8032	469	423	606	477	555	406	2 562	2 554	149	133	93 146	89 240	2 884	2 684
		0,90	13 278	11 711	1 065	979	1 012	801	802	603	2 615	2 602	206	175	113 255	104 529	4 570	4 274
		0,85	16 519	14 840	1 386	1 280	1 202	982	928	693	2 691	2 678	241	206	139 369	129 458	6 248	5 944
Temps	0,10	0,95	0,249	0,244	0,091	0,096	0,076	0,078	0,087	0,078	0,24	0,247	0,078	0,082	0,199	0,201	0,113	0,115
		0,90	0,263	0,254	0,091	0,092	0,078	0,076	0,072	0,076	0,236	0,249	0,079	0,083	0,193	0,187	0,11	0,116
		0,85	0,29	0,289	0,091	0,093	0,078	0,077	0,076	0,078	0,231	0,245	0,082	0,083	0,194	0,207	0,114	0,109
	0,05	0,95	0,418	0,415	0,148	0,154	0,12	0,124	0,079	0,085	0,671	0,682	0,098	0,097	0,952	0,967	0,236	0,246
		0,90	0,444	0,448	0,153	0,147	0,119	0,124	0,082	0,086	0,685	0,684	0,096	0,099	1	0,98	0,238	0,242
		0,85	0,512	0,518	0,153	0,162	0,117	0,128	0,082	0,092	0,683	0,692	0,095	0,098	1,061	1,048	0,24	0,234
	0,01	0,95	0,625	0,613	0,44	0,465	0,167	0,162	0,091	0,094	3,233	3,275	0,13	0,134	55,294	56,634	4,332	4,383
		0,90	0,69	0,687	0,465	0,451	0,164	0,161	0,094	0,094	3,273	3,239	0,129	0,133	56,947	57,627	4,372	4,373
		0,85	0,773	0,761	0,486	0,459	0,166	0,16	0,098	0,095	3,287	3,265	0,132	0,135	64,053	62,883	4,407	4,428

Tableau 5.11 – Règles extraites avec et sans la contrainte de minimalité sur les motifs négatifs

5.5.5 Synthèse

Dans cette partie, nous avons mesuré l'impact des différentes améliorations que nous proposons dans l'algorithme. Comme nous venons de le voir, nous avons mesuré l'impact sur le nombre de règles extraites ainsi que sur les temps d'extraction des quatre propositions faites, à savoir l'utilisation d'un support maximum, l'utilisation de la mesure M_G en comparaison au facteur de certitude, l'utilisation des méta-règles et l'utilisation de la contrainte de minimalité sur les motifs négatifs.

Certains propositions sont plus significatives sur les résultats que d'autres. En effet, le support maximum a un impact important en limitant drastiquement le nombre de règles et en diminuant également le temps d'analyse. La méta-règle MR_9 est également souvent applicable et permet de fortement accélérer les traitements. La contrainte de minimalité influe également énormément sur le nombre de règles extraites même si aucune accélération de l'algorithme ne se fait ressentir dans les expérimentations. Les deux dernières propositions sont plus controversées. La méta-règle MR_4 semble posséder une condition d'application trop restrictive et a tendance à diminuer légèrement le temps d'analyse. Quant à la mesure M_G, ses zones plus restrictives permettent bien de générer moins de règles que le facteur de certitude, cependant dans nos expérimentations assez peu de règles tombent dans la zone différenciant les deux mesures. Cette différence pourrait être plus grande sur certaines bases de données. Dans la prochaine section, nous allons effectuer une comparaison quantitative de notre algorithme par rapport aux autres algorithmes d'extraction de règles d'association positives et négatives.

5.6 Étude quantitative

Dans l'étude quantitative, nous allons comparer l'ensemble des algorithmes en analysant le nombre extrait de chaque type de règles ainsi que les temps d'extraction. Nous analysons ces algorithmes sur l'ensemble des bases de données. Les quatre algorithmes d'extraction de règles d'association positives et négatives ne définissent pas de la même manière ce qu'est une règle valide. Par conséquent, le nombre de règles extraites va différer d'un algorithme à un autre. Dans la prochaine sous-partie, nous allons mettre en place un point de référence afin de pouvoir interpréter plus facilement les résultats.

5.6.1 `Apriori`

Afin d'avoir un point de comparaison, nous commençons par lancer `Apriori` sur l'ensemble des bases de données pour un support de 0,10, 0,05 et 0,01 et pour une confiance de 0,95, 0,90 et 0,85. Le tableau 5.12 contient les résultats obtenus pour l'ensemble des bases de données.

		Apriori															
		Abalone		CMC		Ecoli		Iris		Nursery		Solar Flare 2		Statlog (Heart)		TTTE	
min_{sup}	min_{conf}	Total	Tps	Total	Tps	Total	Tps	Total	Tps	Total	Tps	Total	Tps	Total	Tps	Total	Tps
0,10	0,95	6 297	0,611	543	0,249	989	0,132	122	0,074	24	0,482	207 998	2,424	3 767	0,41	12	0,121
	0,90	9 166	0,71	983	0,243	1 380	0,151	179	0,086	24	0,473	322 513	2,913	9 274	0,572	12	0,146
	0,85	11 476	0,698	1 403	0,232	1 800	0,162	218	0,068	24	0,478	415 611	3,683	15 675	0,484	12	0,179
0,05	0,95	14 389	0,856	2 134	0,361	2 791	0,215	177	0,072	87	0,996	610 145	6,784	26 817	1,666	12	0,213
	0,90	18 581	0,915	3 606	0,41	3 864	0,228	267	0,084	88	0,991	969 901	8,811	55 383	1,872	22	0,199
	0,85	21 831	0,938	4 864	0,425	5 032	0,25	316	0,083	90	0,986	1 190 700	11,078	89 641	1,995	36	0,201
0,01	0,95	24 061	1,39	18 685	1,382	10 308	0,363	334	0,082	1 333	3,757	1 827 882	36,0	2 556 269	113,61	2 316	1,285
	0,90	31 259	1,444	29 032	1,476	12 256	0,354	424	0,088	1 354	3,803	2 694 281	45,089	2 661 940	117,309	3 654	1,314
	0,85	37 213	1,558	38 678	1,473	14 823	0,369	473	0,089	1 390	3,754	3 249 547	49,037	2 964 784	122,349	4 770	1,323

TABLEAU 5.12 – Étude comparative pour l'algorithme `Apriori`

Concernant le nombre de règles, il en extrait en très grande quantité : le million est atteint pour les dernières expérimentations de Solar Flare 2 et de Statlog. Cependant, cet algorithme est assez rapide puisque le temps d'extraction ne dépasse pas trois minutes.

Par la suite, nous allons lancer les algorithmes d'extraction de règles d'association positives et négatives sur l'ensemble des bases de données.

5.6.2 Expérimentations sur les autres algorithmes

Pour l'ensemble des expérimentations suivantes, nous allons faire varier le support et la confiance comme nous venons de le faire pour `Apriori`, c'est-à-dire 0,10, 0,05 et 0,01 pour le support et 0,95, 0,90 et 0,85 pour la confiance. Pour les autres paramètres, nous retenons 0,10 pour la mesure d'intérêt nécessaire à l'algorithme de [Wu et al., 2004] et 0,60 pour la mesure de corrélation présente dans [Antonie and Zaïane, 2004]. Quant à notre algorithme, nous prendrons 0,60 pour M_G, 0,80 pour le seuil maximum du support et $min_{sup} = min_{süp}$.

Concernant notre algorithme, il extrait un type de règles en plus : les règles $\ddot{X} \Rightarrow \ddot{Y}$. Ce nouveau type de règles sera par la suite désigné sous le nom *Nouvelle*. Notre algorithme sera donc le seul à posséder une colonne *Nouvelle*. Les autres colonnes des tableaux seront $+$ $+$ pour désigner les règles positives, - $+$ pour désigner les règles $\overline{X} \Rightarrow Y$, $+$ - pour les désigner les règles $X \Rightarrow \overline{Y}$ et - - pour désigner les règles $\overline{X} \Rightarrow \overline{Y}$. Nous retrouvons également les colonnes *TotalN* qui correspondent à la somme des règles négatives, *Total* qui correspondent à la somme de l'ensemble des règles et *Tps* qui correspond au temps d'extraction en secondes. Pour notre algorithme, le nouveau type de règle ne sera pas pris en compte dans la somme pour les colonnes *TotalN* et *Total* puisque nous sommes les seuls à les extraire.

Afin de ne pas ajouter de biais supplémentaire, nous avons corrigé les algorithmes [Antonie and Zaïane, 2004] et [Cornelis et al., 2006] afin de s'assurer que toutes les règles extraites vérifient bien la contrainte du support minimum. En effet, comme nous l'avons mentionné dans le chapitre 2, [Antonie and Zaïane, 2004] oublient de vérifier la contrainte du support minimum lors de l'extraction des règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. [Cornelis et al., 2006] vont générer à partir d'un motif mixte $\overline{X}Y$ les règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ alors que le support de cette dernière et que la contrainte de minimalité du motif négatif $\overline{Y}$ ne sont pas vérifiés. Nous avons corrigé en extrayant les règles $\overline{X} \Rightarrow Y$ et $Y \Rightarrow \overline{X}$ à partir du motif mixte $\overline{X}Y$. Dans la suite, chaque base de données va être analysée une à une par ordre alphabétique. L'impact de nos modifications sera discuté dans la synthèse de cette sous-section.

5.6.2.1 Résultats synthétiques

Dans cette sous-sous-section, nous résumons les résultats obtenus dans la prochaine sous-sous-section 5.6.2.2. En effet, les résultats étant assez bruts, nous avons décidé de calculer la moyenne et l'écart type des résultats obtenus par la suite. Les tableaux 5.13 et 5.14 restituent les résultats obtenus sur les 4 bases de données : `Abalone`, `Ecoli`, `Iris` et `Nursery`. Nous avons exclu les bases `CMC` et `TTTE` de l'analyse car les méthodes de [Wu et al., 2004] et [Antonie and Zaïane, 2004] n'extraient pas ou trop peu de résultats et

risque donc de biaiser nos résultats. Il en est de même pour les bases `Statlog (Heart)` et `Solar Flare 2` où seule notre méthode `RAPN` arrive à obtenir des résultats pour toutes les expérimentations en moins de huit heures. Ces deux tableaux synthétisent les résultats et facilitent notre interprétation. Chaque case du tableau contient deux lignes : la première ligne correspond à la moyenne tandis que la seconde correspond à l'écart type.

		Apriori		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	1 858	0,32	202	0	187	0	187	388	20,95	188	34	127	17	178	366	0,66
		2 991	0,26	313	0	240	0	240	551	29,60	298	68	246	35	304	599	0,46
	0,9	2 687	0,36	414	5	291	0	295	709	19,81	389	86	231	33	350	739	0,67
		4 361	0,29	690	9	418	0	427	1 114	28,10	660	173	454	66	608	1 265	0,47
	0,85	3 380	0,35	654	28	398	0	426	1 080	19,37	498	133	270	35	437	935	0,75
		5 456	0,29	1 132	55	589	0	643	1 773	27,53	832	266	532	69	777	1 607	0,52
0,05	0,95	4 361	0,53	202	0	279	0	279	480	51,17	554	34	173	21	228	783	1,56
		6 802	0,46	313	0	408	0	408	721	87,07	1 023	68	339	42	395	1 416	1,33
	0,9	5 700	0,55	414	0	460	0	460	873	52,58	756	91	290	21	402	1 158	1,40
		8 762	0,46	690	0	740	0	740	1 429	89,43	1 386	182	573	42	741	2 126	1,22
	0,85	6 817	0,56	654	0	598	0	598	1 252	52,48	868	149	337	21	507	1 375	1,43
		10 265	0,46	1 132	0	973	0	973	2 104	89,73	1 556	297	667	42	951	2 506	1,18
0,01	0,95	9 009	1,40	202	0	446	0	446	647	152,26	630	46	227	149	422	1 052	9,66
		10 991	1,67	313	0	741	0	741	1 053	269,75	998	64	385	241	509	1 458	12,81
	0,9	11 323	1,42	414	0	536	0	536	950	147,83	833	103	345	149	597	1 429	9,57
		14 335	1,69	690	0	915	0	915	1 604	257,45	1 359	175	621	241	815	2 151	12,83
	0,85	13 475	1,44	654	0	536	0	536	1 190	157,96	957	161	392	149	701	1 658	9,36
		17 131	1,67	1 132	0	915	0	915	2 047	281,47	1 525	290	714	241	1 012	2 524	12,17

TABLEAU 5.13 – Étude comparative contenant la moyenne (en haut) et l'écart type (en bas) des résultats obtenus sur 4 bases de données pour [Wu et al., 2004] et [Antonie and Zaïane, 2004]

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	1 858	35	1 189	15	1 238	3 096	43,27	400	0	0	10	10	410	0,35	427
		2 991	43	1 179	10	1 227	4 116	69,49	696	0	0	9	9	703	0,22	715
	0,9	2 687	44	1 460	23	1 526	4 213	55,29	582	0	0	15	15	597	0,29	551
		4 361	57	1 446	14	1 512	5 722	93,84	1 011	0	0	14	14	1 023	0,19	917
	0,85	3 380	52	1 630	32	1 714	5 093	45,69	743	0	0	16	16	759	0,36	686
		5 456	71	1 631	18	1 710	6 994	74,86	1 298	0	0	15	15	1 310	0,34	1 165
0,05	0,95	4 361	36	3 236	19	3 292	7 653	73,62	695	0	0	10	10	705	0,71	854
		6 802	44	3 461	15	3 517	10 128	128,36	1 206	0	0	9	9	1 213	0,55	1 463
	0,9	5 700	45	3 819	31	3 895	9 595	78,52	1 066	1	1	15	17	1 083	0,71	1 140
		8 762	59	3 967	22	4 043	12 517	138,26	1 858	2	2	14	17	1 873	0,54	1 956
	0,85	6 817	53	4 185	44	4 281	11 098	76,11	1 338	1	2	16	19	1 357	0,61	1 402
		10 265	72	4 351	29	4 446	14 421	132,77	2 345	2	3	15	19	2 362	0,48	2 450
0,01	0,95	9 009	39	12 468	54	12 560	21 569	148,46	1 252	3	22	10	36	1 288	1,83	1 580
		10 991	47	9 096	34	9 169	19 530	255,99	1 483	6	26	9	38	1 514	2,55	2 078
	0,9	11 323	48	14 186	70	14 304	25 627	153,77	1 765	5	67	15	86	1 851	1,82	2 078
		14 335	61	10 190	47	10 284	23 716	265,67	2 336	10	58	14	76	2 397	2,46	2 870
	0,85	13 475	56	15 289	85	15 430	28 904	169,83	2 186	7	103	16	126	2 312	2,02	2 487
		17 131	74	10 905	57	11 015	27 091	299,60	3 066	13	96	15	118	3 170	2,70	3 582

TABLEAU 5.14 – Étude comparative contenant la moyenne (en haut) et l'écart type (en bas) des résultats obtenus sur 4 bases de données pour [Cornelis et al., 2006] et `RAPN`

Globalement nous pouvons voir que [Wu et al., 2004] est l'algorithme qui extrait en moyenne le moins de règles, devant [Antonie and Zaïane, 2004] et notre méthode `RAPN`. `Apriori` génère en moyenne quatre à sept fois plus de règles que notre méthode (*hors nouveau type*) alors qu'elle n'extrait que les règles positives. La méthode de [Cornelis et al., 2006] est celle qui extrait le plus de résultats en obtenant presque deux fois plus de résultats qu'`Apriori`. En effet, [Cornelis et al., 2006] extrait autant de règles positives qu'`Apriori` mais il extrait également beaucoup plus de règles négatives que les autres méthodes. Concernant le nombre de règles négatives traditionnelles (*hors nouveau type*) extraites, notre algorithme `RAPN` est celui qui en extrait le moins en moyenne. Alors que nous extrayons en moyenne 126 règles pour un support minimum à 0,01 et une confiance minimum à 0,95 sur les quatre bases de données synthétisées, [Wu et al., 2004] en obtient 536, [Antonie and Zaïane, 2004] 701 et [Cornelis et al., 2006] 15 430. Notre méthode `RAPN` semble extraire plus de règles $\overline{X} \Rightarrow \overline{Y}$ quand le support est élevé : 0,10 et 0,05 ; alors qu'il extrait plus de règles $X \Rightarrow \overline{Y}$ pour un support minimum à 0,01. Ce comportement s'explique par le fait que nous n'extrayons pas ou très peu de règles $X \Rightarrow \overline{Y}$ pour un support minimum à 0,10 ou à 0,05. Les autres algorithmes n'ont pas le même comportement puisqu'ils extraient tous en majorité des règles négatives du type $X \Rightarrow \overline{Y}$. En ce qui concerne le nouveau type de règles extraites, un travail complémentaire semble nécessaire puisque la moitié des règles que nous extrayons sont de ce type et il parait peu probable que toutes ces règles soient pertinentes. Les méthodes de [Wu et al., 2004] et [Cornelis et al., 2006] génèrent parfois plus de règles négatives que de règles positives. C'est le cas notamment pour un support minimum à 0,01 et une confiance minimum à 0,90. Nous remarquons également deux particularités concernant [Wu et al., 2004]. La première observation est que [Wu et al., 2004] n'extrait jamais de règles du type $\overline{X} \Rightarrow \overline{Y}$ puisque les moyennes et écarts types sont toujours égaux à 0 pour toutes les combinaisons de seuils utilisés. Même si nous regardons sur notre exemple fil-rouge, c'était également le cas (*cf. tableau* 2.9). Un comportement similaire est visible pour les règles du type $\overline{X} \Rightarrow Y$ extraites par [Wu et al., 2004] où l'algorithme ne semble extraire ces règles que pour un support élevé et que sur certaines bases. La moyenne et l'écart type concernant ces règles est donc souvent à 0. La seconde observation porte sur un comportement inattendu puisque [Wu et al., 2004] extraient plus de règles $\overline{X} \Rightarrow Y$ pour un support minimum à 0,10 que pour les autres valeurs du support minimum : 0,05 et 0,01. En moyenne, 28 règles sont extraites pour un support minimum à 0,10 et une confiance minimum à 0,85 alors qu'on obtient 0 règle pour un support à 0,05 et 0,01 en utilisant le même seuil de confiance minimum. Or théoriquement le nombre de règles augmente avec la diminution du support. Ce comportement anormal provient du fait qu'un motif peut être fréquent dans une expérimentation et non fréquent dans l'autre. Par exemple, un motif XY possédant un support à 0,09 sera considéré comme un motif fréquent d'intérêt potentiel pour un support de 0,05 et 0,01 mais sera un motif non fréquent d'intérêt potentiel pour un support à 0,10. Par conséquent, lorsque le support minimum sera à 0,10 nous essayerons de générer des règles négatives à partir de ce motif. Alors que pour les autres valeurs 0,05 et 0,01 du support, nous essayerons de générer des règles positives à partir de ce motif. Ce qui explique le fait de pouvoir obtenir plus de règles négatives quand le seuil du support est plus élevé. Ce comportement inattendu risque de perturber l'utilisateur métier en rendant ses interprétations plus compliquées. Concernant les temps d'exécution, notre algorithme semble être le plus rapide en s'exécutant presque aussi rapidement qu'`Apriori` en quelques secondes en moyenne. [Antonie and Zaïane, 2004] s'exécutent également en secondes en prenant toutefois entre deux et cinq fois plus de

temps que notre méthode. Pour les deux autres algorithmes [Wu et al., 2004] et [Cornelis et al., 2006], leurs méthodes procèdent en minutes et sont donc beaucoup plus lentes.

Notre interprétation des résultats synthétiques s'arrête ici. Les lecteurs souhaitant éluder l'interprétation des résultats détaillés peuvent directement aller à la section 5.6.3 où une synthèse des résultats est présentée.

5.6.2.2 Résultats détaillés

Dans cette sous-sous-section, nous allons interpréter base par base les résultats obtenus par les cinq algorithmes sur les huit bases de données étudiées. Nous étudierons les bases de données par ordre alphabétique. Commençons donc par `Abalone`.

Pour `Abalone` (*cf. tableau* 5.15), notre algorithme est le plus rapide et procède dans le même ordre de grandeur qu'`Apriori` qui n'extrait que des règles positives. Les trois autres algorithmes sont un peu plus lents mais s'exécutent tout de même en moins d'une minute. Il est logique que [Cornelis et al., 2006] soit plus lent qu'`Apriori` puisqu'il reprend sa méthode pour l'extraction des règles positives et extrait ensuite les règles négatives. En mettant de côté le nouveau type de règles, notre algorithme *RAPN* extrait quatre à cinq fois moins de règles qu'`Apriori` mais un peu plus que [Wu et al., 2004] et [Antonie and Zaïane, 2004]. Quant à [Cornelis et al., 2006], il extrait autant de règles positives qu'`Apriori` mais il extrait également beaucoup plus de règles négatives que les autres méthodes en atteignant plusieurs dizaines de milliers de règles pour un support minimum à 0,01. Nous remarquons que [Wu et al., 2004] extrait parfois plus de règles négatives que de règles positives. C'est le cas pour un support de 0,05 et 0,01 et une confiance à 0,90 et 0,95. Le comportement inattendu pour [Wu et al., 2004] que nous avions remarqué lors de l'interprétation des résultats synthétiques est également présent sur cette base. [Wu et al., 2004] extraient notamment 110 règles du type $\overline{X} \Rightarrow Y$ pour un support de 0,10 et une confiance à 0,85 alors que nous obtenons 0 règle $\overline{X} \Rightarrow Y$ pour un support de 0,05 et une confiance à 0,85. Nous constatons le même problème pour les règles de type $X \Rightarrow \overline{Y}$ entre un support à 0,05 et une confiance à 0,85 où 2 050 règles sont extraites alors qu'en diminuant le support à 0,01 nous obtenons que 1 906 règles. Ce comportement inattendu sur les règles $X \Rightarrow \overline{Y}$ avait été précédemment caché par la moyenne mais l'analyse détaillé met en avant ce problème. Théoriquement le nombre de règles augmente avec la diminution du support mais comme nous pouvons le voir ici, la méthode de [Wu et al., 2004] est contre-intuitive et risque de perturber l'interprétation des résultats. Comme nous l'avons dit précédemment, ce comportement anormal provient du fait qu'un motif peut être fréquent dans une expérimentation et non fréquent dans l'autre. Sur cette base, le motif $Diameter\ =]0,45,\ \infty[$, $Viscera\ weight\ =]-\infty,\ 0,25]$, $Shell\ weight\ =]-\infty,\ 0,34]$ possède un support à 0,099 et sera donc un motif fréquent d'intérêt potentiel pour un support de 0,05 et 0,01 mais sera un motif non fréquent d'intérêt potentiel pour un support à 0,10. Par conséquent, lorsque le support minimum sera à 0,10 nous essayerons de générer des règles négatives à partir de ce motif. Alors que pour les autres valeurs 0,05 et 0,01 du support, nous essayerons de générer des règles positives à partir de ce motif. L'analyse moyenne avait mis en avant que [Wu et al., 2004] n'extrayaient pas de règle négative du type $\overline{X} \Rightarrow \overline{Y}$ et comme l'analyse synthétique prenait la base `Abalone` en compte, c'est logiquement le cas ici. Concernant les nouvelles règles $\ddot{X} \Rightarrow \ddot{Y}$ nous en extrayons autant que le nombre total des autres types de règles.

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	669	0	538	0	538	1 207	18,328	630	136	495	0	631	1 261	1,223
	0,90	1 446	18	908	0	926	2 372	17,064	1 376	345	912	0	1 257	2 633	1,198
	0,85	2 350	110	1 270	0	1 380	3 730	16,436	1 742	531	1 067	0	1 598	3 340	1,223
0,05	0,95	669	0	886	0	886	1 555	20,868	2 088	136	681	0	817	2 905	3,051
	0,90	1 446	0	1 564	0	1 564	3 010	21,641	2 834	363	1 149	0	1 512	4 346	2,905
	0,85	2 350	0	2 050	0	2 050	4 400	20,941	3 200	594	1 337	0	1 931	5 131	2,666
0,01	0,95	669	0	1 554	0	1 554	2 223	40,975	2 115	136	800	90	1 026	3 141	7,65
	0,90	1 446	0	1 906	0	1 906	3 352	46,432	2 861	363	1 273	90	1 726	4 587	7,559
	0,85	2 350	0	1 906	0	1 906	4 256	40,116	3 227	594	1 461	90	2 145	5 372	7,956

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	6 297	88	2 612	24	2 724	9 021	19,012	1 441	0	0	21	21	1 462	0,58	1 497
	0,90	9 166	121	3 160	34	3 315	12 481	18,445	2 096	0	0	31	31	2 127	0,531	1 923
	0,85	11 476	151	3 544	40	3 735	15 211	18,661	2 686	0	0	34	34	2 720	0,816	2 430
0,05	0,95	14 389	90	7 543	38	7 671	22 060	21,394	2 503	0	0	21	21	2 524	1,243	3 048
	0,90	18 581	123	8 521	53	8 697	27 278	21,218	3 851	4	4	31	39	3 890	1,178	4 072
	0,85	21 831	153	9 335	72	9 560	31 391	21,916	4 854	4	6	34	44	4 898	0,981	5 076
0,01	0,95	24 061	93	21 671	89	21 853	45 914	44,641	3 325	12	58	21	91	3 416	1,219	4 616
	0,90	31 259	126	24 449	114	24 689	55 948	45,736	5 185	20	138	31	189	5 374	1,488	6 337
	0,85	37 213	156	26 173	135	26 464	63 677	43,49	6 724	26	231	34	291	7 015	1,166	7 825

TABLEAU 5.15 – Étude comparative sur la base `Abalone`

Pour `CMC` (*cf. tableau* 5.16), *RAPN* est encore le plus rapide et procède dans le même ordre de grandeur qu'`Apriori`. Les méthodes de [Antonie and Zaïane, 2004] et [Wu et al., 2004] prennent plus de temps (*jusqu'à 18 minutes*) pour n'extraire presque aucune règle. En effet, [Wu et al., 2004] extraient uniquement 5 règles positives pour une confiance minimum à 0,85 et ceci quel que soit le seuil du support minimum. Quant à [Antonie and Zaïane, 2004], aucune règle n'est extraite quels que soient les seuils utilisés [Cornelis et al., 2006] est aussi lent que [Wu et al., 2004] mais extrait un très grand nombre de règles. En effet, cette méthode génère 150 213 règles pour un support de 0,01 et une confiance de 0,85. Parmi ces 150 213 règles, 74% des règles sont négatives soit 111 535. Notre algorithme extrait pour ces mêmes seuils un nombre assez faible de règles : 570 alors qu'`Apriori` en extrait 38 678. Ici aussi, le nombre de règles $\ddot{X} \Rightarrow \ddot{Y}$ est sensiblement identique au nombre total des autres types de règles.

Pour la base de données `Ecoli` (*cf. tableau* 5.17), notre algorithme est encore le plus rapide et procède toujours dans le même ordre de grandeur qu'`Apriori` alors que [Antonie and Zaïane, 2004] et [Wu et al., 2004] prennent un peu plus de temps. [Cornelis et al., 2006] est ici encore l'algorithme le plus lent, mais procède tout de même en moins de 20 secondes. [Wu et al., 2004], [Antonie and Zaïane, 2004] et *RAPN* extraient toujours beaucoup moins de règles qu'`Apriori`. Nous remarquons également que [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006] extraient plus de règles négatives que de règles positives. C'est aussi notre cas si nous n'omettons pas le nouveau type de règles puisque ici encore nous extrayons un peu plus de règles $\ddot{X} \Rightarrow \ddot{Y}$ que le nombre total des autres types de règles. Tout comme pour la base `Abalone`, nous remarquons le même comportement anormal pour [Wu et al., 2004]. En effet, le nombre de règles du type $X \Rightarrow \overline{Y}$ diminue entre un support de 0,05 et une confiance à 0,90 et 0,85 et un support à 0,01. Nous remarquons également un problème avec les règles du type $\overline{X} \Rightarrow \overline{Y}$ pour [Antonie and Zaïane, 2004] où le nombre de règles pour une confiance de 0,90 et 0,85 n'augmente pas entre un support à 0,10 et 0,05. Ce comportement anormal provient du fait que [Antonie and Zaïane, 2004] travaillent par « opposition ». En effet, quand la corrélation est supérieure au seuil minimum de corrélation, ils étudient la règle positive si le motif est fréquent **sinon** ils étudient la règle $\overline{X} \Rightarrow \overline{Y}$, ce qui va entraîner ce problème. Prenons par exemple, le motif $mcg =]0,30,\ 0,59]$, $alm1 =]0,68,\ \infty[$, $alm2 =]0,66,\ \infty[$, $class = im$ qui possède un support à 0,095. La corrélation entre les motifs $mcg =]0,30,\ 0,59]$, $alm2 =]0,66,\ \infty[$ et $alm1 =]0,68,\ \infty[$, $class = im$ est supérieure au seuil minimum de corrélation et par conséquent les auteurs vont étudier soit les règles positives soit les règles $\overline{X} \Rightarrow \overline{Y}$ en fonction du support du motif. Lorsque le support minimum sera fixé à 0,01 ou 0,05 les auteurs étudieront donc la règle positive mais pour un support minimum à 0,10 ils étudieront la règle entièrement négative. Ce travail par « opposition » explique le fait de pouvoir obtenir plus de règles négatives quand le seuil du support est plus élevé. Ce problème était passé inaperçu lors de l'analyse synthétique.

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	0	0	0	0	0	0	233,725	0	0	0	0	0	0	1,178
	0,90	0	0	0	0	0	0	241,113	0	0	0	0	0	0	1,097
	0,85	5	0	0	0	0	5	253,297	0	0	0	0	0	0	1,106
0,05	0,95	0	0	0	0	0	0	217,637	0	0	0	0	0	0	5,001
	0,90	0	0	0	0	0	0	225,254	0	0	0	0	0	0	6,127
	0,85	5	0	0	0	0	5	240,28	0	0	0	0	0	0	5,325
0,01	0,95	0	0	0	0	0	0	1023,446	0	0	0	0	0	0	113,328
	0,90	0	0	0	0	0	0	1078,03	0	0	0	0	0	0	117,358
	0,85	5	0	0	0	0	5	1066,607	0	0	0	0	0	0	143,777

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	543	3	831	3	837	1 380	224,38	8	0	0	0	0	8	0,195	4
	0,90	983	27	1 722	26	1 775	2 758	213,351	14	0	0	1	1	15	0,13	14
	0,85	1 403	46	2 634	89	2 769	4 172	210,54	14	0	0	1	1	15	0,19	18
0,05	0,95	2 134	3	5 350	7	5 360	7 494	249,223	35	0	0	0	0	35	0,153	32
	0,90	3 606	29	8 884	71	8 984	12 590	248,156	65	0	0	1	1	66	0,209	96
	0,85	4 864	49	11 872	144	12 065	16 929	233,248	78	0	0	1	1	79	0,293	115
0,01	0,95	18 685	3	68 290	86	68 379	87 064	1 119,016	221	0	2	0	2	223	1,159	200
	0,90	29 032	32	91 938	160	92 130	121 162	1 087,793	394	0	21	1	22	416	1,158	563
	0,85	38 678	52	111 248	235	111 535	150 213	1 129,982	529	0	40	1	41	570	0,83	710

Tableau 5.16 – Étude comparative sur la base `CMC`

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	45	0	138	0	138	183	1,765	18	0	12	69	81	99	0,517
	0,90	81	0	182	0	182	263	1,699	45	0	12	132	144	189	0,525
	0,85	126	0	236	0	236	362	1,739	96	0	12	138	150	246	0,484
0,05	0,95	45	0	138	0	138	183	2,661	27	0	12	84	96	123	0,823
	0,90	81	0	182	0	182	263	2,609	54	0	12	84	96	150	0,765
	0,85	126	0	236	0	236	362	2,487	120	0	12	84	96	216	0,797
0,01	0,95	45	0	138	0	138	183	11,855	304	48	108	504	660	964	2,573
	0,90	81	0	146	0	146	227	11,875	334	48	108	504	660	994	2,345
	0,85	126	0	146	0	146	272	12,16	448	48	108	504	660	1108	2,417

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	989	52	1 692	11	1 755	2 744	7,099	18	0	0	3	3	21	0,18	51
	0,90	1 380	54	2 153	25	2 232	3 612	6,99	41	0	0	5	5	46	0,15	75
	0,85	1 800	56	2 418	44	2 518	4 318	6,6	58	0	0	5	5	63	0,122	84
0,05	0,95	2 791	55	4 532	16	4 603	7 394	7,261	58	0	0	3	3	61	0,342	118
	0,90	3 864	57	5 691	39	5 787	9 651	7,236	112	0	0	5	5	117	0,237	179
	0,85	5 032	59	6 247	58	6 364	11 396	7,566	159	0	0	5	5	164	0,192	197
0,01	0,95	10 308	62	17 076	76	17 214	27 522	17,537	192	0	24	3	27	219	0,401	258
	0,90	12 256	66	19 058	108	19 232	31 488	17,855	282	0	88	5	93	375	0,288	426
	0,85	14 823	68	20 446	132	20 646	35 469	17,212	354	0	120	5	125	479	0,416	503

TABLEAU 5.17 – Étude comparative sur la base `Ecoli`

Pour `Iris` (*cf. tableau* 5.18), les temps d'extraction sont sensiblement similaires pour l'ensemble des algorithmes. L'algorithme de [Antonie and Zaïane, 2004] n'extrait aucune règle négative. Les 3 autres algorithmes n'extraient pas non plus certains types de règles négatives : les règles $\overline{X} \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ pour [Wu et al., 2004], les règles $\overline{X} \Rightarrow Y$ pour [Cornelis et al., 2006] et les règles $\overline{X} \Rightarrow Y$ pour *RAPN*. Cependant [Cornelis et al., 2006] extraient ici aussi plus de règles négatives que de règles positives. Ici encore, les règles composées de conjonctions de motifs négatifs sont présentes en quantité similaire au nombre total de règles extraites des autres types. Encore une fois, [Wu et al., 2004] extraient plus de règles $\overline{X} \Rightarrow Y$ pour un support à 0,05 et une confiance à 0,85 que pour un support à 0,01.

Pour `Nursery` (*cf. tableau* 5.19), notre algorithme est encore le plus rapide mais s'exécute un peu plus lentement qu'`Apriori`. Viennent ensuite [Antonie and Zaïane, 2004], [Wu et al., 2004] puis [Cornelis et al., 2006] qui est encore le plus lent. Pour l'extraction de règles négatives, on retrouve exactement le même comportement que pour la base de données `Iris`. L'algorithme de [Antonie and Zaïane, 2004] n'extrait aucune règle négative. Les 3 autres algorithmes n'extraient pas non plus certains types de règles négatives : les règles $\overline{X} \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ pour [Wu et al., 2004], les règles $\overline{X} \Rightarrow Y$ pour [Cornelis et al., 2006] et les règles $\overline{X} \Rightarrow Y$ pour *RAPN*. [Wu et al., 2004] et [Cornelis et al., 2006] extraient plus de règles négatives que de règles positives pour l'ensemble des expérimentations. Concernant les nouvelles règles $\ddot{X} \Rightarrow \ddot{Y}$ nous en extrayons autant que le nombre total des autres types de règles.

Pour `Solar Flare 2` (*cf. tableau* 5.20), [Wu et al., 2004] et [Cornelis et al., 2006] mettent plus de huit heures pour s'exécuter : les résultats ne sont donc pas renseignés. Il en est de même pour [Antonie and Zaïane, 2004] pour un support à 0,01. Notre algorithme s'exécute plus rapidement qu'`Apriori` tandis que [Antonie and Zaïane, 2004] prennent environ 55 minutes pour un support à 0,01 et une confiance à 0,85. `Apriori` passe le million de règles générées, en extrayant 3 249 547 règles pour un support de 0,01 et une confiance à 0,85. Un comportement anormal est détecté chez [Antonie and Zaïane, 2004] qui extraientt plus de règles négatives avec un support à 0,10 qu'à 0,05. Ce comportement provient du travail par opposition dont nous avons parlé précédemment mais cette fois-ci l'impact est beaucoup plus important. Ici encore, la moitié des règles que nous extrayons sont du type $\ddot{X} \Rightarrow \ddot{Y}$. Par ailleurs, nous extrayons assez peu de règles. En effet, seulement 92 règles sont extraites pour un support à 0,01 et une confiance à 0,85.

En ce qui concerne `Statlog (Heart)` (*cf. tableau* 5.21), on retrouve le même comportement pour les temps d'extraction que pour la base de données `Solar Flare 2`. En effet, [Wu et al., 2004], [Cornelis et al., 2006] mettent ici aussi plus de huit heures pour s'exécuter. C'est également le cas pour [Antonie and Zaïane, 2004] avec un support à 0,01 tandis que pour les autres seuils du support, ils n'extraient aucune règle. Ici encore notre extraction est plus rapide que celle d'`Apriori` qui génère ici aussi entre deux et trois millions de règles pour un support à 0,01, alors que nous extrayons un nombre plus raisonnable : entre 56 000 et 72 000.

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	87	0	62	0	62	149	0,119	96	0	0	0	0	96	0,126
	0,90	121	0	64	0	64	185	0,11	129	0	0	0	0	129	0,099
	0,85	134	0	78	0	78	212	0,124	146	0	0	0	0	146	0,16
0,05	0,95	87	0	82	0	82	169	0,124	96	0	0	0	0	96	0,115
	0,90	121	0	84	0	84	205	0,118	129	0	0	0	0	129	0,121
	0,85	134	0	98	0	98	232	0,124	146	0	0	0	0	146	0,126
0,01	0,95	87	0	82	0	82	169	0,147	96	0	0	0	0	96	0,121
	0,90	121	0	85	0	85	206	0,139	129	0	0	0	0	129	0,136
	0,85	134	0	85	0	85	219	0,146	146	0	0	0	0	146	0,136

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	122	0	350	21	371	493	0,114	115	0	0	15	15	130	0,134	137
	0,90	179	0	400	30	430	609	0,114	167	0	0	21	21	188	0,132	185
	0,85	218	0	424	38	462	680	0,111	203	0	0	24	24	227	0,078	207
0,05	0,95	177	0	469	21	490	667	0,138	132	0	0	15	15	147	0,137	165
	0,90	267	0	540	30	570	837	0,133	212	0	0	21	21	233	0,242	223
	0,85	316	0	574	38	612	928	0,135	248	0	0	24	24	272	0,208	250
0,01	0,95	334	0	720	21	741	1 075	0,19	171	0	6	15	21	192	0,101	214
	0,90	424	0	791	30	821	1 245	0,196	251	0	28	21	49	300	0,112	303
	0,85	473	0	825	38	863	1 336	0,196	287	0	42	24	66	353	0,451	340

TABLEAU 5.18 – Étude comparative sur la base `Iris`

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	6	0	8	0	8	14	63,607	6	0	0	0	0	6	0,781
	0,90	6	0	8	0	8	14	60,362	6	0	0	0	0	6	0,844
	0,85	6	0	8	0	8	14	59,164	6	0	0	0	0	6	1,142
0,05	0,95	6	0	8	0	8	14	181,039	6	0	0	0	0	6	2,244
	0,90	6	0	8	0	8	14	185,949	6	0	0	0	0	6	1,819
	0,85	6	0	8	0	8	14	186,349	6	0	0	0	0	6	2,15
0,01	0,95	6	0	8	0	8	14	556,063	6	0	0	0	0	6	28,294
	0,90	6	0	8	0	8	14	532,886	6	0	0	0	0	6	28,235
	0,85	6	0	8	0	8	14	579,423	6	0	0	0	0	6	26,948

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	24	0	100	2	102	126	146,842	24	0	0	2	2	26	0,49	22
	0,90	24	0	125	2	127	151	195,594	24	0	0	2	2	26	0,344	22
	0,85	24	0	134	6	140	164	157,377	24	0	0	2	2	26	0,441	22
0,05	0,95	87	0	400	2	402	489	265,705	87	0	0	2	2	89	1,104	84
	0,90	88	0	524	2	526	614	285,49	88	0	0	2	2	90	1,178	84
	0,85	90	0	582	6	588	678	274,804	90	0	0	2	2	92	1,069	85
0,01	0,95	1 333	0	10 404	29	10 433	11 766	531,462	1 321	0	1	2	3	1 324	5,582	1 230
	0,90	1 354	0	12 445	29	12 474	13 828	551,28	1 342	0	12	2	14	1 356	5,387	1 246
	0,85	1 390	0	13 712	33	13 745	15 135	618,433	1 378	0	19	2	21	1 399	6,036	1 279

TABLEAU 5.19 – Étude comparative sur la base `Nursery`

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	temps supérieur à 8h							40 662	0	216	136 323	136 539	177 201	165,613
	0,90								65 880	0	216	201 204	201 420	267 300	167,404
	0,85								85 320	0	216	201 204	201 420	286 740	195,976
0,05	0,95								96 795	0	216	0	216	97 011	2 269,952
	0,90								162 108	0	216	0	216	162 324	3 194,441
	0,85								209 250	0	216	0	216	209 466	3 342,43
0,01	0,95								temps supérieur à 8h						
	0,90														
	0,85														

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	temps supérieur à 8h							16	0	0	2	2	18	0,165	14
	0,90								16	0	0	2	2	18	0,105	14
	0,85								17	0	0	2	2	19	0,227	17
0,05	0,95								31	0	0	2	2	33	0,326	29
	0,90								31	0	0	2	2	33	0,156	35
	0,85								34	0	0	2	2	36	0,119	41
0,01	0,95								63	0	0	2	2	65	0,179	68
	0,90								67	0	11	2	13	80	0,242	95
	0,85								74	0	16	2	18	92	0,356	114

TABLEAU 5.20 – Étude comparative sur la base `Solar Flare 2`

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	temps supérieur à 8h							0	0	0	0	0	0	57,316
	0,90								0	0	0	0	0	0	58,009
	0,85								0	0	0	0	0	0	54,821
0,05	0,95								0	0	0	0	0	0	1 997,51
	0,90								0	0	0	0	0	0	1 696,41
	0,85								0	0	0	0	0	0	1 549,454
0,01	0,95								temps supérieur à 8h						
	0,90														
	0,85														

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	temps supérieur à 8h							53	0	0	0	0	53	0,615	79
	0,90								196	0	0	0	0	196	0,424	220
	0,85								369	0	0	1	1	370	0,414	400
0,05	0,95								669	0	0	0	0	669	1,926	821
	0,90								1 527	0	0	0	0	1 527	1,527	1 776
	0,85								2 516	0	0	1	1	2 517	1,768	2 851
0,01	0,95								55 930	0	2	0	2	55 932	114,664	33 308
	0,90								60 599	0	336	0	336	60 935	109,198	43 594
	0,85								70 649	1	1 135	1	1 137	71 786	109,718	57 672

TABLEAU 5.21 – Étude comparative sur la base `Statlog (Heart)`

Et enfin pour la dernière base `TTTE` (*cf. tableau* 5.22), notre algorithme s'exécute un peu plus lentement qu'`Apriori` mais est toujours plus rapide que les autres. Viennent ensuite [Antonie and Zaïane, 2004], puis [Wu et al., 2004] et [Cornelis et al., 2006]. [Wu et al., 2004] et [Antonie and Zaïane, 2004] n'extraient aucune règle pour l'ensemble des expérimentations. *RAPN* n'en génère pas non plus pour un support à 0,10. [Cornelis et al., 2006] génèrent encore une fois plus de règles négatives que de règles positives. Alors que nous n'extrayons uniquement des règles négatives du type $X \Rightarrow \overline{Y}$ que pour un support à 0,01 et une confiance à 0,85.

		[Wu et al., 2004]							[Antonie and Zaïane, 2004]						
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps
0,10	0,95	0	0	0	0	0	0	4,388	0	0	0	0	0	0	0,502
	0,90	0	0	0	0	0	0	4,592	0	0	0	0	0	0	0,621
	0,85	0	0	0	0	0	0	3,914	0	0	0	0	0	0	0,682
0,05	0,95	0	0	0	0	0	0	25,722	0	0	0	0	0	0	2,069
	0,90	0	0	0	0	0	0	21,425	0	0	0	0	0	0	1,66
	0,85	0	0	0	0	0	0	22,225	0	0	0	0	0	0	2,088
0,01	0,95	0	0	0	0	0	0	1315,638	0	0	0	0	0	0	86,307
	0,90	0	0	0	0	0	0	1356,987	0	0	0	0	0	0	79,085
	0,85	0	0	0	0	0	0	1322,955	0	0	0	0	0	0	86,686

		[Cornelis et al., 2006]							*RAPN*							
min_{sup}	min_{conf}	+ +	- +	+ -	- -	TotalN	Total	Tps	+ +	- +	+ -	- -	TotalN	Total	Tps	Nouvelle
0,10	0,95	0	0	6	0	6	6	16,918	0	0	0	0	0	0	0,241	0
	0,90	0	0	8	0	8	8	18,157	0	0	0	0	0	0	0,157	0
	0,85	0	0	87	1	88	88	17,196	0	0	0	0	0	0	0,121	0
0,05	0,95	12	0	60	0	60	72	83,414	12	0	0	0	0	12	0,365	6
	0,90	22	0	178	0	178	200	87,807	22	0	0	0	0	22	0,281	6
	0,85	36	0	787	1	788	824	87,564	36	0	0	0	0	36	0,6	8
0,01	0,95	2 316	0	9 982	0	9 982	12 298	948,547	2 050	0	0	0	0	2 050	8,067	634
	0,90	3 654	0	23 198	0	23 198	26 852	944,54	3 188	0	0	0	0	3 188	8,394	1 086
	0,85	4 770	0	37 757	1	37 758	42 528	937,924	4 086	0	66	0	66	4 152	9,403	1 792

TABLEAU 5.22 – Étude comparative sur la base TTTE

5.6.3 Synthèse

Dans cette partie, nous avons comparé, sur l'ensemble des bases de données, `Apriori` et les quatre algorithmes d'extraction de règles d'association positives et négatives : [Wu et al., 2004], [Antonie and Zaïane, 2004], [Cornelis et al., 2006] et notre algorithme `RAPN`. Pour chaque expérimentation, nous avons détaillé le nombre de règles extraites de chaque type ainsi que les temps d'exécution.

Concernant le nombre de règles extraites, nous constatons qu'`Apriori` génère sur les plus importantes bases de données plusieurs millions de règles. C'est également le cas pour [Cornelis et al., 2006] qui reprennent l'algorithme d'`Apriori` pour les règles positives et qu'ils extraient en plus les règles négatives. Par ailleurs, [Cornelis et al., 2006] extraient souvent plus de règles négatives que de règles positives (*jusqu'à dix fois plus pour* `Nursery` *et* `TTTE`), ce qui augmente encore plus le nombre de règles extraites. On retrouve également ce comportement sur [Wu et al., 2004] et [Antonie and Zaïane, 2004] pour plusieurs bases. Si nous n'omettons pas le nouveau type de règles négatives $\ddot{X} \Rightarrow \ddot{Y}$, notre algorithme possède un comportement similaire. C'est d'autant plus le cas, que la majorité des règles extraites par notre algorithme sur l'ensemble des bases sont de ce type. Un travail supplémentaire sera à apporter afin de diminuer ce nombre car il est peu probable que toutes ces règles soient intéressantes. Il arrive également que certains algorithmes n'extraient aucune règle négative sur certaines bases, ce qui est le cas pour [Antonie and Zaïane, 2004] sur les bases `Iris` et `Nursery` ou encore le cas de [Wu et al., 2004] sur la base `CMC` ; voir également aucune règle, ce qui est le cas pour [Antonie and Zaïane, 2004] sur les bases `CMC`, `Statlog (Heart)` et `TTTE` ou encore le cas de [Wu et al., 2004] sur la base `TTTE`. Concernant [Wu et al., 2004], nous avons remarqué que quelle que soit la base utilisée lors de nos expérimentations et quels que soit les seuils pris en considération, la méthode ne parvenait pas à générer de règles négatives du type $\overline{X} \Rightarrow \overline{Y}$, et des règles du type $\overline{X} \Rightarrow Y$ n'ont pu être extraites que sur la base `Abalone` et sur l'exemple fil-rouge de la thèse. La dernière remarque que nous pouvons faire sur le nombre de règles extraites concerne le comportement anormal des méthodes de [Wu et al., 2004] et [Antonie and Zaïane, 2004] sur les règles négatives. En effet, contrairement à ce qu'on pourrait penser, le fait de diminuer le support ne va pas forcément faire augmenter le nombre de règles négatives. Ce comportement oblige l'utilisateur à faire attention lors de l'interprétation des résultats.

En ce qui concerne les temps d'extraction, notre algorithme `RAPN` est globalement l'algorithme le plus rapide. Il est même aussi voir parfois plus rapide qu'`Apriori` qui n'extrait que des règles positives. Les autres algorithmes sont beaucoup plus lents. En effet, sur les bases les plus importantes, à savoir `Solar Flare 2` et `Statlog`, [Wu et al., 2004] et [Cornelis et al., 2006] prennent plus de huit heures pour s'exécuter quel que soit le seuil du support. [Antonie and Zaïane, 2004] procèdent également en plus de huit heures sur ces deux bases pour un support à 0,01. En comparaison, `Apriori` et `RAPN` s'exécutent en moins de deux minutes.

Dans ces expérimentations, nous avons implémenté les versions corrigées des algorithmes de [Antonie and Zaïane, 2004] et de [Cornelis et al., 2006]. Pour rappel, nos modifications sont visibles dans les tableaux 2.10 et 2.20 représentant respectivement les critères de validité d'une règle selon [Antonie and Zaïane, 2004] et selon [Cornelis et al., 2006].

Concernant [Antonie and Zaïane, 2004], nos modifications n'ont aucun impact sur les résultats dans nos expérimentations. Cela provient notamment du fait que les différents supports minimaux choisis sont assez faibles. Afin de comparer concrètement l'impact de nos modifications, nous avons relancé sur la base de données `Abalone` avec d'autres paramètres : 0,25 et 0,40 pour le support, 0,85 pour la confiance, et 0,50 pour la corrélation.

En résultat, pour la version corrigée avec un support à 0,25, nous obtenons 4 258 règles au total : 2 291 règles positives $X \Rightarrow Y$, 240 règles négatives du type $\overline{X} \Rightarrow Y$, 1 026 règles négatives du type $X \Rightarrow \overline{Y}$ et 701 règles négatives du type $\overline{X} \Rightarrow \overline{Y}$. Alors que pour la version non corrigée, nous obtenons 4 306 règles au total : 2 291 règles positives $X \Rightarrow Y$, 288 règles négatives du type $\overline{X} \Rightarrow Y$, 1 026 règles négatives du type $X \Rightarrow \overline{Y}$ et 701 règles négatives du type $\overline{X} \Rightarrow \overline{Y}$. Par conséquent, seules les règles du type $\overline{X} \Rightarrow Y$ ont été impactées et l'impact semble assez réduit.

En résultat, pour la version corrigée avec un support à 0,40, nous obtenons 1 938 règles au total : 1 305 règles positives $X \Rightarrow Y$, 45 règles négatives du type $\overline{X} \Rightarrow Y$, 102 règles négatives du type $X \Rightarrow \overline{Y}$ et 486 règles négatives du type $\overline{X} \Rightarrow \overline{Y}$. Alors que pour la version non corrigée, nous obtenons 2 085 règles au total : 1 305 règles positives $X \Rightarrow Y$, 90 règles négatives du type $\overline{X} \Rightarrow Y$, 204 règles négatives du type $X \Rightarrow \overline{Y}$ et 486 règles négatives du type $\overline{X} \Rightarrow \overline{Y}$. Ici le nombre de règles du type $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ a doublé, l'impact est donc plus important.

Finalement les résultats de [Antonie and Zaïane, 2004] ne sont pas impactés dans nos expérimentations, cependant il faudra faire attention en utilisant cette méthode puisque plus le seuil du support minimum sera élevé et plus l'impact sera important.

Concernant [Cornelis et al., 2006], toutes les expérimentations sont fortement impactées. Par exemple, sur la base de données `Abalone` avec un support minimum à 0,10 et une confiance minimum à 0,95, la version modifiée extrait 2 612 règles $X \Rightarrow \overline{Y}$ tandis que la version originale en extrait 3 150. Parmi toutes ces règles, seules 31 sont communes aux deux expérimentations. Ou encore sur la base `Abalone` avec un support minimum à 0,01 et une confiance minimum à 0,85, la version modifiée extrait 26 173 règles $X \Rightarrow \overline{Y}$ tandis que la version originale en extrait 35 756. Parmi toutes ces règles, seules 134 sont communes aux deux expérimentations.

5.7 Conclusion

Dans ce chapitre, nous avons réalisé une étude quantitative sur notre algorithme. Les expérimentations ont eu lieu sur huit différentes bases de données récupérées sur le site de l'UCI Machine Learning Repository [UCI, 2015]. Cette étude se divise en trois parties distinctes. Dans la première partie, nous avons mesuré l'impact des différents paramètres de notre algorithme en faisant varier leurs valeurs, à savoir : le support minimum min_{sup}, le support maximum max_{sup}, le support minimum du motif négatif $\ddot{X}$ $min_{s\ddot{u}p}$, la confiance minimum min_{conf} et la valeur de M_G minimum min_{M_G}. Nous constatons que les paramètres ayant le plus d'impact sur le nombre de règles et sur les temps d'extraction sont le support minimum et le support maximum. Ces résultats ne sont pas totalement surprenants puisque l'impact du support minimum était connu depuis le célèbre algorithme `Apriori`. Le support minimum pour le motif négatif $\ddot{X}$ et la confiance minimum sont ensuite les paramètres qui impactent le plus. Le paramètre semblant le moins impacter les résultats est la mesure M_G mais cela peut provenir du

choix des valeurs ou des bases de données étudiées.

Dans la deuxième partie, nous avons mesuré l'impact des différentes améliorations que nous proposons dans notre méthode, à savoir : le nombre de règles élaguées par le support maximum, la différence de règles élaguées par M_G par rapport au facteur de certitude utilisé par [Wu et al., 2004], l'impact de l'utilisation des méta-règles et l'application de la contrainte de minimalité sur les motifs négatifs. Nous constatons que les améliorations les plus notables sont l'utilisation d'un support maximum et l'utilisation de la méta-règle MR_9. Cette méta-règle correspond à la propriété de la confiance appliquée aux nouveaux types de règles. La contrainte de minimalité influe également énormément sur le nombre de règles extraites même si aucune accélération de l'algorithme ne se fait ressentir dans les expérimentations. La mesure M_G possède bien des zones plus restrictives que le facteur de certitude, cependant la différence n'est pas flagrante dans nos expérimentations mais pourrait l'être davantage sur des bases de données différentes. Concernant la méta-règle MR_4, des travaux supplémentaires sont nécessaires afin d'en tirer pleinement parti. En effet, il faudrait arriver à rendre la condition pour appliquer la méta-règle MR_4 moins restrictive puisqu'à l'heure actuelle, notre algorithme ne l'utilise presque pas.

Dans la troisième et dernière partie, nous avons comparé notre algorithme à `Apriori` et aux trois autres algorithmes d'extraction de règles d'association positives et négatives : [Wu et al., 2004], [Antonie and Zaïane, 2004], [Cornelis et al., 2006]. Notre méthode est aussi rapide qu'`Apriori` qui n'extrait que des règles positives. Les autres algorithmes sont beaucoup plus lents puisque sur les plus grosses bases de données ils mettent plus de huit heures pour retourner les résultats alors que nous procédons en environ deux minutes. En ce qui concerne les règles extraites, notre algorithme `RAPN` semble le plus équilibré même si un travail supplémentaire est nécessaire concernant l'extraction du nouveau type de règles puisque cinquante pour cent des règles que nous générons sont du nouveau type. Malgré ce point faible, notre algorithme comble les différents problèmes des autres méthodes. En effet, nous ne générons jamais plusieurs millions de règles en résultats comme le font `Apriori` et [Cornelis et al., 2006]. Si nous omettons les règles $\ddot{X} \Rightarrow \ddot{Y}$, nous n'extrayons pas, contrairement aux autres algorithmes, plus de règles négatives que de règles positives. Il arrive également que [Antonie and Zaïane, 2004] et [Wu et al., 2004] n'extraient aucune règle négative voir aucune règle du tout sur une base de données. Et enfin, notre méthode ne possède pas de comportement anormal comme nous avons détecté dans [Wu et al., 2004] et [Antonie and Zaïane, 2004] qui en diminuant le support minimum risque de faire diminuer le nombre de règles négatives.

Dans le chapitre 2, après l'étude des différents algorithmes d'extraction de règles d'association positives et négatives nous avions mis en avant deux failles : un nombre encore important de règles inintéressantes et un parcours non optimisé de recherche des règles. Nous avons par la suite, proposé des améliorations permettant de combler théoriquement ces deux failles. Dans ce chapitre, nous avons pu vérifier que nos propositions avaient un effet bénéfique sur ces deux problèmes comme en atteste la deuxième partie des expérimentations. Cependant l'analyse synthétique présente dans la troisième partie des expérimentations a également révélé que nous extrayons plus de règles en moyenne (*hors nouveau type)* que les méthodes de [Wu et al., 2004] et [Antonie and Zaïane, 2004]. Différents facteurs peuvent expliquer ces résultats. Pour [Wu et al., 2004], cela provient

notamment du seuil utilisé pour le facteur de certitude. En effet, une règle est générée si la valeur du facteur de certitude est supérieure ou égale à min_{conf}. Cette contrainte est très forte en comparaison du seuil que nous utilisons pour la mesure M_G qui est de 0,60. Si nous fixons M_G minimum à la valeur de la confiance minimum alors nous générons un petit peu moins de règles (*hors nouveau type*) que leur méthode. Par exemple, sur la base `Abalone`, en utilisant un support à 0,01 et une confiance à 0,85, nous ne générons plus que 4 110 règles traditionnelles alors qu'ils en génèrent 4 256. Pour [Antonie and Zaïane, 2004], comme nous l'avons mentionné au début de l'étude quantitative, les auteurs oublient de vérifier la contrainte du support minimum lors de l'extraction des règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. En rétablissant cette contrainte, cela va donc restreindre le nombre de règles extraites. Par ailleurs, nous ne sommes pas certains que notre méthode `RAPN` génère globalement plus de résultats que leur méthode. En effet, si nous observons attentivement les résultats obtenus sur la base de données `Solar Flare 2`, [Antonie and Zaïane, 2004] génèrent en moyenne 243 747 règles pour un support minimum à 0,10 et 156 267 pour un support minimum à 0,05 alors que nous en générons en moyenne (*hors nouveau type*) respectivement 18 et 34. En conclusion, le nombre de règles extraites par chaque algorithme va fortement dépendre de la base de données étudiée, et aucune déduction ne peut être faite sur l'algorithme extrayant le moins de règles.

Dans les expérimentations précédentes, nous avons considéré la qualité d'une méthode aux faibles nombres de règles extraites et à son temps d'exécution. Bien que logique pour les temps d'extraction, cela ne l'est pas forcément pour le nombre de règles générées. Cela provient du fait que pour un scientifique des données, un nombre trop important de résultats rend impossible leurs interprétations. Afin de faciliter et d'accélérer ce travail d'interprétation, il faut donc obtenir les résultats les plus pertinents le plus rapidement possible. Pour se faire, il faut donc élaguer les règles inintéressantes dès la phase de recherche au lieu de les éliminer dans une phase de post-traitement. Si les différents critères pour élaguer une règle inintéressante sont pertinents, et si on s'assure de ne pas supprimer de règles intéressantes au passage, alors plus le nombre de règles sera faible et plus on s'assure d'obtenir les meilleurs résultats. Seulement rien ne prouve que l'on ne perd pas de règles intéressantes, et encore moins que les critères d'élagage sont les plus pertinents. Dans le prochain chapitre, nous allons effectuer une analyse qualitative des résultats afin de savoir quel algorithme extrait les règles les plus intéressantes.

Chapitre 6

Comparaisons approfondies

Sommaire

6.1 Introduction

Dans ce dernier chapitre nous allons comparer qualitativement notre algorithme avec les autres algorithmes que nous avons étudiés, à savoir `Apriori`, [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006]. Nous commençons par présenter le logiciel Association Rules Analyzer (`ARA`) que nous avons développé avec l'aide de Guillaume SOUSA AMARAL dans le but de pouvoir comparer les règles extraites par les différentes méthodes. Nous utilisons ensuite `ARA` sur les règles extraites sur l'exemple fil-rouge par les différentes méthodes afin de mettre en avant les règles communes ainsi que les règles spécifiques à chacune. Nous effectuons ensuite la même analyse mais cette fois-ci sur l'une des bases de données présentées dans le chapitre précédent afin de vérifier si on obtient les mêmes résultats sur une base plus importante. La dernière partie de ce chapitre concerne la qualité des règles extraites. Nous commençons donc par définir un ensemble de propriétés nécessaires afin de pouvoir considérer une règle comme intéressante. Puis nous appliquons certaines mesures de qualité aux règles extraites afin de vérifier si celles-ci sont intéressantes.

6.2 ARA : Association Rules Analyzer

`ARA` est un logiciel open source et multiplateforme permettant d'effectuer une comparaison quantitative et qualitative des algorithmes d'extraction de règles d'association positives et négatives. Développé en C++ avec la librairie Qt, `ARA` va utiliser les fichiers de résultats provenant du logiciel `Weka` puis va permettre à l'utilisateur de les analyser de manière intuitive et interactive. Divers traitements sont disponibles : l'affichage de règles communes à plusieurs algorithmes, l'affichage de règles spécifiques à un algorithme, le calcul de mesures, le filtrage par type de règles, par motif ou encore par valeurs de mesures.

Le logiciel est composé de deux onglets : un onglet de sélection des fichiers à analyser (*cf. figure* 6.1) et un onglet permettant d'analyser ces mêmes fichiers (*cf. figure* 6.2).

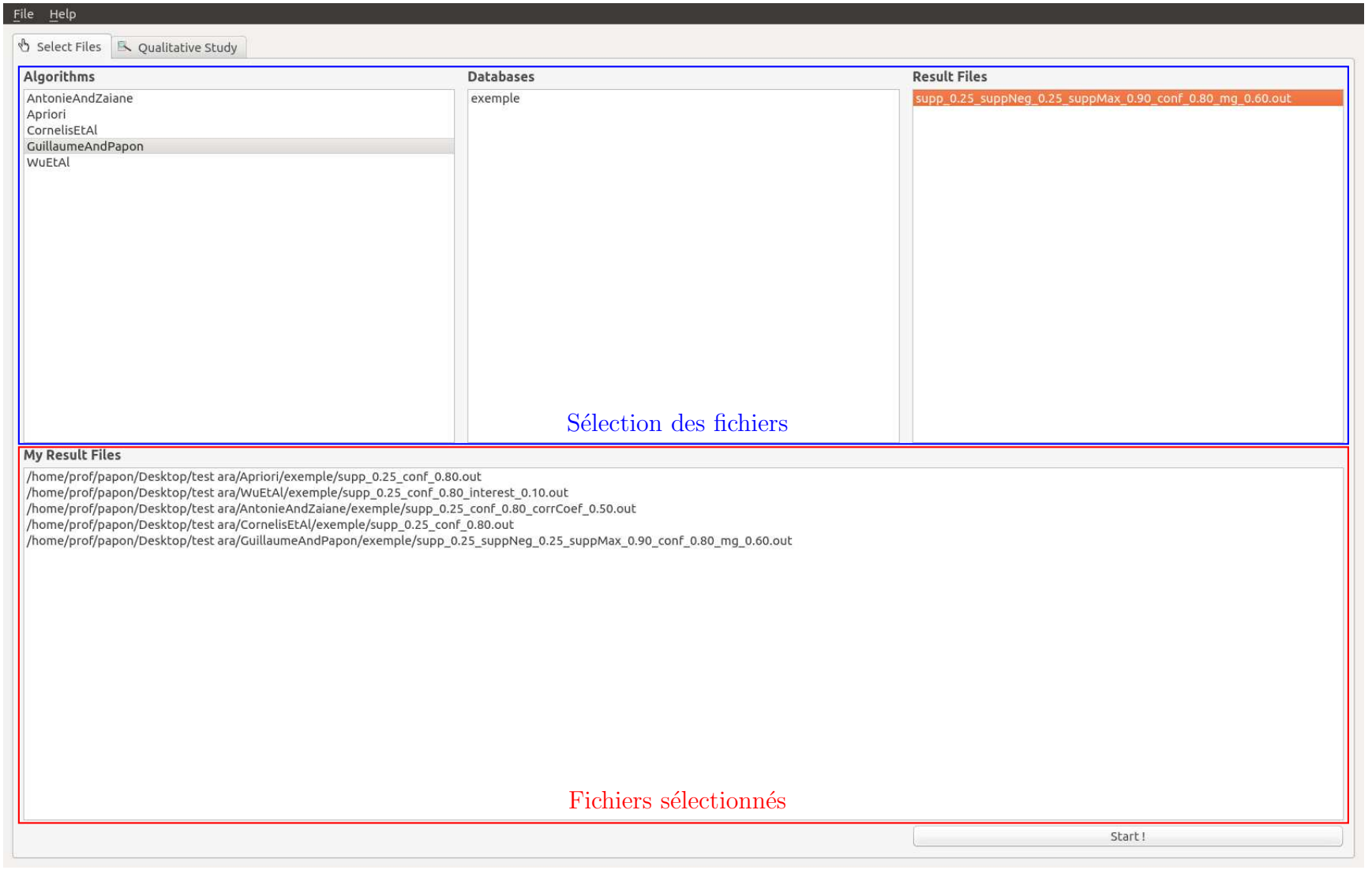

FIGURE 6.1 – Onglet de sélection des fichiers dans ARA

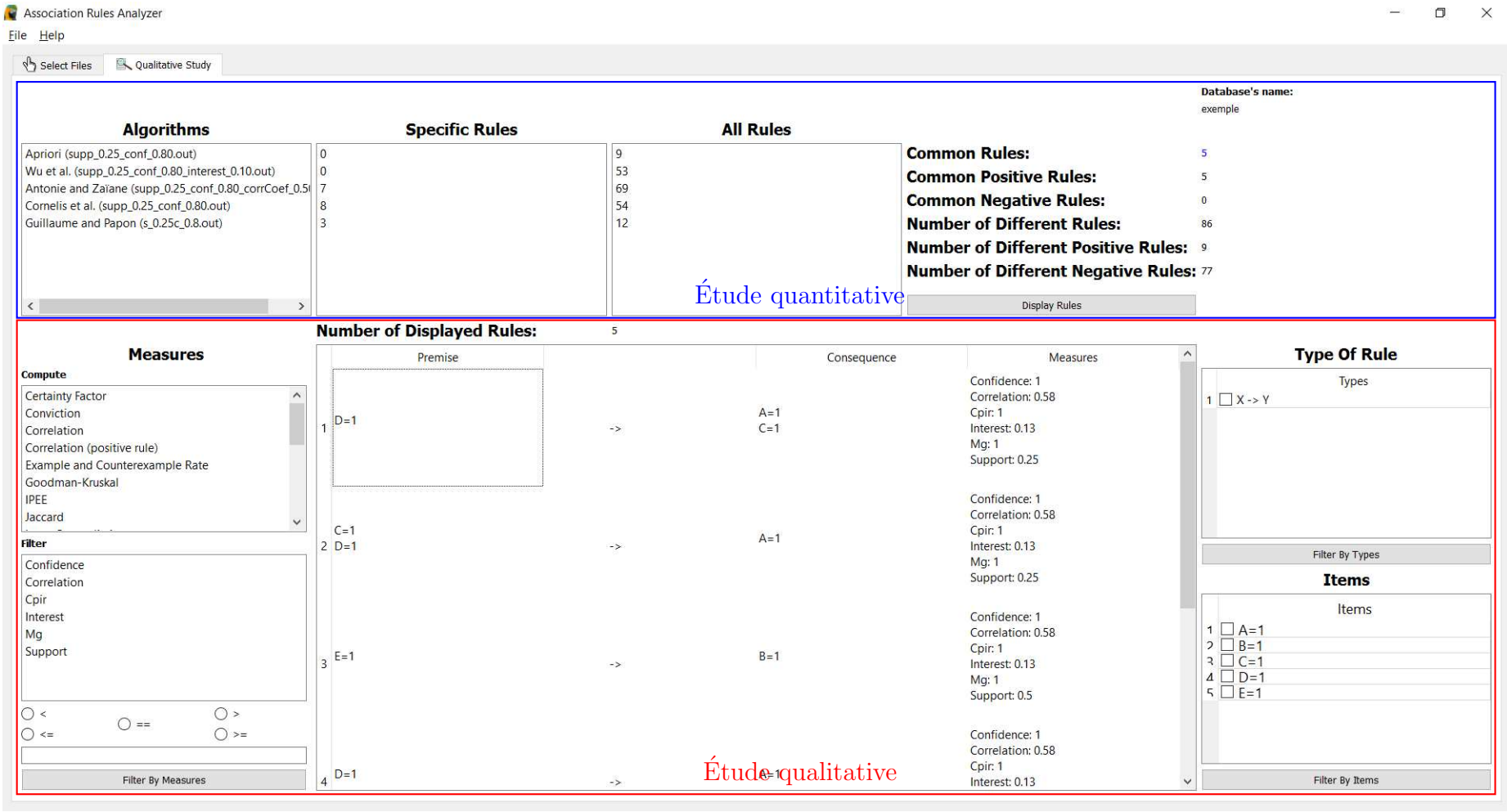

FIGURE 6.2 – Onglet de travail dans ARA

Afin de faciliter l'explication du premier onglet, nous l'avons divisé en deux zones distinctes. La première zone, correspondant au rectangle bleu, est intitulée *Sélection des fichiers* et se situe dans la partie haute de la fenêtre. La seconde zone, correspondant au rectangle rouge, est intitulée *Fichiers sélectionnés* et se situe dans la partie basse de la fenêtre.

Le premier onglet est composé de quatre zones. Les trois zones de la partie supérieure permettent à l'utilisateur de parcourir le dossier contenant les résultats, tandis que la zone inférieure affiche les fichiers sélectionnés pour l'analyse. Dans cet onglet, l'utilisateur doit commencer par charger le fichier contenant les résultats en allant dans le menu `File`. Ce dossier doit suivre une arborescence précise (*cf. figure* 6.3) afin que le logiciel puisse fonctionner correctement. Cette arborescence contient les résultats triés par algorithmes puis par bases de données.

```
.
├── AntonieAndZaiane
│   ├── exemple
│   │   └── supp_0.25_conf_0.80_corrCoef_0.50.out
│   └── iris
│       ├── supp_0.01_conf_0.85_corrCoef_0.60.out
│       ├── supp_0.01_conf_0.90_corrCoef_0.60.out
│       └── supp_0.01_conf_0.95_corrCoef_0.60.out
├── Apriori
│   ├── exemple
│   │   └── supp_0.25_conf_0.80.out
│   └── iris
│       ├── supp_0.01_conf_0.85.out
│       ├── supp_0.01_conf_0.90.out
│       └── supp_0.01_conf_0.95.out
├── CornelisEtAl
│   ├── exemple
│   │   └── supp_0.25_conf_0.80.out
│   └── iris
│       ├── supp_0.01_conf_0.85.out
│       ├── supp_0.01_conf_0.90.out
│       └── supp_0.01_conf_0.95.out
├── GuillaumeAndPapon
│   ├── exemple
│   │   └── supp_0.25_suppNeg_0.25_suppMax_0.90_conf_0.80_mg_0.60.out
│   └── iris
│       ├── supp_0.01_suppNeg_0.01_suppMax_0.80_conf_0.90_mg_0.50.out
│       ├── supp_0.01_suppNeg_0.01_suppMax_0.80_conf_0.90_mg_0.60.out
│       └── supp_0.01_suppNeg_0.01_suppMax_0.80_conf_0.90_mg_0.70.out
└── WuEtAl
    ├── exemple
    │   └── supp_0.25_conf_0.80_interest_0.10.out
    └── iris
        ├── supp_0.01_conf_0.85_interest_0.10.out
        ├── supp_0.01_conf_0.90_interest_0.10.out
        └── supp_0.01_conf_0.95_interest_0.10.out
```

FIGURE 6.3 – Exemple d'arborescence de fichiers de résultats valides

Une fois le dossier de résultats chargé, la partie `Algorithms` contenue dans la partie *Sélection des fichiers* du premier onglet se remplit automatiquement en récupérant le nom des différents algorithmes. En sélectionnant l'algorithme (*par un clic*), la partie `Databases` liste automatiquement le nom des différentes bases de données. Puis en sélectionnant la base de données, la partie `Result Files` énumère automatiquement les différents fichiers de résultats. Quand l'utilisateur sélectionne un fichier à analyser, il apparaîtra ensuite dans la partie *Fichiers sélectionnés*. Par ailleurs, lorsque un fichier sera sélectionné, les autres bases de données deviendront inaccessibles car l'analyse n'est pertinente que si les fichiers analysés proviennent de la même base de données. En cas d'erreur, l'utilisateur peut supprimer un fichier de l'analyse en cliquant sur son nom.

Pour lancer l'analyse, l'utilisateur doit cliquer sur `Start !`. `ARA` va ensuite extraire les informations nécessaires contenues dans les différents fichiers de résultats sélectionnés puis ouvrira le second onglet (*cf. figure* 6.2). Il va récupérer les noms des différents algorithmes, les différentes mesures utilisées puis les règles avec leurs valeurs pour les différentes mesures. Le parseur est générique et permet de récupérer ces informations quel que soit le nombre de règles et quelles que soient les mesures utilisées par l'algorithme. La seule contrainte est de respecter la même structure de fichiers de résultats (*non présentée dans ce manuscrit*) qu'`Apriori` dans `Weka`.

Pour faciliter l'explication du second onglet, nous avons également divisé la fenêtre en deux zones distinctes. La première zone, correspondant au rectangle bleu, est intitulée *Étude quantitative* et se situe dans la partie haute de la fenêtre. La seconde zone, correspondant au rectangle rouge, est intitulée *Étude qualitative* et se situe dans la partie basse de la fenêtre.

Une fois l'analyse lancée, le second onglet se remplit (*cf. figure* 6.2). L'interface de cet onglet est beaucoup plus dense, car elle restitue à l'utilisateur l'ensemble des informations provenant des différents fichiers analysés. La zone *Étude quantitative*, qui se situe dans la partie supérieure, permet d'obtenir des informations quantitatives sur les règles. Elle est composée de différentes éléments :

- `Database's name` : cette zone affiche le nom de la base de données analysée.
- `Algorithms` : cette zone permet d'afficher les noms des différents algorithmes sélectionnés dans l'onglet précédent ainsi que les noms des fichiers qui contiennent les résultats. On affiche un algorithme par ligne.
- `Specific Rules` : cette zone permet d'afficher le nombre de règles spécifiques à chaque algorithme, c'est-à-dire le nombre de règles qui n'existent dans aucun autre fichier analysé.
- `All Rules` : cette zone permet d'afficher le nombre de règles produites par chaque algorithme.
- `Common Rules` : cette zone permet d'afficher le nombre de règles communes à l'ensemble des fichiers analysés.
- `Common Positive/Negative Rules` : ces zones permettent d'afficher le nombre de règles positives/négatives communes à l'ensemble des fichiers analysés.
- `Number of Different Rules` : cette zone permet d'afficher le nombre total de règles produites par l'ensemble des algorithmes.
- `Number of Different Positive/Negative Rules` : ces zones permettent d'afficher le nombre total de règles positives/négatives produites par l'ensemble des algorithmes.
- `Bouton Display Rules` : ce bouton permet d'afficher les règles dans la partie inférieure de la fenêtre. L'utilisateur peut choisir parmi les règles spécifiques, les règles communes ou l'ensemble des règles pour un algorithme et les afficher. De plus, si l'utilisateur sélectionne plusieurs noms d'algorithme puis clique sur ce bouton, cela affichera dans la partie inférieure les règles communes à ces algorithmes.

La zone *Étude qualitative* située dans la partie inférieure de l'affichage, permet à l'utilisateur d'analyser la qualité des règles. Elle se décompose également en plusieurs zones :

- `Number of Displayed Rules` : ce champ affiche le nombre de règles présentes dans le tableau central présent juste en dessous. Ce tableau central donne toutes les

informations concernant les règles : la prémisse, la conclusion, ainsi que l'ensemble des mesures calculées pour cette règle.

— `Measures` : cette zone, située à gauche, permet le calcul de mesures puis le filtrage par mesures. Une liste de mesures disponibles dans le logiciel est présentée à l'utilisateur qui peut demander le calcul de ces mesures pour les règles présentes dans le tableau central. Une fois calculées, les valeurs s'ajoutent aux valeurs de mesures présentes dans la colonne `Measures` du tableau central mais également dans la liste des filtres. Pour filtrer, l'utilisateur sélectionne une mesure, un test et écrit une valeur puis clique sur le bouton `Filter By Measures`. Cela déclenche la mise à jour du tableau central, en supprimant les règles qui ne respectent pas le critère défini.

— `Type of Rule` : cette zone, située à droite, recense les types de règles présents dans le tableau central. Dans l'exemple présent dans la figure 6.2, les règles affichées dans le tableau central sont des règles positives de la forme $X \Rightarrow Y$. Dans le cas où il existe plusieurs types de règles dans le tableau central, l'utilisateur peut filtrer en gardant uniquement le(s) type(s) qu'il veut étudier en cliquant sur `Filter By Types` après avoir coché les cases correspondantes aux types choisis.

— `Items` : cette zone, située à droite, recense les items présents dans les règles affichées. L'utilisateur peut cocher certains items puis choisir de filtrer les règles ne les possédant pas en cliquant sur `Filter By Items`.

Dans la prochaine section, nous allons comparer les règles extraites par notre algorithme à celles des différents algorithmes présentés dans le chapitre 2.

6.3 Comparaison sur l'exemple fil-rouge

Nous allons maintenant comparer les règles extraites par les différents algorithmes. Nous rappelons tout d'abord dans le tableau 6.1 le nombre de règles de chaque type extraites par chaque algorithme. La colonne $TotalN$ correspond aux nombres de règles négatives extraites tandis que la colonne $Total$ correspond à l'ensemble des règles extraites par la méthode. Le nouveau type de règles pour notre méthode `RAPN` ne sera pas pris en compte dans les colonnes $TotalN$ et $Total$ car nous sommes le seul algorithme à les extraire. Les $\times$ spécifient que l'algorithme concerné n'extrait pas ce type de règles.

	$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$	$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$	$\ddot{X} \Rightarrow \ddot{Y}$	TotalN	Total
`Apriori`	9	$\times$	$\times$	$\times$	$\times$	$\times$	9
Wu	5	16	32	0	$\times$	48	53
Antonie	5	24	40	0	$\times$	64	69
Cornelis	9	17	24	4	$\times$	45	54
`RAPN`	7	0	0	2	3	2	9

TABLEAU 6.1 – Récapitulatif des règles extraites par les différents algorithmes sur l'exemple fil-rouge

Le premier constat que l'on peut faire c'est que le nombre de règles dépend fortement de l'algorithme d'extraction employé. L'hypothèse est que les algorithmes extraient les

mêmes règles mais comme les contraintes de validité des règles sont plus ou moins fortes les algorithmes extraient plus ou moins de règles. Vérifions cette hypothèse à l'aide d'`ARA`. Nous allons diviser notre étude en deux parties : une pour les règles positives et une pour les règles négatives. Le tableau 6.2 présente le nombre de règles positives communes / spécifiques à chaque algorithme.

	`Apriori`	Wu	Antonie	Cornelis	`RAPN`	Communes / Spécifiques
`Apriori`	×	5 / 4	5 / 4	9 / 0	7 / 2	9 / 0
Wu	5 / 0	×	5 / 0	5 / 0	5 / 0	5 / 0
Antonie	5 / 0	5 / 0	×	5 / 0	5 / 0	5 / 0
Cornelis	9 / 0	5 / 4	5 / 4	×	7 / 2	9 / 0
`RAPN`	7 / 0	5 / 2	5 / 2	7 / 0	×	7 / 0

TABLEAU 6.2 – Nombre de règles positives communes / spécifiques pour chaque algorithme

On peut voir dans la colonne *Communes / Spécifiques* et la ligne `Apriori` que 9 règles apparaissent dans au moins un des autres algorithmes et ne possède donc aucune règle spécifique ou propre à lui même. [Wu et al., 2004] et [Antonie and Zaïane, 2004] sont les algorithmes les plus restrictifs puisqu'ils génèrent seulement 5 règles positives alors que `RAPN` et [Cornelis et al., 2006] en génèrent 7 et 9. Il est normal que [Cornelis et al., 2006] génèrent les mêmes règles positives qu'`Apriori` puisqu'ils réutilisent son algorithme pour la partie positive. Les 5 règles positives de [Wu et al., 2004] et [Antonie and Zaïane, 2004] sont communes à l'ensemble des algorithmes : $A \Rightarrow C$, $CD \Rightarrow A$, $D \Rightarrow A$, $D \Rightarrow AC$, et $E \Rightarrow B$. Notre algorithme génère également $AD \Rightarrow C$ et $D \Rightarrow C$. Et enfin, seul `Apriori` et [Cornelis et al., 2006] extraient $AB \Rightarrow C$ et $CE \Rightarrow B$.

[Wu et al., 2004] ne génèrent pas $AB \Rightarrow C$, $AD \Rightarrow C$, $CE \Rightarrow B$ et $D \Rightarrow C$ puisque la valeur de l'intérêt pour ces règles n'est pas suffisante. En effet, la valeur est de 0,0625 pour l'ensemble de ces règles et par conséquent la fonction *fip* n'est pas égale à 1. Les motifs correspondants ne sont donc pas d'intérêt potentiel et ne vont donc pas générer de règles. [Antonie and Zaïane, 2004] ne génèrent pas $AB \Rightarrow C$, $AD \Rightarrow C$, $CE \Rightarrow B$ et $D \Rightarrow C$ puisque la valeur du coefficient de corrélation pour ces règles n'est pas suffisante. En effet, la valeur est de 0,33 pour l'ensemble de ces règles et par conséquent elles ne sont pas extraites. Notre algorithme `RAPN` n'extrait pas $AB \Rightarrow C$ et $CE \Rightarrow B$ car les motifs ABC et BCE ne sont pas des motifs raisonnablement fréquents. En effet, même si leur support vérifie la contrainte du support minimum et du support maximum, le support du motif $\ddot{XY}$ ne vérifie pas $min_{süp}$ puisqu'il est nul pour les deux motifs.

Concernant les règles positives, l'hypothèse est donc vérifiée car l'ensemble des règles extraites par les algorithmes d'extraction de règles positives et négatives existent dans `Apriori`. L'étude détaillée pour les règles négatives est plus délicate puisque le nombre de règles générées est plus important. Regardons le tableau 6.3 qui présente le nombre de règles négatives communes / spécifiques à chaque algorithme.

[Antonie and Zaïane, 2004] génèrent 7 règles négatives spécifiques, [Cornelis et al., 2006] en génèrent 8 et nous en générons 3 qui sont du nouveau type : $\ddot{CD} \Rightarrow \ddot{A}$. Par conséquent aucun des algorithmes n'extrait l'ensemble des règles négatives. De plus sur

	Wu	Antonie	Cornelis	RAPN	Communes / Spécifiques
Wu	×	48 / 0	26 / 22	0 / 48	48 / 0
Antonie	48 / 16	×	35 / 29	0 / 64	57 / 7
Cornelis	26 / 19	35 / 10	×	2 / 43	37 / 8
RAPN	0 / 5	0 / 5	2 / 3	×	2 / 3

TABLEAU 6.3 – Nombre de règles négatives communes / spécifiques pour chaque algorithme

les 77 règles négatives différentes (*non affichées dans le tableau*), aucune n'est commune à l'ensemble des algorithmes. Cela provient notamment de notre algorithme qui génère seulement 5 règles dont 3 qui sont du type $\ddot{X} \Rightarrow \ddot{Y}$. Les 2 règles restantes sont uniquement communes à [Cornelis et al., 2006]. 26 règles sont communes à [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006]. Ne voulant pas détailler l'ensemble des 77 règles, notre interprétation est limitée. Nous remarquons seulement que l'ensemble des règles de [Wu et al., 2004] est contenu dans celle d' [Antonie and Zaïane, 2004]. Concernant les règles négatives, l'**hypothèse** n'est donc pas vérifiée.

Ce petit exemple met en avant les différences de résultats entre les algorithmes. Certains algorithmes sont plus proches que d'autres par rapport aux règles extraites. En effet, un groupe de règles négatives plus ou moins important est commun à [Wu et al., 2004], [Antonie and Zaïane, 2004] et [Cornelis et al., 2006] alors que pour les règles positives les algorithmes les plus proches sont `Apriori`, [Cornelis et al., 2006] et `RAPN`. Cependant cet exemple étant assez petit, les résultats peuvent être dûs au hasard. Dans la prochaine section, nous allons faire la même étude sur l'une des bases de données de l'UCI Machine Learning Repository.

6.4 Comparaison sur la base de données `Abalone`

Afin d'obtenir une comparaison plus rigoureuse, nous allons effectuer notre étude sur l'une des bases précédemment utilisées. Notre choix se porte sur la base `Abalone` avec un support minimum à 0,01 et une confiance minimum à 0,85 car cette extraction génère un nombre suffisant de résultats pour chaque méthode. Rappelons tout d'abord dans la tableau 6.4 le nombre de règles extraites par chaque algorithme.

	$X \Rightarrow Y$	$\overline{X} \Rightarrow Y$	$X \Rightarrow \overline{Y}$	$\overline{X} \Rightarrow \overline{Y}$	$\ddot{X} \Rightarrow \ddot{Y}$	TotalN	Total
Apriori	37 213	×	×	×	×	×	37 213
Wu	2 350	0	1 906	0	×	1 906	4 256
Antonie	3 227	594	1 461	90	×	2 145	5 372
Cornelis	37 213	156	26 173	135	×	26 464	63 677
RAPN	6 724	26	231	34	7 825	291	7 015

TABLEAU 6.4 – Récapitulatif des règles extraites par les différents algorithmes sur *Abalone*

Même constat que précédemment, le nombre de règles dépend fortement de l'al-

gorithme d'extraction. En effet, `Apriori` génère environ entre cinq et neuf fois plus de règles que [Wu et al., 2004], [Antonie and Zaïane, 2004] et que notre algorithme si on omet le nouveau type de règles négatives. Seul l'algorithme de [Cornelis et al., 2006] génère environ deux fois plus de règles qu'`Apriori`. Regardons dans le tableau 6.5 le nombre de règles positives et négatives communes / spécifiques à chaque algorithme.

	`Apriori`	Wu	Antonie	Cornelis	`RAPN`	Communes / Spécifiques
`Apriori`	×	2 350 / 34 863	3 227 / 33 986	37 213 / 0	6 724 / 30 489	37 213 / 0
Wu	2 350 / 1 906	×	1 517 / 2 739	2 678 / 1 578	784 / 3 472	3 020 / 1 236
Antonie	3 227 / 2 145	1 517 / 3 855	×	3 660 / 1 712	657 / 4 715	4 002 / 1 370
Cornelis	37 213 / 26 464	2 678 / 60 999	3 660 / 60 017	×	7 015 / 56 662	38 108 / 25 569
`RAPN`	6 724 / 8 116	784 / 14 056	657 / 14 183	7 015 / 7 825	×	7 015 / 7 825

TABLEAU 6.5 – Nombre de règles communes / spécifiques pour chaque algorithme sur la base `Abalone`

`ARA` nous informe que 366 règles sont communes à l'ensemble des algorithmes et que toutes ces règles communes sont positives. Il nous indique également que 74 450 règles différentes sont extraites par ces algorithmes : 37 213 règles positives et 37 237 règles négatives. On retrouve dans cette expérimentation le même comportement pour les règles positives que précédemment : tous les algorithmes d'extraction de règles d'association positives et négatives génèrent un sous-ensemble des règles générées par `Apriori`. Parmi les règles négatives, aucune n'est commune à l'ensemble des algorithmes d'extraction de règles positives et négatives. Sur l'exemple fil-rouge, nous avions l'excuse que l'exemple était trop petit, mais ce n'est plus le cas ici. L'explication d'une telle différence dans les résultats provient en partie du nouveau type de règles extraites par notre algorithme, des seuils utilisés pour les différentes mesures mais également de la méthode de construction des motifs appliqués pour générer les règles négatives. En effet, les méthodes de recherche des motifs varient énormément d'une méthode à l'autre. [Wu et al., 2004] construisent les motifs en combinant deux k-motifs fréquents pour créer un *(k+1)*-motif candidat. Si le seuil du support de ce nouveau motif candidat n'est pas vérifié alors il sera ajouté à l'ensemble des motifs non fréquents d'intérêt potentiel et sera utilisé par la suite pour générer les règles négatives. Alors que [Antonie and Zaïane, 2004] décident de combiner un k-motif fréquent avec un 1-motif fréquent pour construire leurs motifs M de taille supérieure. Si la corrélation d'une combinaison $X \cup Y = M$ est inférieure à la négation du seuil de corrélation ou qu'elle est supérieure au seuil de corrélation avec un support non fréquent alors ces combinaisons vont être utilisées pour générer des règles négatives. [Cornelis et al., 2006] utilisent également des méthodes de construction singulières pour leurs motifs négatifs et mixtes. Effectivement, la construction des motifs négatifs passent par une augmentation d'un item fréquent à l'une des parties négatives alors que pour les motifs mixtes, ils augmentent d'abord la partie positive du motif puis la partie négative. Dans notre proposition *RAPN*, nous réutilisons la même méthode de construction qu'*Apriori* en ajoutant simplement des contraintes supplémentaires. Ces différentes méthodes de construction procurent un ensemble différent de motifs à étudier, ce qui va amener à des règles négatives différentes. Dans la prochaine partie, nous allons comparer la qualité des règles extraites par les différentes méthodes.

6.5 Comparaison qualitative des règles extraites

Dans cette section, nous allons focaliser notre comparaison uniquement sur la qualité des règles extraites et non sur la vitesse d'extraction ou encore sur la quantité de règles générées comme nous l'avons fait précédemment. Cependant il est assez difficile de comparer ces algorithmes puisque leurs définitions pour une règle valide est différente, et par conséquent ils n'obtiennent pas les mêmes ensembles de règles en sortie. À l'aide d'`ARA`, nous analyserons donc en post-traitement, l'utilité des règles extraites, en regardant si celles-ci vérifient certaines propriétés, que nous jugeons intéressantes, en s'appuyant sur différentes mesures de qualité. Nous définissons donc dans un premier temps, quelques propriétés qu'une règle doit respecter pour être intéressante, puisqu'il ne suffit pas d'être valide pour l'être.

6.5.1 Règles intéressantes

Dans cette partie, nous exposons quelques propriétés qu'une règle doit respecter pour être considérée comme intéressante puisque le respect du support et de la confiance minimum ne suffit pas pour juger de la qualité d'une règle. Nous proposons également une solution afin de vérifier ces propriétés. Cependant, afin de ne pas biaiser les expérimentations concernant la vérification des propriétés, nous choisissons, quand c'est possible, des mesures qui ne sont pas déjà utilisées dans un des algorithmes de l'étude. De plus, certaines mesures permettent de vérifier plusieurs propriétés, mais nous voulons quantifier indépendamment chaque problème.

➤ Omniprésence

La contrainte du support minimum permet de sélectionner uniquement les motifs fréquents. Cependant il existe un problème car certains motifs fréquents vont conduire à des règles non pertinentes. Ces problèmes interviennent quand la base de données étudiée contient au moins un motif omniprésent. Comme nous avons pu le voir dans la section 3.2.1, les règles composées d'un motif omniprésent sont inintéressantes.

➤ Indépendance

De manière générale, l'indépendance désigne l'absence d'influence entre deux variables ou événements. En fouille de données, deux motifs sont dit indépendants lorsque l'apparition d'un des motifs n'affecte pas l'apparition de l'autre, c'est-à-dire quand $P(XY) = P(X)P(Y)$. Si l'apparition de la prémisse n'influe pas sur l'apparition de la conclusion, il est inutile d'étudier des règles issues de motifs indépendants (*ou trop proches de celle-ci*).

Diverses mesures permettent de déterminer le point d'indépendance mais nous allons utiliser le lift [Brin et al., 1997b] défini comme suit $Lift(X \rightarrow Y) = \frac{P(XY)}{P(X)P(Y)}$. Le lift représente le rapport à l'indépendance de la règle. Plus la valeur du lift sera proche de 1 et plus la règle sera proche de l'indépendance. Si la règle $X \rightarrow Y$ possède une valeur du lift à 3, cela signifie qu'il y a trois fois plus d'individus qui vérifient X et Y que ce qui est attendu.

➤ **Équilibre**

Le point d'équilibre [Blanchard et al., 2005] intervient quand le nombre d'exemples est égal au nombre de contre-exemples, c'est-à-dire lorsqu'il existe autant de chances de voir se réaliser Y que $\overline{Y}$. À l'équilibre, l'égalité suivante est vérifiée $P(XY) = P(X\overline{Y})$. Afin de ne garder que les règles les plus pertinentes, il faut éliminer les règles où l'écart à l'équilibre est trop faible.

Plusieurs mesures sont capables de déterminer le point d'équilibre. Dans nos expérimentations, nous utilisons la mesure de la moindre contradiction [Azé and Kodratoff, 2004] définie comme suit $\textit{moindre contradiction}(X \rightarrow Y) = \frac{P(XY)-P(X\overline{Y})}{P(Y)}$. Plus la valeur de la moindre contraction est proche de 0 et plus la règle est proche de l'équilibre. Quand la valeur est supérieure à 0, la règle possède plus d'exemples que de contre-exemples.

➤ **Minimalité de la négation**

Lors de l'extraction de règles d'association négatives, un motif $\overline{X}$ représente la négation du motif X. Par définition, le motif $\overline{X}$ indique l'absence d'au moins un des items x_i $(i \in 1, ..., p)$ composant le motif X. Cette définition sous-entend que tout sur-ensemble d'un motif négatif fréquent est fréquent. Autrement dit, si le motif $\overline{X}$ est fréquent alors tout motif négatif composé de $\overline{X}$ le sera également. Par exemple si $\overline{X}$ est fréquent alors $\overline{XY}$ sera fréquent quel que soit le motif Y. Il va être important de prendre en compte cette propriété afin de ne pas se retrouver avec une multitude de règles négatives qui par ailleurs ne seront pas intéressantes. Par exemple, si la règle $Y \rightarrow \overline{X}$ est valide alors la règle $Y \rightarrow \overline{XZ}$ sera également valide quel que soit le motif Z. Cette dernière règle est donc inintéressante puisque le motif $\overline{XZ}$ n'est pas minimal. En effet il existe $\overline{X}$ qui est un sous-ensemble de $\overline{XZ}$ également fréquent. Pour parer ce problème, il faut donc s'assurer de la minimalité des motifs négatifs.

6.5.2 Analyse qualitative

Dans cette partie, nous allons vérifier pour chaque algorithme que les règles extraites respectent les propriétés présentées précédemment. Nous focalisons notre analyse sur les bases `Abalone`, `Ecoli` et `Iris` puisque ce sont les seules bases qui contiennent assez de résultats pour l'ensemble des méthodes étudiées. Nous allons vérifier les différentes propriétés pour les règles extraites avec un support minimum à 0,10 et 0,01 et une confiance à 0,85. Les algorithmes ne générant pas le même nombre de règles, nous évaluerons l'impact via le nombre de règles posant problème mais également via le pourcentage de règles élaguées.

➤ **Omniprésence**

Pour vérifier l'omniprésence, nous avons développé une autre version des algorithmes où nous supprimons l'ensemble des motifs omniprésents pour les motifs positifs et pour les motifs négatifs. Nous vérifions également que les motifs omniprésents ne sont

pas utilisés pour générer les motifs de taille supérieure. Par conséquent, pour cette partie de l'analyse, nous n'utilisons pas `ARA`. Dans cette expérimentation, nous fixons le seuil du support maximum à 0,80 puisque cela correspond au seuil que nous avions utilisé dans le chapitre précédent pour notre algorithme. Le tableau 6.6 restitue le nombre de règles éliminées ainsi que le pourcentage correspondant. Le nombre de règles éliminées correspond à la différence entre le nombre de règles obtenues dans les expérimentations du chapitre 5 moins le nombre de règles obtenues dans la version des algorithmes prenant en compte cette contrainte. Le pourcentage est donc calculé par rapport au nombre de règles sans la nouvelle contrainte. Notre algorithme `RAPN`, vérifiant déjà cette contrainte, ne sera pas présent dans cette étude.

	Abalone		Ecoli		Iris	
Support / Algorithmes	0,10	0,01	0,10	0,01	0,10	0,01
`Apriori`	7 878 (69%)	25 826 (69%)	1 687 (94%)	13 526 (91%)	0 (0%)	0 (0%)
[Wu et al., 2004]	2 568 (77%)	2 871 (67%)	318 (88%)	238 (88%)	0 (0%)	0 (0%)
[Antonie and Zaïane, 2004]	2 265 (68%)	3 649 (68%)	233 (95%)	1017 (92%)	0 (0%)	0 (0%)
[Cornelis et al., 2006]	9 734 (64%)	45 533 (72%)	3 814 (88%)	31 738 (89%)	83 (12%)	139 (10%)

Tableau 6.6 – Impact des motifs omniprésents sur les résultats

Cette étude montre que l'impact des motifs omniprésents sur les résultats est important et ce quel que soit l'algorithme utilisé. Globalement [Cornelis et al., 2006] semble être l'algorithme le plus impacté car l'ajout de la contrainte élague entre 10 et 12% des règles pour la base `Iris` alors que les autres algorithmes ne semblent pas impactés sur cette base. Pour les autres bases, à savoir `Abalone` et `Ecoli` entre 64% et 95% des règles sont élaguées en ajoutant la contrainte.

➤ Indépendance

Nous allons maintenant vérifier que les règles extraites sont issues de motifs indépendants. Pour se faire, nous allons utiliser la mesure du lift. Nous fixons la valeur minimum du lift à 1,1. Par conséquent, toutes les règles possédant une valeur pour le lift inférieure à ce seuil seront éliminées. Nous vérifierons également le nombre de règles dont la valeur du lift est inférieure à 1. En effet, un lift inférieur à 1 signale que l'étude de la règle ne s'est pas tournée vers le bon type de règles à extraire. Le tableau 6.7 restitue le nombre de règles élaguées par la contrainte ainsi que le pourcentage associé.

L'algorithme de [Wu et al., 2004] et notre méthode `RAPN` ne semblent pas ou peu impactés par l'indépendance. La méthode de [Antonie and Zaïane, 2004] est également peu impactée sur les bases `Abalone` et `Iris` mais est la plus impactée pour la base `Ecoli` avec 44 et 54% des règles élaguées. Les algorithmes `Apriori` et [Cornelis et al., 2006] sont les plus impactés et sont également les deux seuls algorithmes à extraire des règles possédant un lift inférieur à 1. Le taux de "mauvaises règles" atteint même 6% pour l'algorithme `Apriori` sur la base `Ecoli` avec un support à 0,1. Pour `Apriori`, l'extraction de règles avec un lift inférieur à 1, signifie que ces règles positives ont été extraites mais

Algorithmes \ Support	Abalone		Ecoli		Iris	
	0,10	0,01	0,10	0,01	0,10	0,01
`Apriori`	682 (6%)	3 062 (8%)	801 (45%)	4 938 (33%)	0 (0%)	0 (0%)
dont $lift < 1$	0 (0%)	48 (0,1%)	106 (6%)	428 (3%)	0 (0%)	0 (0%)
[Wu et al., 2004]	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
[Antonie and Zaïane, 2004]	0 (0%)	90 (2%)	108 (44%)	600 (54%)	0 (0%)	0 (0%)
[Cornelis et al., 2006]	699 (5%)	16 054 (25%)	1 091 (25%)	12 248 (35%)	38 (6%)	43 (3%)
dont $lift < 1$	0 (0%)	1 900 (3%)	151 (3%)	1 272 (4%)	7 (1%)	11 (1%)
`RAPN`	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)

TABLEAU 6.7 – Impact de l'indépendance sur les résultats

que ce sont les règles négatives $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ qui sont les plus pertinentes. Afin de prouver ceci prenons la règle $A \Rightarrow B$ où A représente l'item $alm2 =]-inf,\ 0,33]$ et B l'item $lip =]-inf,\ 0,65]$ extraite par `Apriori` sur `Ecoli` avec un seuil minimum du support à 0,1. Le support de la règle est de 0,21 et la confiance 0,93. De plus le support de la prémisse est de 0,22 et celui de la conclusion est de 0,97. Le lift de cette règle est égal à 0,96 ce qui dénote qu'une des règles négatives est plus intéressante.

Prenons la règle $\overline{A} \Rightarrow B$ et vérifions. Le support de cette règle négative est donc égal à $sup(\overline{A}B) = sup(B) - sup(AB) = 0,97 - 0,21 = 0,76$. Sa confiance est égale à $conf(\overline{A} \Rightarrow B) = \frac{sup(\overline{A}B)}{sup(\overline{A})} = \frac{0,76}{1-sup(A)} = \frac{0,76}{1-0,22} = \frac{0,76}{0,78} \simeq 0,97$. Le support et la confiance de cette règle sont valides puisqu'ils sont supérieurs à $min_{sup} = 0,10$ et $min_{conf} = 0,85$. Vérifions maintenant la valeur pour le lift. Celui-ci est donc égal à $lift(\overline{A} \Rightarrow B) = \frac{sup(\overline{A}B)}{sup(\overline{A}) \times sup(B)} = \frac{0,76}{0,78 \times 0,97} \simeq 1$. La règle négative $\overline{A} \Rightarrow B$ est donc plus intéressante que la règle $A \Rightarrow B$ puisque son support, sa confiance et son lift sont meilleurs. Voilà pourquoi `Apriori` extrait parfois une règle positive même si en réalité c'est la règle négative qui est la plus intéressante. Cependant même si la règle négative est plus pertinente que la règle positive, sa valeur du lift la situe trop proche de l'indépendance.

➤ Équilibre

Nous vérifions ensuite que les règles extraites sont suffisamment loin du point d'équilibre. Pour se faire, nous allons utiliser la moindre contradiction. Nous fixons la valeur minimum de cette mesure à 0,15 puisque c'est à partir de ce seuil que les résultats commencent à se différencier d'un algorithme à l'autre. Par conséquent, toutes les règles possédant une valeur pour la moindre contradiction inférieure à ce seuil seront éliminées. Nous vérifions également le nombre de règles dont la valeur pour la moindre contradiction est inférieure à 0. En effet, une valeur négative signale que l'étude de la règle ne s'est pas tournée vers le bon type de règles à extraire. Le tableau 6.8 synthétise les résultats observés.

Lors de cette expérimentation, les résultats obtenus par les méthodes de [Wu et al., 2004] et [Antonie and Zaïane, 2004] ne sont pas du tout impactés et ceci quelle que soit la base de données testée et quelle que soit la valeur du support. Parmi les autres

Algorithmes \ Support	Abalone		Ecoli		Iris	
	0,10	0,01	0,10	0,01	0,10	0,01
`Apriori`	505 (4%)	22 262 (60%)	470 (26%)	11 060 (75%)	0 (0%)	137 (29%)
[Wu et al., 2004]	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
[Antonie and Zaïane, 2004]	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
[Cornelis et al., 2006]	676 (4%)	42 260 (66%)	988 (23%)	28 590 (81%)	18 (3%)	549 (41%)
`RAPN`	9 (0,2%)	4 100 (28%)	1 (0,7%)	348 (35%)	0 (0%)	43 (6%)

Tableau 6.8 – Impact de l'équilibre sur les résultats

méthodes, c'est notre méthode qui semble la moins impactée. En effet, sur les trois expérimentations utilisant un support à 0,10, la moindre contradiction élimine moins de 1% de nos résultats. `Apriori` et [Cornelis et al., 2006] sont les plus impactés puisque entre 4% et 81% de leurs résultats sont élagués par la moindre contradiction fixée à 0,15. Comme nous l'avons signalé dans le paragraphe précédent, nous avons également vérifié que certaines règles possédaient une valeur inférieure à 0 pour la moindre contradiction. Les résultats n'apparaissent pas dans le tableau car aucune règle avec une valeur inférieure à 0 n'a été trouvée. Par conséquent, la moindre contradiction laisse supposer que l'ensemble des règles trouvées par tous les algorithmes sont du bon type. Passons maintenant à la dernière étude.

➤ Minimalité de la négation

La dernière étude consiste à vérifier que les règles négatives extraites respectent bien la contrainte de minimalité des motifs négatifs. Pour se faire, nous avons implémenté d'autres versions en ajoutant cette contrainte aux différentes méthodes. L'algorithme `Apriori` a été retiré de cette étude puisqu'il n'extrait que des règles positives. De plus, l'algorithme de [Cornelis et al., 2006] et `RAPN` respectent déjà cette contrainte et n'apparaissent pas non plus dans cette étude. Les résultats obtenus sont visibles dans le tableau 6.9.

Algorithmes \ Support	Abalone		Ecoli		Iris	
	0,10	0,01	0,10	0,01	0,10	0,01
[Wu et al., 2004]	900 (71%)	1 578 (83%)	168 (71%)	102 (70%)	21 (27%)	30 (35%)
[Antonie and Zaïane, 2004]	1 165 (73%)	1 712 (80%)	144 (96%)	618 (94%)	0 (0%)	0 (0%)

Tableau 6.9 – Impact de la minimalité de la négation sur les résultats

Les deux méthodes testées sont fortement impactées par la contrainte de minimalité de la négation. L'ajout de cette contrainte supprime entre 27% et 83% des règles extraites pour [Wu et al., 2004]. Quant à [Antonie and Zaïane, 2004], l'ajout de la contrainte n'a aucun impact sur la base `Iris`, cependant 96% et 94% des résultats sont élagués sur la base `Ecoli` pour un support à 0,10 et 0,01.

6.6 Conclusion

`ARA` est un logiciel que nous avons développé pour répondre à notre besoin d'analyse. Nous avons fait en sorte de le rendre le plus générique possible afin que d'autres personnes puissent également l'utiliser. Étant open source, il est aisé de pouvoir ajouter de nouveaux types de règles mais également d'ajouter d'autres mesures de qualité, voire même d'autres onglets pour d'autres types d'étude.

La première partie de notre analyse a consisté à comparer les règles extraites par chaque algorithme. Les algorithmes n'extrayant pas le même nombre de règles, nous nous sommes demandés si la différence provenait du fait que certains algorithmes possèdent des contraintes plus fortes qui entraînent l'élagage de règles par rapport aux autres. Nous avons vérifié cette hypothèse avec deux expérimentations en vérifiant à chaque fois le nombre de règles communes extraites par les algorithmes sélectionnés ainsi que le nombre de règles spécifiques à chacun. Dans la première expérimentation, nous avons comparé les algorithmes sur notre exemple fil-rouge où nous avons pu remarquer que l'ensemble des règles positives était extrait par `Apriori`. Les autres algorithmes extrayaient donc un sous-ensemble des règles positives extraites par `Apriori`. Pour les règles négatives, nous n'avons pas observé le même comportement. En effet, `ARA` nous indique qu'il existe 77 règles négatives différentes trouvées lors nos expérimentations sur l'exemple fil-rouge. Or [Antonie and Zaïane, 2004] qui est l'algorithme qui extrait le plus de règles négatives n'en extrait que 64. Même en comptant les 3 règles négatives du nouveau type que notre algorithme est le seul à extraire, il reste 10 règles négatives extraites par un autre algorithme qu' [Antonie and Zaïane, 2004] (*ou plusieurs*). Par ailleurs, aucune règle négative n'est commune à l'ensemble des algorithmes. Afin de vérifier si ce comportement ne provenait pas de la petite taille de l'exemple choisi, nous avons reproduit la même étude sur la base de données `Abalone` en utilisant un support à 0,01 et une confiance à 0,85. Cette deuxième expérimentation a confirmé nos précédents résultats. À partir de ce constat comment expliquer une telle différence dans les résultats pour les règles négatives ? Cette différence provient en partie du nouveau type de règles extraites par notre algorithme, des seuils utilisés pour les différentes mesures mais également de la méthode de construction des motifs appliqués pour générer les règles négatives. Cette première étude peut sembler sans grande importance mais permet de trouver les règles respectant l'ensemble des critères de sélection des algorithmes étudiés. En effet, les règles communes sont peut être les plus intéressantes pour un expert puisqu'elles respectent un maximum de critères différents.

Dans la deuxième partie de ce chapitre, nous avons proposé quatre critères pour évaluer de façon générique les règles extraites, à savoir l'omniprésence des motifs, l'indépendance, l'équilibre et la minimalité de la négation. Ces critères ne sont pas exhaustifs mais permettent d'avoir une première idée de la qualité des règles extraites par les différents algorithmes. Nous avons commencé par démontrer que si ces critères n'étaient pas respectés, la règle étudiée n'était pas intéressante. Dans la deuxième partie de notre analyse, nous avons donc cherché à vérifier si les règles extraites par les différents algorithmes respectaient ces critères. Pour certains critères, nous avons modifié les algorithmes pour prendre en compte le nouveau critère. Ce fut le cas pour l'omniprésence des motifs et la minimalité de la négation. Pour les autres critères, nous avons proposé l'utilisation de mesures de qualité de la littérature : le lift pour mesurer l'écart à l'indépendance et la mesure de moindre contradiction pour mesurer l'écart à l'équilibre. En ce qui concerne l'omniprésence et la minimalité de la négation, notre algorithme `RAPN` n'est pas impacté

par ces deux problèmes puisque nous avions pris en considération ces critères au moment de son élaboration. Concernant l'omniprésence, on peut voir que tous les algorithmes sont fortement impactés (*entre 64% et 95% excepté pour* `Iris`) par ce problème. Globalement [Cornelis et al., 2006] semblent les plus impactés puisqu'ils sont les seuls à extraire des règles provenant de motifs omniprésents pour `Iris` (*entre 10% et 12%*). Pour la minimalité de la négation, `Apriori` ne fait pas partie de l'étude car il ne génère pas de règles négatives. [Cornelis et al., 2006] est également exclu de l'étude car la méthode utilisée pour la construction des motifs négatifs empêche ce problème. Les deux algorithmes restant sont fortement impactés puisque l'ajout de cette contrainte supprime entre 27% et 96% des règles extraites auparavant. Ici l'algorithme le plus sensible à ce problème semble être celui de [Wu et al., 2004]. Concernant l'indépendance des données [Wu et al., 2004] et notre méthode `RAPN` ne semblent pas du tout impactés. Les autres algorithmes suppriment entre 1% et 45%, cependant certaines règles extraites possèdent une valeur du lift inférieure à 1, ce qui signale que l'étude de la règle ne s'est pas tournée vers le bon type de règles à extraire. C'est le cas pour `Apriori` et [Cornelis et al., 2006]. Et enfin concernant l'équilibre, [Wu et al., 2004] et [Antonie and Zaïane, 2004] ne sont pas du tout impactés. Même si nous avons pris ce critère en considération lors de l'élaboration de notre méthode, nous élaguons encore des règles trop proches de l'équilibre (*entre 0% et 35%*) selon la mesure de moindre contradiction. Néanmoins `Apriori` et [Cornelis et al., 2006] sont les algorithmes les plus touchés (*entre 3% et 81%*). En étudiant l'ensemble de ces critères, notre algorithme semble extraire les règles possédant la meilleure qualité, cependant certaines faiblesses sont encore présentes.

Les quatre critères retenus permettent de se faire une première idée sur la qualité de la règle, cependant deux autres critères nous semblant importants ont été mis de côté. Le premier critère est la similarité qui permet d'évaluer la ressemblance entre deux individus. Dans le domaine des règles d'association, deux items vont être considérés comme similaires si pour chaque transaction de la base de données, ces deux items possèdent les mêmes valeurs. Supposons deux items Y et Z similaires, si le motif XY est fréquent alors le motif XZ le sera également. On retrouve le même comportement pour la confiance des règles à partir de ces motifs. Par conséquent, si la similarité est connue de l'utilisateur, l'extraction peut se concentrer sur un seul de ces items afin d'élaguer certaines règles redondantes dès la construction des motifs. Le second critère est la redondance. Les algorithmes d'extraction de règles d'association ont tendance à générer un trop grand nombre de règles, règles qui sont plus la plupart inintéressantes et/ou redondantes. La définition formelle de la redondance varie en fonction des chercheurs. Moins formellement, une règle est dite redondante si elle n'apporte pas d'information supplémentaire. Cependant, la majorité des travaux se sont focalisés sur la redondance concernant les règles positives. Or dans notre cas, nous générons également des règles négatives. Notre algorithme peut à la fois extraire des règles $X \Rightarrow Y$, $\overline{X} \Rightarrow \overline{Y}$ et $\ddot{X} \Rightarrow \ddot{Y}$ ou des règles du type $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\ddot{X} \Rightarrow \ddot{Y}$. Dans certains cas de figures, nous avons donc trois règles issues des mêmes prémisses et conclusions positives. Les informations apportées par ces règles sont relativement similaires mais l'une d'entre elles aura forcément plus de valeur aux yeux de l'utilisateur métier, reste à savoir laquelle ?

Conclusion et perspectives

L'extraction de connaissances à partir des données est le processus de découverte de connaissances utiles à partir d'un jeu de données. Ce processus se décompose en plusieurs étapes mais nous nous sommes intéressés uniquement, dans ce manuscrit, à l'étape qui consiste à extraire les connaissances : la fouille de données. Les connaissances extraites peuvent prendre plusieurs formes, mais nous nous sommes concentrés sur les règles d'association positives et négatives.

Les algorithmes d'extraction de règles d'association positives et négatives de la littérature génèrent un nombre important de règles inintéressantes en utilisant un parcours de recherche des règles non optimisé. Afin de combler ces deux failles, nous avons proposé dans cette thèse une nouvelle approche basée sur l'algorithme fondateur `Apriori` pour extraire des règles d'association positives et négatives intéressantes. Notre algorithme a également la particularité d'extraire un nouveau type de règles négatives que les autres algorithmes ne recherchent pas : les règles possédant des conjonctions d'items négatifs en prémisse et en conclusion. Par conséquent, en plus de rechercher les règles $X \Rightarrow Y$, $\overline{X} \Rightarrow Y$, $X \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow \overline{Y}$ comme le font les méthodes classiques, nous extrayons également les règles : $\overline{x_1}..\overline{x_p} \Rightarrow \overline{y_1}..\overline{y_q}$.

Les apports de notre approche

Afin de réduire le nombre de règles inintéressantes, nous avons choisi de ne plus baser notre extraction sur les motifs fréquents comme le font les autres méthodes que nous avons pu étudier dans ce manuscrit. Ce choix se justifie par la possible présence de motifs omniprésents dans les bases de données à analyser. Ces motifs omniprésents vont conduire à des règles non valides pour la confiance, à des règles trop proches de l'indépendance ou encore à des règles redondantes. Par conséquent, ces motifs omniprésents vont amener, dans le cas où l'extraction est possible, à extraire des règles inintéressantes. Notre premier objectif fut donc de supprimer ces motifs omniprésents de l'étude. Pour se faire, nous avons proposé de baser notre extraction sur les motifs raisonnablement fréquents qui permettent d'écarter les motifs omniprésents dès la première phase de l'extraction.

Nous avons également ajouté une autre contrainte lors de la recherche des motifs raisonnablement fréquents : le support minimum pour les conjonctions d'items négatifs. Cette contrainte supplémentaire renforce la contrainte du support maximum et permet d'extraire des motifs XY plus intéressants. C'est également l'ajout de cette contrainte qui nous a permis de découvrir les règles intéressantes du nouveau type.

Une fois les motifs raisonnablement fréquents extraits, la seconde étape a consisté à générer les règles à partir de ces motifs. Afin de connaître les types de règles à étudier, nous avons utilisé la mesure M_G. La mesure M_G commence par déterminer la zone d'appartenance de la règle positive en comparant sa confiance au support de sa conclusion. Cette étape nous a indiqué si nous devions étudier les règles $X \Rightarrow Y$ et $\overline{X} \Rightarrow \overline{Y}$ ou bien si nous devions étudier les règles $\overline{X} \Rightarrow Y$ et $X \Rightarrow \overline{Y}$. Cette mesure nous a permis donc d'optimiser le parcours de recherche puisque comme nous l'avons démontré seule la moitié des règles peuvent être intéressantes. Cette mesure possède un autre avantage puisqu'elle permet d'élaguer un autre type de règles inintéressantes : les règles trop proches de l'équilibre.

Afin d'optimiser le parcours de recherche des règles, nous avons utilisé la mesure M_G comme nous l'avons expliqué précédemment mais nous avons également eu recours à deux méta-règles afin de déduire l'intérêt d'une règle à partir d'une autre. La première méta-règle MR_9 correspond à la propriété de la confiance. Cette propriété abandonnée par les autres méthodes, nous a permis d'élaguer certaines règles du nouveau type. La deuxième méta-règle MR_4 permet d'inférer la non validité des règles $\overline{X} \Rightarrow \overline{Y}$ et $\overline{X} \Rightarrow Y$ à partir des règles $X \Rightarrow Y$ et $X \Rightarrow \overline{Y}$ respectivement. Comme nous l'avons vu dans les expérimentations, l'ajout de la méta-règle MR_9 accélère fortement les traitements. Quant à l'effet de la méta-règle MR_4 elle impacte peu dans nos expérimentations puisque les conditions nécessaires à son application ne sont pas souvent réunies. Cependant, l'effet pourrait être différent sur d'autres bases de données.

Lors de la recherche des règles négatives, nous avons ajouté une contrainte afin de s'assurer de l'intérêt de la règle extraite. En effet, nous avons vérifié que la règle est minimale. Pour se faire, après avoir extrait les motifs raisonnablement fréquents, nous avons recherché les motifs négatifs minimaux raisonnablement fréquents. Cette étape intermédiaire, nous est utile lors de la recherche des règles où nous vérifions que chaque prémisse et conclusion négatives sont bien présentes dans la liste des motifs minimaux raisonnablement fréquents.

Les résultats

Nous avons implémenté les trois méthodes étudiées dans le chapitre 2 ainsi que notre méthode dans le logiciel `Weka`. Nos expérimentations sont encourageantes. Globalement, toutes les améliorations que nous avons proposées dans notre algorithme semblent avérées. Certaines propositions semblent plus significatives, c'est le cas pour l'utilisation du support maximum, de la méta-règle MR_9 et de la contrainte de minimalité des motifs négatifs. L'utilisation de la mesure M_G nous permet de connaître les types de règles à étudier, cependant son impact pour éliminer les règles trop proches de l'équilibre semble moins important. En effet, quand on la compare au facteur de certitude, l'impact sur le nombre de règles élaguées est minime. L'expérimentation avec la moindre contradiction semble également montrer que nos règles restent trop proches de l'équilibre. Des investigations supplémentaires semblent nécessaires afin de vérifier si cela provient des seuils choisis, des bases de données sélectionnées, ou de l'implémentation puisque théoriquement nous ne devrions pas obtenir ces résultats. Quant à la comparaison avec les autres algorithmes notre algorithme semble être le plus rapide. En effet, certains

algorithmes mettent plus de huit heures pour restituer les résultats alors que nous les obtenons en deux minutes. En ce qui concerne les règles extraites, notre algorithme semble le plus équilibré puisque nous n'extrayons jamais plusieurs millions de règles, nous n'extrayons pas non plus, plus de règles négatives que de règles positives (*si nous omettons le nouveau type de règles*), nous arrivons à générer à chaque expérimentation ou presque des règles négatives, et aucun comportement anormal n'a pu être détecté comme ce fut le cas sur d'autres méthodes. Le comportement anormal détecté dans d'autres méthodes provient du fait que le nombre de règles négatives diminue quand le seuil du support minimum diminue. Ce comportement anormal rend les interprétations des résultats plus difficiles. Le principal point faible de notre méthode provient du grand nombre de règles du nouveau type générées. En effet, plus de la moitié des règles extraites sont de ce type. Il faudra donc fournir un travail supplémentaire car il est peu probable que toutes ces règles soient intéressantes, ou laisser la possibilité à l'utilisateur de ne pas générer ce type de règles si sa problématique ne nécessite pas l'extraction de telles règles.

Nous avons ensuite continué notre étude en poursuivant notre analyse quantitative et en comparant qualitativement notre méthode avec les autres méthodes de la littérature. Pour faciliter ce travail, nous avons développé le logiciel `ARA` qui analyse les fichiers de résultats issus du logiciel `Weka`. `ARA` commence par nous renseigner sur le nombre de règles communes et spécifiques à chaque algorithme étudié. Nous avons donc, dans un premier temps, comparer les règles extraites par les différents algorithmes sur notre exemple fil-rouge. Puis, nous avons validé nos résultats en effectuant la même analyse sur une des expérimentations de la base de données `Abalone`. Nous avons pu observer que les différents algorithmes d'extraction de règles positives et négatives extraient un sous-ensemble des règles positives extraites par `Apriori`. Nous avons également remarqué qu'un certain nombre de règles positives était commun à l'ensemble des algorithmes. Concernant les règles négatives, nous n'avons pas observé ce même comportement. En effet, chaque algorithme extrait des règles négatives spécifiques et aucune règle n'est commune à l'ensemble des méthodes. L'intérêt de cette première étude peut sembler minime, cependant les règles communes sont des règles très intéressantes puisqu'elles respectent l'ensemble des contraintes exigées par les algorithmes sélectionnés. Dans un second temps, nous nous sommes intéressés au problème de la qualité des règles d'association extraites. Nous exposons plusieurs propriétés intéressantes qu'une règle devrait respecter. Nous avons pu trouver six propriétés qu'il faut vérifier : la non omniprésence des motifs, un écart significatif par rapport à l'indépendance, un écart significatif par rapport à l'équilibre, la minimalité des motifs négatifs, la non similarité et la non redondance. Parmi ces critères, nous avons proposé des méthodes simples pour vérifier les quatre premiers critères. Nous avons donc implémenté des variantes des algorithmes pour vérifier certains critères () tandis que pour les autres nous avons fait appel à des mesures de qualité de la littérature : le lift et la moindre contradiction. Ces mesures ont été implémentées dans `ARA` afin de pouvoir effectuer des comparaisons sur les règles extraites par les différentes méthodes. En étudiant l'intérêt des règles vis-à-vis de ces quatre critères, nous nous sommes rendus compte que notre algorithme semble le plus pertinent même si des faiblesses sont encore présentes.

Des limites à s'affranchir

Dans ce manuscrit, nous avons proposé une méthode qui effectue une recherche exhaustive des règles d'association positives et négatives valides. Cette recherche de l'ensemble des règles va compliquer la tâche de l'utilisateur métier pour discerner les connaissances réellement utiles dont il a besoin. Ce problème, déjà présent lors de l'extraction des règles positives, s'amplifie avec la recherche simultanée des règles négatives. Cependant, la prise en compte des règles négatives est nécessaire puisqu'elles permettent d'étendre les connaissances et renferment des informations non accessibles avec les règles positives. Par exemple en chimie, si deux molécules X et Y produisent un effet Z représenté par la règle $XY \Rightarrow Z$, il est possible que l'ajout d'une troisième molécule W change la donne en inversant l'effet : $XYW \Rightarrow \overline{Z}$. Même si la règle $W \Rightarrow \overline{Z}$ existe, les règles précédentes indiquent que l'effet de la molécule W va parvenir à inverser l'effet des molécules XY. La règle $XYW \Rightarrow \overline{Z}$ est donc être une règle d'exception à la règle générale $XY \Rightarrow Z$. D'autres règles négatives vont permettre de découvrir plus facilement des substituts. Par exemple, les personnes qui ne boivent pas de vin boivent de l'eau $\overline{vin} \Rightarrow eau$ ou des sodas $\overline{vin} \Rightarrow sodas$. On peut donc considérer que l'eau et les sodas sont des substituts au vin. Cet exemple bien que trivial, laisse entrevoir les possibilités offertes par la recherche de ces règles de substitution. Les règles $\overline{X} \Rightarrow \overline{Y}$ peuvent également s'avérer intéressantes dans certains cas. Par exemple, si la règle $\overline{X} \Rightarrow \overline{Y}$ vient en complément de la règle $X \Rightarrow Y$, la présence ou l'absence de X a une incidence directe sur la présence ou l'absence de Y ; les motifs X et Y sont donc fortement corrélés.

En fonction du domaine d'étude, certains types de règles négatives seront plus intéressants que d'autres. Il faut donc définir en fonction de la problématique vers quels types de règles se tourner et également connaître les motifs cibles afin de faciliter et d'accélérer la découverte de connaissances. Même si les coûts computationnels et la difficulté d'interprétation augmentent, les règles négatives doivent venir en complément des règles positives puisque la connaissance induite par ces règles est inaccessible si seules les règles positives sont extraites. L'utilisateur métier doit donc s'affranchir des limites afin de pouvoir pleinement exploiter les données qu'il a en sa possession.

Les perspectives

Les résultats de nos travaux offrent plusieurs perspectives de recherche :

— il serait intéressant d'adapter nos améliorations sur l'algorithme `FP-Growth` [Han et al., 2000]. Même si `Apriori` est l'un des algorithmes les plus connus pour l'extraction de règles d'association, `FP-Growth` est réputé pour être plus rapide. Cet algorithme utilise une structure compacte appelée `FP-Tree` qui permet d'extraire les motifs fréquents sans générer de candidats. De plus, alors qu'`Apriori` interroge la base pour chaque niveau de motifs candidats générés, `FP-Growth` nécessite seulement deux passages, ce qui accélère encore les traitements.

— nous aimerions également étendre notre algorithme à la recherche des règles du type $(X_1 \wedge X_2) \vee X_3 \Rightarrow Y_1 \wedge (Y_2 \vee Y_3)$, c'est-à-dire aux règles possédant en prémisse et/ou en conclusion des conjonctions ou disjonctions de motifs positifs ou négatifs. Cette extension permettrait d'extraire l'ensemble des règles positives et négatives intéressantes au lieu de ne se focaliser que sur certains types de règles négatives. Ce-

pendant, des critères de sélection supplémentaires doivent être trouvés au préalable afin de limiter le nombre de règles extraites.

— une étude plus approfondie sur l'intérêt et la redondance des règles est également nécessaire. Le principal défaut des algorithmes d'extraction de règles d'association est qu'ils génèrent trop de règles. En cherchant les règles négatives du nouveau type en plus des règles traditionnelles, nous amplifions encore ce problème. Même si nous essayons de répondre à ce problème, le nombre de règles générées est encore trop important et l'utilisateur métier est submergé si nous lui fournissons autant de résultats. Nous avons proposé un algorithme d'extraction de règles d'association positives et négatives qui extrait l'ensemble des règles valides, mais il serait plus intéressant pour un utilisateur métier de lui permettre de choisir quel type de règles extraire et sur quels motifs se concentrer.

— même si notre méthode est la plus rapide selon nos expérimentations, son implémentation dans `Weka` n'est pas adaptée pour extraire des données sur de très grandes bases de données. En effet, nos expérimentations se sont focalisées sur des bases possédant moins de 50 items différents, ce qui est très loin des milliers d'articles pouvant être présents dans les bases de la grande distribution par exemple. Il serait donc intéressant de proposer une version parallèle de notre méthode en utilisant le framework `MapReduce` notamment utilisé par `Hadoop` et qui permet de travailler avec des pétaoctects de données.

— dans le dernier chapitre nous avons comparé les différentes méthodes en mesurant l'intérêt des règles. Nous avons vérifié que ces règles possédaient différentes propriétés que nous jugions intéressantes. Même si ces critères sont objectifs, comment savoir si les règles qui les respectent sont réellement utiles pour l'utilisateur métier ? Par exemple, dans le cas où l'utilisateur métier souhaite prédire un comportement, le problème d'extraction des règles est beaucoup plus simple. En effet, il suffit d'extraire uniquement des règles dont la conclusion est composée uniquement du ou des motifs à prédire. En utilisant ces règles dans un classifieur, nous pouvons vérifier quel ensemble de règles prédit le mieux et par conséquent nous pouvons comparer les différentes méthodes sur d'autres critères.

Bibliographie

[UCI, 2015] (2015). Uci machine learning repository. http://archive.ics.uci.edu/ml/.

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the International Conference on Management of Data*, SIGMOD, pages 207–216.

[Agrawal and Srikant, 1994] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB, pages 487–499.

[Amir et al., 1997] Amir, A., Feldman, R., and Kashi, R. (1997). A new and versatile method for association generation. In *Proceedings of the 1st European Symposium of Data Mining and Knowledge Discovery*, PKDD, pages 221–231.

[Antonie and Zaïane, 2004] Antonie, M.-L. and Zaïane, O. R. (2004). Mining positive and negative association rules : an approach for confined rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD, pages 27–38.

[Azé and Kodratoff, 2004] Azé, J. and Kodratoff, Y. (2004). Mesures de qualité pour la fouille de données. In *Extraction de "pépites" de connaissance dans les données : une nouvelle approche et une étude de la sensibilité au bruit.*, pages 147–170.

[Blanchard et al., 2005] Blanchard, J., Guillet, F., Briand, H., and Gras, R. (2005). Assessing rule interestingness with a probabilistic measure of deviation from equilibrium. In *Proceedings of the 11th International Symposium on Applied Stochastic Models and Data Analysis*, ASMDA, pages 191–200.

[Boulicaut et al., 2001] Boulicaut, J.-F., Bykowski, A., and Jeudy, B. (2001). Towards the tractable discovery of association rules with negations. In *Flexible Query Answering Systems*, pages 425–434.

[Brin et al., 1997a] Brin, S., Motwani, R., and Silverstein, C. (1997a). Beyond market baskets : Generalizing association rules to correlations. In *Proceedings of the International Conference on Management of Data*, SIGMOD, pages 265–276.

[Brin et al., 1997b] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997b). Dynamic itemset counting and implication rules for market basket data. pages 255–264.

[Cai et al., 1990] Cai, Y., Cercone, N., and Han, J. (1990). An attribute-oriented approach for learning classification rules from relational databases. In *Proceedings of the 6th International Conference on Data Engineering*, ICDE, pages 281–288.

[Cohen, 1988] Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. Routledge Academic, 2 edition.

[Cornelis et al., 2006] Cornelis, C., Yan, P., Zhang, X., and Chen, G. (2006). Mining positive and negative association rules from large databases. In *Cybernetics and Intelligent Systems*, CIS, pages 613–618.

[Dong et al., 2007a] Dong, X., Niu, Z., Shi, X., Zhang, X., and Zhu, D. (2007a). Mining both positive and negative association rules from frequent and infrequent itemsets. In *Advanced Data Mining and Applications*, ADMA, pages 122–133.

[Dong et al., 2006] Dong, X., Sun, F., Han, X., and Hou, R. (2006). Study of positive and negative association rules based on multi-confidence and chi-squared test. In *Advanced Data Mining and Applications*, ADMA, pages 100–109.

[Dong et al., 2007b] Dong, X., Zheng, Z., Niu, Z., and Jia, Q. (2007b). Mining infrequent itemsets based on multiple level minimum supports. In *Proceedings of the 2nd International Conference on Innovative Computing, Information and Control*, ICICIC, pages 528–528.

[Duval et al., 2007] Duval, B., Salleb, A., and Vrain, C. (2007). On the discovery of exception rules : A survey. In *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 77–98.

[Fisher, 1925] Fisher, R. A. (1925). *Statistical methods for research workers*. Genesis Publishing Pvt Ltd.

[Fleury, 1996] Fleury, L. (1996). *Extraction de connaissances dans une base de données pour la gestion des ressources humaines*. PhD thesis, Université de Nantes.

[Frawley et al., 1991] Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. (1991). Knowledge discovery in databases : An overview. pages 1–27.

[Gan et al., 2006] Gan, M., Zhang, M., and Wang, S. (2006). Extended negative association rules and the corresponding mining algorithm. In *International Conference on Machine Learning and Cybernetics*, ICMLC, pages 159–168.

[Gasmi et al., 2007] Gasmi, G., Yahia, S. B., Nguifo, E. M., and Bouker, S. (2007). Extraction of association rules based on literalsets. In *Data Warehousing and Knowledge Discovery*, DAWAK, pages 293–302.

[Gras, 1979] Gras, R. (1979). *Contribution à l'étude expérimentale et à l'analyse de certaines acquisitions cognitives et de certains objectifs didactiques en mathématiques*. PhD thesis, Université de Rennes.

[Gras and Kuntz, 2006] Gras, R. and Kuntz, P. (2006). Discovering R-rules with a directed hierarchy. *Soft computing*, 10(5) :453–460.

[Gras et al., 2007] Gras, R., Kuntz, P., and Suzuki, E. (2007). Règles d'exception en analyse statistique implicative. In *Actes des cinquièmes journées Extraction et Gestion des Connaissances*, EGC, pages 87–98.

[Gras et al., 2013] Gras, R., Suzuki, E., and Kuntz, P. (2013). Règle et r-règle d'exception en analyse statistique implicative. pages 305–314.

[Guillaume, 2000] Guillaume, S. (2000). *Traitement des données volumineuses. Mesures et algorithmes d'extraction des règles d'association et règles ordinales*. PhD thesis, Université de Nantes.

[Guillaume, 2010] Guillaume, S. (2010). Améliorations de la mesure d'intérêt M_{GK}. In *Actes des XVIIèmes rencontres de la Société Francophone de Classification*, pages 41–45.

[Guillaume and Papon, 2012] Guillaume, S. and Papon, P.-A. (2012). Méta-règles pour la génération de règles négatives. In *Actes de la 12ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Revue des Nouvelles Technologies de l'Information RNTI-E-23, ISBN 978-2-7056 8310 8*, EGC, pages 231–236.

[Guillaume and Papon, 2013a] Guillaume, S. and Papon, P.-A. (2013a). Étude comparative d'extraction de règles d'association positives et négatives et optimisations. In *Apprentissage Artificiel et Fouille de Données, Revue des Nouvelles Technologies de l'Information, vol. RNTI-A.6*, AAFD, pages 27–56.

[Guillaume and Papon, 2013b] Guillaume, S. and Papon, P.-A. (2013b). Extraction optimisée de règles d'association positives et négatives (RAPN). In *Actes de la 13ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Revue des Nouvelles Technologies de l'Information RNTI-E-24, ISBN 978-2-7056 8656 7*, EGC, pages 157–168.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software : An update. *SIGKDD Explorations Newsletter*, 11(1) :10–18.

[Hämäläinen, 2012] Hämäläinen, W. (2012). Kingfisher : an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and information systems*, 32(2) :383–414.

[Hamrouni et al., 2010] Hamrouni, T., Yahia, S. B., and Nguifo, E. M. (2010). Generalization of association rules through disjunction. *Annals of Mathematics and Artificial Intelligence*, 59(2) :201–222.

[Han et al., 2000] Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, pages 1–12.

[Hussain et al., 2000] Hussain, F., Liu, H., Suzuki, E., and Lu, H. (2000). Exception rule mining with a relative interestingness measure. In *Proceedings of the 4th Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, PAKDD, pages 86–97.

[Kadir et al., 2011] Kadir, A. S. A., Bakar, A. A., and Hamdan, A. R. (2011). Frequent absence and presence itemset for negative association rule mining. In *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications*, ISDA, pages 965–970.

[Kouris et al., 2007] Kouris, I. N., Makris, C. H., and Tsakalidis, A. K. (2007). Uncovering hidden associations through negative itemsets correlations. In Global, I., editor, *Intelligent Databases : Technologies and Applications*, pages 1–28.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86.

[Lavrac et al., 1999] Lavrac, N., Flach, P. A., and Zupan, B. (1999). Rule evaluation measures : A unifying view. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, ILP, pages 174–185.

[Lerman and Azé, 2007] Lerman, I.-C. and Azé, J. (2007). A new probabilistic measure of interestingness for association rules, based on the likelihood of the link. In *Quality Measures in Data Mining. Studies in Computational Intelligence*, pages 207–236. Springer.

[Lerman and Guillaume, 2013] Lerman, I.-C. and Guillaume, S. (2013). Comparing two discriminant probabilistic interestingness measures for association rules. In *Studies in Computational Intelligence*, volume 471, pages 59–83. Springer.

[Liu et al., 1999a] Liu, B., Hsu, W., and Ma, Y. (1999a). Mining association rules with multiple minimum supports. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, KDD, pages 337–341.

[Liu et al., 1999b] Liu, H., Lu, H., Feng, L., and Hussain, F. (1999b). Efficient search of reliable exceptions. In *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, PAKDD, pages 194–203.

[Missaoui et al., 2008] Missaoui, R., Nourine, L., and Renaud, Y. (2008). Generating positive and negative exact rules using formal concept analysis : problems and solutions. In *Formal Concept Analysis*, pages 169–181.

[Missaoui et al., 2010] Missaoui, R., Nourine, L., and Renaud, Y. (2010). An inference system for exhaustive generation of mixed and purely negative implications from purely positive ones. In *CLA*, pages 271–282.

[Missaoui et al., 2012] Missaoui, R., Nourine, L., and Renaud, Y. (2012). Computing implications with negation from a formal context. *Fundamenta Informaticae*, 115(4) :357–375.

[Padmanabhan and Tuzhilin, 1998] Padmanabhan, B. and Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, KDD, pages 94–100.

[Padmanabhan and Tuzhilin, 1999] Padmanabhan, B. and Tuzhilin, A. (1999). Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems*, 27(3) :303–318.

[Padmanabhan and Tuzhilin, 2000] Padmanabhan, B. and Tuzhilin, A. (2000). Small is beautiful : Discovering the minimal set of unexpected patterns. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, KDD, pages 54–63.

[Pearson, 1896] Pearson, K. (1896). Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 187 :253–318.

[Pearson, 1900] Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, pages 157–175.

[Pennerath, 2009] Pennerath, F. (2009). *Méthodes d'extraction de connaissances à partir de données modélisables par des graphes. Application à des problèmes de synthèse organique*. PhD thesis, Université de Nancy.

[Piatetsky-Shapiro, 1991] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248.

[Piatetsky-Shapiro and Matheus, 1994] Piatetsky-Shapiro, G. and Matheus, C. J. (1994). The interestingness of deviations. In *AAAI'94 Workshop on Knowledge Discovery in Databases*, pages 25–36.

[Pietracaprina and Zandolin, 2003] Pietracaprina, A. and Zandolin, D. (2003). Mining frequent itemsets using patricia tries.

[Power, 2002] Power, D. J. (10/11/2002). Theory and generalized dss, "true story" about data mining, beer and diapers. http://www.dssresources.com/newsletters/66.php.

[Savasere et al., 1998] Savasere, A., Omiecinski, E., and Navathe, S. (1998). Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the 14th International Conference on Data Engineering*, ICDE, pages 494–502.

[Shortliffe, 1976] Shortliffe, E. (1976). *Computer-Based Medical Consultations : MYCIN*.

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

[Smyth and Goodman, 1992] Smyth, P. and Goodman, R. M. (1992). An information theoretic approach to rule induction from databases. *Transactions on Knowledge and Data Engineering*, 4(4) :301–316.

[Subramanian et al., 2003] Subramanian, D. K., Ananthanarayana, V. S., and Murty, M. N. (2003). Knowledge-based association rule mining using and–or taxonomies. *Knowledge-Based Systems*, 16(1) :37–45.

[Suzuki, 1997] Suzuki, E. (1997). Autonomous discovery of reliable exception rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, KDD, pages 259–262.

[Suzuki, 1999] Suzuki, E. (1999). Scheduled discovery of exception rules. In *Proceedings of the 2nd Internation Conference on Discovery Science*, DS, pages 184–195.

[Suzuki, 2004] Suzuki, E. (2004). Discovering interesting exception rules with rule pair. In *Proceedings of the ECML/PKDD Workshop on Advances in Inductive Rule Learning*, pages 163–178.

[Suzuki, 2006] Suzuki, E. (2006). Data mining methods for discovering interesting exceptions from an unsupervised table. *Journal of Universal Computer Science*, 12(6) :627–653.

[Suzuki and Kodratoff, 1998] Suzuki, E. and Kodratoff, Y. (1998). Discovery of surprising exception rules based on intensity of implication. In *Proceedings of the 2nd European Symposium on Principles and Practice of Knowledge Discovery in Databases*, PKDD, pages 10–18.

[Suzuki and Shimura, 1996] Suzuki, E. and Shimura, M. (1996). Exceptional knowledge discovery in databases based on information theory. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, KDD, pages 275–278.

[Suzuki and Zytkow, 2000] Suzuki, E. and Zytkow, J. M. (2000). Unified algorithm for undirected discovery of execption rules. PKDD, pages 169–180.

[Taniar et al., 2012] Taniar, D., Rahayu, W., Daly, O., and Nguyen, H.-Q. (2012). Mning hierarchical negative association rules. *International Journal of Computational Intelligence Systems*, 5(3) :434–451.

[Teng et al., 2002] Teng, W.-G., Hsieh, M.-J., and Chen, M.-S. (2002). On the mining of substitution rules for statistically dependent items. In *Proceedings of the 3rd International Conference on Data Mining*, ICDM, pages 442–449.

[Teng et al., 2005] Teng, W.-G., Hsieh, M.-J., and Chen, M.-S. (2005). A statistical framework for mining substitution rules. *Knowledge and Information Systems*, 7(2) :158–178.

[Thiruvady and Webb, 2004] Thiruvady, D. R. and Webb, G. I. (2004). Mining negative rules using grd. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD, pages 161–165.

[Tsai et al., 2010] Tsai, L.-M., Lin, S.-J., and Yang, D.-L. (2010). Efficient mining of generalized negative association rules. In *International Conference on Granular Computing*, GrC, pages 471–476.

[Wang et al., 2008] Wang, H., Zhang, X., and Chen, G. (2008). Mining a complete set of both positive and negative association rules from large databases. In *Advances in Knowledge Discovery and Data Mining*, pages 777–784.

[Webb, 2000] Webb, G. I. (2000). Efficient search for association rules. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, KDD, pages 99–107.

[Wu et al., 2004] Wu, X., Zhang, C., and Zhang, S. (2004). Efficient mining of both positive and negative association rules. *Transactions on Information Systems (TOIS)*, 22(3) :381–405.

[Wu et al., 2011] Wu, Y.-Y., Chen, Y. L., and Chang, R.-I. (2011). Mining negative generalized knowledge from relational databases. *Knowledge-Based Systems*, 24(1) :134–145.

[Yuan et al., 2002] Yuan, X., Buckles, B. P., Yuan, Z., and Zhang, J. (2002). Mining negative association rules. In *Proceedings of the 7th Symposium on Computers and Communications*, ISCC, pages 623–628.

[Zaki et al., 1997] Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997). New algorithms for fast discovery of association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, SIGKDD, pages 283–286.

Index

S

T